1001نکته در C++

C++

۱۳۸۸

صابر عزیزپوریان

۱) زبان C++ از پایه و اساس زبانی شیء گرا می باشد.

۲) معمولا در تمامی برنامه های زبان C++ نیاز به افزودن هدر فایل داریم. جهت افزودن هدر فایل باید به طریق #include <headerfilename> عمل نمایید.

۳) درزبان C++ برنامه به توابع تقسیم بندی شده است. که با نوشتن توابع یک برنامه کامل می شود.

۴) در زبان C++ تمامی برنامه ها باید دارای تابع main باشند.

۵) تابع main همیشه یک مقدار بازگشتی دارد. برای تعیین نوع مقدار بازگشتی تابع به صورت type main(parameters) عمل می کنیم.

۶) در نکته قبل باید به جای type یکی از انواع تعریف شده زبان را قرار دهیم.

۷) به جای عبارت parameters باید پارامترهای ورودی تابع main را قرار دهیم. که در نکته های بعدی به آن خواهیم پرداخت.

۸) اگر به جای عبارت type یک نوع داده ای را قرار دهیم تابع باید حتما مقدار بازگشتی داشته باشد. که در انتهای برنامه هم باید return value را قرار دهیم.

۹) به جای value باید مقدار بازگشتی را که قرار است تابع بر گرداند را قرار دهیم.

۱۰)اگر بخواهیم تابع مقدار بازگشتی نداشته باشد باید به جای type عبارت void را قرار دهیم.

۱۱)زبان C++ به حروف کوچک و بزرگ حساس است و بین حروف کوچک و بزرگ تفاوت قائل می شود. و این عمل هم به دلیل پردازش حروف با کد اسکی آنها می باشد.

۱۲)در زبان C++ تمامی دستورات باید با حروف کوچک باشد.

۱۳)در زبان C++ معمولا ثوابت را با حروف بزرگ به کار می برند.

۱۴)در زبان C++ تمامی دستورات با ; خاتمه می باند.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۱۵)در زبان C++ حداکثر طول یک دستور ۲۵۵ کاراکتر می باشد. که این تعداد با احتساب فضاهای خالی و تمامی علائم و (;) ختم دستور می باشد.

۱۶)در زبان C++ شما هم می توانید یک دستور را در چند خط بنویسیم و هم اینکه می توانید چند دستور را در یک خط بگنجانید ، البته هیچکدام از دو مورد فوق توصیه نمی شود ، که دلیل آن هم بالا رفتن خوانایی کد شما می باشد.

۱۷)در زبان C++ شما باید برای توضیحات تک خطی خود ، بعد از اتمام دستور از دو علامت // پشت سر هم استفاده می کنیم.

۱۸)در زبان C++ شما باید برای توضیحات چند خطی خود ، بعد از اتمام دستور با علامت /* شروع شده و بعد از اتمام توضیحات با علامت */ به توضیحات خاتمه می دهیم.

۱۹)در زبان C++ برای انجام محاسبات و عملیات های ورودی و خروجی باید از متغیرها استفاده کنیم. که متغیرها را هرکدام نوع خاصی دارند.

۲۰)در زبان C++ برای تعریف متغیر از نوع صحیح باید از نوع داده ای int استفاده کنیم.

۲۱)در زبان C++ برای تعریف متغیر از نوع کاراکتر باید از نوع داده ای char استفاده کنیم.

۲۲)در زبان C++ برای تعریف متغیر از نوع اعشاری باید از نوع داده ای float استفاده کنیم.

۲۳)در زبان C++ برای تعریف متغیر از نوع اعشاری بزرگ باید از نوع داده ای double استفاده کنیم.

۲۴)در زبان C++ برای تعریف متغیر از نوع پوچ باید از نوع داده ای void استفاده کنیم.

۲۵)در زبان C++ میتوانیم از انواع موجود انواع دیگری را ساخت. این عمل با استفاده از کلمات کلیدی که با انواع داده ای ترکیب می شوند امکان پذیر می شود.

۲۶)در زبان C++ نوع داده ای int در سیستم های ۱۶ بیتی ۲ بایت می باشد و در سیستم های ۳۲ بیتی ۴ بایت می باشد.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۲۷)اگر به ابتدای نوع داده ای **int** کلمه **short** را بیافزاییم طول این نوع داده ای حتما ۲ بایت خواهد بود.

۲۸)اگر به ابتدای نوع داده ای **int** کلمه **long** را بیافزاییم طول این نوع داده ای حتما ۴ بایت خواهد بود.

۲۹)اگر به ابتدای نوع داده ای **double** کلمه **short** را بیافزاییم طول این نوع داده ای حتما ۸ بایت خواهد بود.

۳۰)اگر به ابتدای نوع داده ای **double** کلمه **long** را بیافزاییم طول این نوع داده ای حتما ۱۰ بایت خواهد بود.

۳۱)اگر به ابتدای یک نوع داده ای کلمه **unsigned** را بیافزاییم ، آن نوع داده ای فقط مقدارهای مثبت را دریافت می کند.

۳۲)اگر به ابتدای یک نوع داده ای کلمه **signed** را بیافزاییم ، آن نوع داده ای هم مقدارهای مثبت و هم مقدارهای منفی را می پذیرد.

۳۳)نام یک متغیر در زبان C++ حتما باید با حرف آغاز شود.

۳۴)نام یک متغیر در زبان C++ فقط ترکیبی از حروف و ارقام و زیر خط (_) می باشد.

۳۵)اگر به یک متغیر مثلا از نوع **unsigned Int** که حداکثر مقدار ۶۵۵۳۵ را می گیرد، عددی بیشتر از این رنج داده شود، مثلا، ۶۵۵۳۶ چون گنجایش آن را ندارد، دوباره شروع به مقدار دهی ارز ابتدا می کند، در مثال ذکر شده به جای ۶۵۵۳۶، عدد ۰ (صفر) ذخیره می شود.

۳۶)اگر به یک متغیر مثلا از نوع **signed int** که حداکثر مقدار ۳۲۷۶۷ را میگیرد و حداقل مقدار 32768- را میگیرد اگر مقداری بیش از مقدار قابل ذخیره دریافت کند مثلا ۳۲۷۶۸

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

مقدار متغیر به 32768- باز می گردد و در صورتی که کمتر از پایین ترین مقدار را دریافت کند مثلا 32769- مقدار متغیر به ۳۲۷۶۷ بازگشت خواهد کرد.

۳۷)نکته قبل در مورد تمامی انواع متغیرها یکسان می باشد و در تمامی انواع دیگر همین قضیه پابرجا خواهد بود.

۳۸)در زبان C++ هنگام تعریف متغیر اگر مقدار اولیه به آن نسبت داده نشود ، هر مقداری که از قبل در حافظه وجود داشته ، اکنون مقدار متغیر ما خواهد بود.

۳۹)در برنامه اگر بخواهیم مقداری را تعریف کنیم که در طول برنامه تغییر نکند و ثابت بماند ، باید در برنامه یک ثابت تعریف کنیم که تعریف ثابت در زبان C++ ۲ راه دارد.

۴۰)راه اول برای تعرییف ثابت در برنامه استفاده از #define می باشد. و نحوه تعریف ثابت به شکل # define constantname value می باشد.

۴۱)باید به جای عبارت constantname نام ثابت را وارد نمایید.

۴۲)باید به جای عبارت value مقدار ثابت را وارد نمایید.

۴۳)در تعریف ثابت از نوع مذکور برنامه خود به خود نوع ثابت را شناسایی می کند و نیازی به تعریف نوع ثابت نیست.

۴۴)راه دوم برای تعریف ثابت در برنامه استفاده از const می باشد. که برای تعریف ثابت باید به روش const typeconst constantname = value; عمل نمایید.

۴۵)به جای عبارت typeconst باید نوع ثابت را تعریف نمایید.

۴۶)به جای عبارت constantname باید نام ثابت را مشخص نمایید.

۴۷)به جای عبارت value مقدار ثابت را تعیین می نمایید.

۴۸)در برنامه ها برای فرق قائل شدن بین متغیر و ثابت معمولا متغیرها را با حروف کوچک و ثوابت را با حروف بزرگ تعریف می کنند.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۴۹)در زبان C++ برای استفاده از متغیرها و انجام عملیات های مختلف باید در برنامه از عملگر ها استفاده شود.

۵۰)عملگر ++ در زبان C++ برای افزودن یک واحد به مقدار متغیر استفاده می شود. در واقع x++ برابر است با x=x+1 ، که شکل اول حالت کوتاه شده است.

۵۱)عملگر – در زبان C++ برای کاهش یک واحد به مقدار متغیر استفاده می شود. در واقع x--برابر است با x=x-1 ، که شکل اول حالت کوتاه شده است.

۵۲)در زبان C++ برای انجام عملیات جمع به صورت x=a+b عمل می نماییم.

۵۳)در زبان C++ برای انجام عملیات تفریق به صورت x=a-b عمل می نماییم.

۵۴)در زبان C++ برای انجام عملیات ضرب به صورت x=a*b عمل می نماییم.

۵۵)در زبان C++ برای انجام عملیات تقسیم کامل به صورت x=a/b عمل می نماییم.

۵۶)در نکته قبل مقدار a بر مقدار b تقسیم می شود و خارج قسمت آن بدست می آید که این خارج قسمت معمولا اعشاری خواهد بود.

۵۷)در زبان C++ برای بدست آوردن باقیمانده تقسیم a بر b به صورت x=a%b عمل می نماییم.

۵۸)در برنامه ها برای جدا کردن عملیات های ریاضی از عملگر پرانتز استفاده می کنیم.

۵۹)در برنامه ها برای And کردن متغیرها باید از عملگر & به صورت x=a&b عمل کنیم.

۶۰) جدول ذیل جدول صحت عملگر & می باشد .

E1	E2	E1 & E2
0	0	0
1	0	0
0	1	0
1	1	1

۶۱)در برنامه ها برای Or کردن متغیرها باید از عملگر | به صورت x=a|b عمل می کنیم.

۶۲)جدول ذیل که در صفحه بعد آمده جدول صحت عملگر | می باشد.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

E1	E2	E1 | E2
0	0	0
1	0	1
0	1	1
1	1	1

۶۳)در برنامه ها برای Xor کردن متغیرها از عملگر ^ به صورت x=x^y عمل می کنیم.

۶۴)جدول ذیل جدول صحت عملگر ^ می باشد.

E1	E2	E1 ^ E2
0	0	0
1	0	1
0	1	1
1	1	0

۶۵)در برنامه ها برای Not کردن مقدار یک متغیر از عملگر ~ به صورت x=~a استفاده می کنیم.

۶۶)در برنامه ها برای شیفت دادن یک متغیر به سمت راست برای یک بار از عملگر << به صورت x=a>>1 استفاده می کنیم. که می توانیم به جای عدد ۱ تعداد چرخش دلخواه را وارد نمایید.

۶۷)در برنامه ها برای شیفت دادن یک متغیر به سمت چپ برای یک بار از عملگر >> به صورت x=a<<1 استفاده می کنیم. که می توانیم به جای عدد ۱ تعداد چرخش دلخواه را وارد نمایید.

۶۸)در زبان C++ برای انجام عملیات And در شرط ها باید از عملگر && استفاده کنید.

۶۹)جدول صحت && در ذیل آمده است.

A	B	A&&B
T	T	T
T	F	F
F	T	F
F	F	F

۷۰)در زبان C++ برای انجام عملیات Or در شرط ها باید از عملگر || استفاده کنید.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۷۱)جدول صحت || در ذیل آمده است.

A	B	A||B
T	T	T
T	F	T
F	T	T
F	F	F

۷۲)در زبان C++ برای Not کردن در شرط ها از عملگر ! استفاده می کنیم.

۷۳)جدول صحت ! در ذیل آمده است.

A	!A
T	F
F	T

۷۴)در زبان C++ برای کاهش کدنویسی کاربران شما می توانید به جای x=x+y عبارت x+=y را جایگزین نمایید.

۷۵)در زبان C++ برای کاهش کدنویسی کاربران شما می توانید به جای x=x-y عبارت x-=y را جایگزین نمایید.

۷۶)در زبان C++ برای کاهش کدنویسی کاربران شما می توانید به جای x=x*y عبارت x*=y را جایگزین نمایید.

۷۷)در زبان C++ برای کاهش کدنویسی کاربران شما می توانید به جای x=x/y عبارت x/=y را جایگزین نمایید.

۷۸)در زبان C++ برای کاهش کدنویسی کاربران شما می توانید به جای x=x%y عبارت x%=y را جایگزین نمایید.

۷۹)در زبان C++ برای کاهش کدنویسی کاربران شما می توانید به جای x=x&y عبارت x&=y را جایگزین نمایید.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۸۰)در زبان C++ برای کاهش کدنویسی کاربران شما می توانید به جای x=x|y عبارت x|=y را جایگزین نمایید.

۸۱)در زبان C++ برای کاهش کدنویسی کاربران شما می توانید به جای x=x^y عبارت x^=y را جایگزین نمایید.

۸۲)در زبان C++ برای کاهش کدنویسی کاربران شما می توانید به جای x=x<<y عبارت x<<=y را جایگزین نمایید.

۸۳)در زبان C++ برای کاهش کدنویسی کاربران شما می توانید به جای x=x>>y عبارت x>>=y را جایگزین نمایید.

۸۴)تمامی برنامه ها میتوانند ورودی نداشته باشند ، اما نمی توانند خروجی نداشته باشند.

۸۵)در زبان C++ برای نمایش اطلاعات در خروجی از تابع cout استفاده می شود. این تابع در هدر فایل #include <iostream.h> قرار دارد.

۸۶)حالت کلی این تابع به صورت cout<<"constant string"<<variable; می باشد. که به جای عبارت constant string شما می توانید پیغام در صفحه نمایش چاپ کنیدو به جای عبارت variable متغیرهایی را که میخواهید مقدارشان در خروجی نمایش داده شود را قرار می دهید. البته این عملیات می تواند چندین بار تکرار شود.

۸۷)در زبان C++ برای کنترل خروجی و فرمت بندی صفحه نمایش کاراکترهای کنترلی وجود دارد که در زیر با آنها آشنا می شویم.

۸۸)کاراکتر \n برای رد کردن سطر جاری و هدایت کرسر به سطر بعد استفاده می شود.

۸۹)کاراکتر endl برای رد کردن سطر جاری و هدایت کرسر به سزر بعداستفاده می شود.

۹۰)کاراکتر \t برای انتقال کرسر به ۸ ستون جلوتر استفاده می شود.

۹۱)کاراکتر \a برای به صدا درآوردن بوق سیستم استفاده می شود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۹۲)کاراکتر a\ به دلیل اینکه یک وقفه سخت افزاری را فراخوانی و اجرا می نماید. زمانی را از بر نامه صرف می کند. اگر در برنامه ها این دستور تعداد فراخوانی بالایی داشته باشد. زمان زیادی از برنامه هم تلف خواهد شد.

۹۳)کاراکتر \\ برای چاپ کردن کاراکتر \ در خروجی استفاده می شود.

۹۴)کاراکتر "\ برای چاپ کردن کاراکتر " در خروجی استفاده می شود.

۹۵)کاراکتر v\ برای انتقال کرسر به ۸ سطر بعد استفاده می شود.

۹۶)کاراکتر b\ برای حذف کاراکتر قبل از خود استفاده می شود.

۹۷)کاراکتر r\ برای انتقال کرسر به ابتدای سطر جاری به کار می رود.

۹۸)کاراکتر ?\ برای چاپ کردن کاراکتر ? در خروجی استفاده می شود.

۹۹)کاراکتر :\ برای چاپ کردن کاراکتر : در خروجی استفاده می شود.

۱۰۰) کاراکتر '\ برای چاپ کردن کاراکتر ' در خروجی استفاده می شود.

۱۰۱) برای دریافت اطلاعات از ورودی باید از تابع cin که در هدر فایل #include <iostream.h> قرار دارد.

۱۰۲) حالت کلی تابع به صورت cin>>var>>var>>...; می باشد. که به جای عبارت var نام متغیری را که میخواهید اطلاعات آن را از ورودی دریافت کنید قرار می دهید.

۱۰۳) روند اجرای دستورات در برنامه ها به این صورت است که همیشه دستورات از بالا به پایین سطر به سطر و در هر سطر از چپ به راست اجرا می گردد.

۱۰۴) گاهی اوقات در برنامه ها مجبوریم از روال عادی اجرای دستورات خارج شویم و یک سری از دستورات را اجرا نکنیم و یک سری از دستورات را در بعضی شرایط خاص اجرا کنیم. در این شرایط مجبوریم که از شرط ها در برنامه بهره ببریم.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۱۰۵) حالت کلی ساختار if به صورت if(condition) می باشد. که به جای عبارت condition باید یک عبارت شرطی را قرار دهیم.

۱۰۶) شرط ها همیشه یک مقدار نهایی ایجاد می کنند که از دو حالت خارج نمی باشد. حالت اول:True و حالت دوم:False می باشد.

۱۰۷) در زبان C++ مقدار False برابر با ۰ بوده و مقدار True برابر با مقداری غیر صفر می باشد.

۱۰۸) طبق نکته قبل شما در شرط ها به جای نوشتن یک عبارت شرطی می توانید از یک عبارت محاسباتی استفاده نمایید.

۱۰۹) در ساختار if اگر بدنه if دارای چند دستور باشد باید ابتدای دستورات { و انتهای دستورات نیز } را قرار دهیم. تا دستورات درون یک بلاک قرار گیرند.

۱۱۰) در ساختار if اگر بدنه if دارای یک دستور باشد نیازی به استفاده از { و } نداریم.

۱۱۱) در ساختار if اگر بعد از ساختار علامت ; را قرار دهیم بدنه if خالی شده و دستورات مربوط به if در هر حالتی اجرا خواهند شد. مقدار شرط true باشد یا false تاثیری در اجرای دستورات بدنه نخواهد داشت و بدنه حتما اجرا خواهد شد.

۱۱۲) توجه داشته باشید که هر شرط فقط یک دوراهی را مشخص می کند. که البته هر کدام از راه ها ممکن است خود شامل چندین راه باشد. هنگامی که میخواهید در چنین شرایطی برنامه بنویسید سعی کنید که حالت ها را از هم جدا کنید و یک به یک آنها را در شرط های پشت سر هم چک کنید. تا مشکلی در برنامه ایجاد نشود.

۱۱۳) کاربرد عملگر ! در شرط ها این است که اگر مقدار یک شرط True باشد آن را به False و بالعکس تبدیل می کند.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۱۱۴) اگر در شرط ها بخواهیم برابری دو مقدار را با هم چک کنیم باید از عملگر == استفاده نماییم.

۱۱۵) اگر در شرط ها بخواهیم نابرابری دو مقدار را با هم چک کنیم باید از عملگر =! استفاده نماییم.

۱۱۶) اگر بخواهیم چک کنیم که مقدار سمت چپ از مقدار سمت راست بزرگتر باشد باید از عملگر > استفاده نماییم.

۱۱۷) اگر بخواهیم چک کنیم که مقدار سمت چپ از مقدار سمت راست کوچکتر باشد باید از عملگر < استفاده نماییم.

۱۱۸) اگر بخواهیم چک کنیم که مقدار سمت چپ از مقدار سمت راست کوچکتر مساوی باشد باید از عملگر => استفلده نماییم.

۱۱۹) اگر بخواهیم چک کنیم که مقدار سمت چپ از مقدار سمت راست بزرگتر مساوی باشد باید از عملگر =< استفاده نماییم.

۱۲۰) عملگر ? در برنامه برای چک کردن شرط به کار می رود. حالت کلی این عملگر به شکل var=condition?val1:val2; می باشد. که به جای var متغیر مقصد را قرار می دهیم و به جای condition شرط را می نویسیم ، اگر مقدار شرط True ارزیابی شود val1 در var قرار می گیرد و اگر مقدار شرط False ارزیابی شود val2 در var قرار می گیرد.

۱۲۱) در شرط ها اگر بخواهیم حالت های غیر از شرط مورد نظر را هم در نظر بگیریم باید از دستور else استفاده کنیم که بیانگر این است که در غیر اینصورت... چه کاری انجام گیرد.

۱۲۲) در استفاده از else باید دقت کرد زیرا تمامی حالات به جز حالت چک شده در این حالت قرار می گیرد. باید دقت کرد که حالت های معتبر را در else بررسی کنیم.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۱۲۳) هنگامی که می خواهیم چندین حالت نقطه ای و معلوم را بررسی کنیم معمولا به جای استفاده از if/else برای بالا رفتن خوانایی کد از ساختار switchاستفاده می کنیم.

۱۲۴) حالت کلی switch به شکل روبرو می باشد.

```
Switch ( condition )
{
Case   value1:
            Statements
Case   value2:
            Statements
...
Case   value(n):
            Statements
Defaults:
            Statements
}
```

۱۲۵) در ساختار نکته ۱۰۷ به جای عبارت condition شرط مورد نظر را قرار می دهید. به جای value ها مقدارهایی که قرار است با برابر بودن شرط عملی انجام دهیم. و به جای statements دستورات را تایپ میکنیم. و در حالت default در واقع تمامی حالات ذکر نشده قابل پذیرش هستند.

۱۲۶) دستور goto در برنامه ها برای پرش از قسمتی از برنامه به قسمت دیگری از برنامه استفاده می شود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۱۲۷) در زبان C++ برای تکرار یک دستور به تعداد محدود و حتی نامحدود ابزارهایی وجود دارد که ۳ ابزار مستقیم و یک ابزار غیر مستقیم وجود دارد.که ۳ ابزار مستقیم حلقه های تعریفی خود زبان می باشند و یک ابزار غیر مستقیم دستور goto به همراه شرط می باشد.

۱۲۸) یکی از پرکاربردترین حلقه های موجود در برنامه نویسی حلقه for می باشد. که حالت کلی آن به شکل for(first value ; condition ; step) می باشد.

۱۲۹) به جای قسمت اول حلقه first value مقدار اولیه حلقه را تعیین کنیم. معمولا در این قسمت یک متغیر به عنوان شمارنده حلقه وجود دارد که مقدار اولیه به آن نسبت داده می شود.

۱۳۰) به جای قسمت دوم condition شرط پایان حلقه را قرار می دهیم. اگر مقدار شرط false شود حلقه به کار خود خاتمه خواهد داد.

۱۳۱) به جای قسمت سوم step گام حلقه قرار می گیرد. این قسمت بعد از هر بار تکرار حلقه انجام می گیرد.

۱۳۲) حلقه for معمولا برای تکرارهای محدود که تعداد تکرارشان معلوم می باشد استفاده می شود.

۱۳۳) در حلقه for شما ابتدا یک مقدار اولیه را به شمارنده حلقه می دهید و برای حلقه تعیین می کنید که تا زمانی که شرط حلقه بر قرار می باشد تکرار حلقه را ادامه بده و در هر بار تکرار حلقه گام حلقه را به شمارنده اعمال کند.

۱۳۴) هرکدام از سه قسمت حلقه for را به دلخواه می توانید حذف کنید و به هر ترتیبی که نیاز است از آن استفاده کنید.

۱۳۵) در مورد قسمت اول حلقه که مقدار اولیه شمارنده را دریافت می کند، اگر خالی بماند مقدار شمارنده هرچه که از قبل بوده تنظیم می گردد و ممکن ایت که حلقه دچار استباه شود. برای اینکه چنین مشکلی پیش نیاید باید قبل از حلقه مقدار را تنظیم نمایید.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۱۳۶) در مورد قسمت دوم حلقه که شرط حلقه قرار می گیرد، اگر خالی بماند حلقه ممکن است به حلقه بی نهایت تبدیل شود. زیرا شرط را همیشه درست در نظر می گیرد. و فقط در صورتی می توان از حلقه خارج شد که درون حلقه با دستوری خاص از حلقه خارج شویم.

۱۳۷) در مورد قسمت سوم حلقه که گام حلقه قرار می گیرد، اگر خالی بماند حلقه ممکن است بی نهایت شود و در مقدار اولیه خود در جا بزند. مگر اینکه درون حلقه با دستوری مقدار شمارنده را دستکاری کنید.

۱۳۸) اگر بدنه حلقه دارای چند دستور باشد باید دستورات را درون } و { قرار دهید تا به عنوان یک بلاک با هر بار تکرار حلقه آنها هم تکرار شوند.

۱۳۹) اگر بدنه حلقه دارای یک دستور باشد نیازی به قرار دادن } و { نیست و حلقه به طور اتوماتیک با هر بار تکرار خود دستور بعد از خود را نیز اجرا می نماید.

۱۴۰) اگر بعد از حلقه ; قرار دهید ، بدنه حلقه خالی فرض شده و دستورات فقط یک بار اجرا خواهند شد.

۱۴۱) توجه داشته باشید که اگر حلقه را به صورت for (; ;) بنویسید ، حلقه بی نهایت خواهد بود. زیرا تمامی گزینه ها به صورت همیشه درست خواهند بود.

۱۴۲) در حلقه می توان در کنار هر قسمت دستورات متفرقه هم نوشت. که این عمل با قرار دادن جدا کننده ها امکان پذیر خواهد بود.

۱۴۳) یکی دیگر از حلقه های پرکاربرد ، حلقه while می باشد، که حالت کلی این حلقه به صورت while(condition) می باشد.

۱۴۴) به جای قسمت condition باید شرط خاتمه را قرار دهید.

۱۴۵) این حلقه معمولا برای تکرارهایی که تعداد آن معلوم نمی باشد استفاده می شود.

۱۴۶) در این حلقه در صورت وجود شمارنده مقدار اولیه را باید قبل از حلقه تنظیم نمایید.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۱۴۷) در این حلقه در صورت وجود شمارنده گام حلقه را باید درون حلقه دستکاری کنید.

۱۴۸) اگر بعد از while(condition) از ; استفاده کنیم بدنه حلقه خالی می شود. و دستورات فقط یک بار و خارج از حلقه اجرا می شوند.

۱۴۹) اگر تعداد دستورهای حلقه بیش از یک دستور باشد باید آنها را درون } و { قرار داد.

۱۵۰) اگر تعداد دستورات حلقه یک دستور باشد ، نیازی به } و { نیست و اتوماتیک اجرا می شود.

۱۵۱) یکی دیگر از حلقه های پرکاربرد ، حلقه do/while می باشد، که حالت کلی این حلقه به صورت do statements… while(condition) می باشد.

۱۵۲) به جای قسمت statements… دستوراتی را که میخواهیم درون حلقه تکرار شوند را قرار می دهیم.

۱۵۳) به جای قسمت condition شرطی را قرار می دهیم که حلقه تا زمانی که این شرط برقرار باشد اجرا می گردد.

۱۵۴) این حلقه زمانی استفاده می شود که بخواهیم دستورات یک بار اجرا شود و اگر شرط حلقه بر قرار بود حلقه به کار خود ادامه دهد و در غیر اینصورت از حلقه خارج شویم.

۱۵۵) این حلقه برای تعداد تکرارهای نامعلوم استفاده می شود.

۱۵۶) اگر بخواهیم در حلقه از شمارنده استفاده کنیم باید شمارنده را قبل از حلقه مقدار دهی کنیم.

۱۵۷) اگر بخواهیم از شمارنده در حلقه استفاده کنیم باید درون حلقه به صورت دستی گام حلقه را تنظیم و دستکاری کنیم.

۱۵۸) در این حلقه بعد از while(condition) باید ; قرار دهیم تا پایان حلقه مشخص گردد.

۱۵۹) در زبان C++ دستوراتی وجود دارند که جزء زبان نمی باشد ولی زبان از آنها پشتیبانی می کند، این دستورات را دستورات پیش پردازنده می نامند.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۱۶۰) گاهی اوقات دربرنامه ها نیاز به تعریف تعدادزیادی متغیر از یک نوع را داریم. در این مواقع می توانیم به جای اینکه چندین متغیر را تعریف کنیم یک آرایه را تعریف کنیم با چندین خانه که کار برنامه نویس راحت شود.

۱۶۱) نحوه تعریف آرایه به صورت type namearray[length]; می باشد.

۱۶۲) به جای عبارت type باید یکی از انواع موجود در زبان و یا یکی از انواع تعریفی خود کاربر باشد.

۱۶۳) به جای عبارت namearray باید نام آرایه را وارد نمایید. نام آرایه از قوانین تعریف نام متغیر تبعیت می کند.

۱۶۴) به جای عبارت length هم باید طول آرایه را وارد نماییم. البته این مقدار باید حتما یک عدد صحیح باشد و حداکثر طول هم دارد.

۱۶۵) حد اکثر طول تعریفی آرایه در مجموع نباید بیش از 64kb شود.

۱۶۶) نحوه بدست آوردن حجم اشغالی یک آرایه باید از فرمول: حجم کلی=تعداد خانه ها * حجم یک خانه بدست می آید.

۱۶۷) هنگامی که بخواهید چند خاصیت از چند چیز را در کنار هم ذخیره کنید باید آرایه دو بعدی تعریف نماییم.

۱۶۸) اگر بخواهیم شکل هندسی یک آرایه یک بعدی را ترسیم نماییم باید یک لیست را مثال بزنیم.

۱۶۹) نحوه تعریف یک آرایه دو بعدی به شکل type namearray [lrow][lcol]; می باشد.

۱۷۰) به جای عبارت type باید یکی از انواع موجود در زبان و یا یکی از انواع تعریفی خود کاربر باشد.

۱۷۱) به جای عبارت namearray باید نام آرایه دوبعدی را وارد نمایید. نام آرایه از قوانین تعریف نام متغیر تبعیت می کند.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۱۷۲) به جای عبارت lrow هم باید تعداد سطرهای ارایه دو بعدی را تعیین نمایید.

۱۷۳) به جای عبارت lcol هم باید تعداد ستون های آرایه دو بعدی را تعیین نمایید.

۱۷۴) حد اکثر طول تعریفی یک آرایه دو بعدی در مجموع نباید بیش از 64kb شود.

۱۷۵) نحوه بدست آوردن حجم اشغالی یک آرایه باید از فرمول: حجم کلی=تعداد سطرها *تعداد ستون ها * حجم یک خانه بدست می آید.

۱۷۶) اگر بخواهیم شکل هندسی یک آرایه دو بعدی را ترسیم نماییم باید یک ماتریس را مثال بزنیم.

۱۷۷) به همان ترتیبی که در بالا توضیح داده شد می توان آرایه های با ابعاد بالاتر را نیز ایجاد کرد.

۱۷۸) در برنامه ها می توان به سادگی یک نوع داده ای جدید را به دلخواه خود تعریف کنیم. که این عمل را با استفاده از struct و union انجام می دهیم.

۱۷۹) برای تعریف یک نوع داده ای جدید با استفاده از ساختار باید به صورت struct stname عمل نماییم.

۱۸۰) به جای عبارت stname نام ساختار جدید را تعیین نماییم. که این نام از قوانین تعریف نام متغیر تبعیت می کند.

۱۸۱) سپس برای تعریف اجزاء این ساختار به شکل زیر عمل کنیم.

```
Struct stname{
Type  p1;
...
}s1,s2,...;
```

۱۸۲) به جای عبارت type نوع داده ای که در زبان وجود دارد و یا نوع داده ای که از قبل تعریف کرده ایم.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۱۸۳) به جای عبارت p1 نام متغیری از نوع type می باشد. که این متغیرها می توانند تعدادی بیش از یکی داشته باشند.

۱۸۴) در قسمت آخر هم به جای هر یک از عبارات s1 و s2 ... می توان یک متغیر از نوع ساختار تعریفی ایجاد کرد.

۱۸۵) برای دسترسی به عناصر یک ساختار به صورت st.p عمل می نماییم.

۱۸۶) به جای عبارت st باید نام ساختاری را که می خواهیم با آن کار کنیم را وارد می کنیم.

۱۸۷) به جای عبارت p باید نام جزء مورد نظر از ساختار را وارد کنیم که می خواهم به آن دستیابی داشته باشیم.

۱۸۸) اگر بخواهیم حجم اشغالی توسط یک ساختار را بدست آوریم باید حجم تک تک اجزاء ساختار را با هم جمع کنیم تا حجم کل ساختار بدست آید.

۱۸۹) شما می توانید از یک ساختار یک آرایه نیز تعریف نمایید. به طور مثال زمانی که بخواهیم اطلاعات چندین دانشجو را ذخیره نماییم از این روش استفاده می نماییم.

۱۹۰) اگر بخواهیم حجم اشغالی توسط یک آرایه از ساختار را بدست آوریم باید از فرمول حجم آرایه = حجم یک ساختار * تعداد خانه های آرایه بدست آوریم.

۱۹۱) در تعریف ساختار می توان از روش union هم استفاده کرد. برای تعریف یک union از روش union uniname استفاده می نماییم.

۱۹۲) Union و struct دقیقا همانند یک دیگر عمل می نمایند. با این تفاوت که در هر زمان فقط می توان به یکی از union ها دسترسی داشت.

۱۹۳) برای ایجاد لیست های پیوندی معمولا از struct ها استفاده می شود.

۱۹۴) در یک struct میتوان اجزاء را اشاره گر هایی به خود struct قرار داد.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۱۹۵) در برنامه ها اگر بخواهیم یک شرط را چک کنیم و در صورت برقرار بودن شرط برنامه اجرا شود و یا اجرا نشود باید از دستور پیش پردازنده #if استفاده نماییم.

۱۹۶) در برنامه ها اگر بخواهیم دو شرط را چک نماییم و در صورت برقرار بودن شرط برنامه را اجرا کنیم یا نکنیم. باید از دستور پیش پردازنده #else استفاده نمایید.

۱۹۷) در برنامه ها اگر بخواهیم بیش از یک شرط را چک کنیم و در صورت برقرار بودن شرط ها برنامه را اجرا نماییم یا اجرا نکنیم باید از دستور #elif استفاده کنید.

۱۹۸) هنگامی که شرط های ابتدایی برنامه را چک کردیم باید در انتهای شرط ها دستور #endif را اضافه نمایید.

۱۹۹) اگر در برنامه ها بخواهیم قبل از اجرای برنامه تعریف بودن یک شناسه را چک کنیم باید از دستور #ifdef استفاده نماییم.

۲۰۰) اگر بخواهیم قبل از اجرای برنامه تعریف نبودن یک شناسه را در برنامه چک کنیم باید از دستور #ifndef استفاده نماییم.

۲۰۱) برای نمایش پیغام خطا در برنامه ها به صورت پیش پردازنده باید از دستور #error mes استفاده نمایید.

۲۰۲) به جای mes باید پیغام خطای مورد نظر را وارد نمایید.

۲۰۳) هنگامی که در برنامه بخواهیم یک ثابت تعریف شده که از نوع ماکرو تعریف شده است را از حالت تعریف خارج کنیم باید از دستور پیش پردازنده #undef constantname استفاده کنید.

۲۰۴) به جای عبارت constantname باید نام ثابتی را که می خواهید از برنامه حذف نمایید را وارد کنید.

۲۰۵) برای نمایش پیغام توجه در برنامه باید از دستور پیش پردازنده #warn warn_level استفاده شود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۲۰۶) به جای عبارت warn_level یکی از اعداد ۰ تا ۳ را بلید قرار دهید.

۲۰۷) در زبان C++ برای راحتی کار برنامه نویس کتابخانه بسیار بزرگی از توابع از پیش نوشته شده قرار دارد. برای استفاده از این توابع باید هدر فایل ها را به برنامه افزود. که در ادامه با این توابع و حالت کلی و طرز کار آنها آشنا خواهید شد.

۲۰۸) هنگامی که بخواهیم یک برنامه را به صورت غیر نرمال به پایان برسانیم از یک تابع به نام abort( ) استفاده می کنیم. بهد از فراخوانی این تابع یک پیغام مبنی بر abnormal program termination را چاپ می کند، سپس تابع exit را با کد ۳ فراخوانی می کند و به اجرای برنامه پایان می دهد. این تابع در هدر فایل های <stdlib.h> یا <process.h> قرار دارد.

۲۰۹) تابع abs(number) برای بدست آوردن قدر مطلق یک عدد به کار می رود که به جای number باید یک عدد صحیح از نوع int قرار بگیرد. که برای استفاده از این توابع باید یکی از هدر فایل <math.h> را به برنامه افزود.

۲۱۰) x=absread(int drive,int nsects,long lsect,void *buffer) تابع فوق برای خواندن سکتور به سکتور اطلاعات می باشد.

۲۱۱) در قسمت int drive شماره درایو مورد نظر را وارد می نمایید، که A=0,B=1,C=2,... به همین ترتیب تا شماره آخرین درایو که در سیستم شما موجود است.

۲۱۲) در قسمت int nsects عددی را مبنی بر تعداد سکتوری که می خواهید بخوانید را وارد می نمایید.

۲۱۳) در قسمت long lsect شماره سکتور منطقی که می خواهید اطلاعات را از آن بخوانید وارد نمایید.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۲۱۴) به جای void *buffer آدرس بافری را که می خواهید اطلاعات را در آن ذخیره نمایید را در آن وارد می نمایید.

۲۱۵) بعد از اجرای تابع دو مقدار بازگردانده می شود. ۰ به معنای اجرای موفقیت امیز تابع و 1- به معنای اجرای ناموفق تابع می باشد.

۲۱۶) بعد از اجرای تابع شماره خطای اتفاق افتاده را در رجیستر AX ذخیره می کند. این تابع در هدر فایل <dos.h> قرار دارد.

۲۱۷) x=abswrite(int drive,int nsects,long lsect,void *buffer)

تابع فوق برای نوشتن سکتور به سکتور اطلاعات بر روی دیسک می باشد .

۲۱۸) به این شکل که در قسمت int drive شماره درایو مورد نظر را وارد می نمایید، که A=0,B=1,C=2,... به همین ترتیب تا شماره آخرین درایوی که در سیستم شما موجود می باشد.

۲۱۹) سپس در قسمت int nsects عددی را مبنی بر تعداد سکتوری که می خواهید بنویسید را وارد می نمایید.

۲۲۰) در قسمت long lsect شماره سکتور منطقی که می خواهید اطلاعات را در آن بنویسید وارد نمایید.

۲۲۱) به جای void *buffer آدرس بافری را که می خواهید اطلاعات را از آن خوانده و در دیسک ذخیره نمایید را در آن وارد می نمایید.

۲۲۲) سپس بعد از اجرای تابع دو مقدار بازگردانده می شود. ۰ به معنای اجرای موفقیت امیز تابع و 1- به معنای اجرای ناموفق تابع. سپس شماره خطای اتفاق افتاده را در رجیستر AX ذخیره می کند. این تابع در هدر فایل <dos.h> قرار دارد.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۲۲۳) تابع acos(double x) یک مقدار اعشاری از نوع double را به عنوان پارامتر دریافت کرده و مقدار آرک کسینوس آن پارامتر را محاسبه کرده و یک مقدار اعشاری double را برمیگرداند. این تابع در هدرفایل <math.h> قرار دارد.

۲۲۴) تابع acos(complex x) یک مقدار مختلط از نوع complex را به عنوان پارامتر دریافت کرده و مقدار آرک کسینوس آن پارامتر را محاسبه کرده و یک مقدار مختلط را برمیگرداند. این تابع در هدرفایل <complex.h> قرار دارد.

۲۲۵) تابع acosl(long double x) یک مقدار اعشاری از نوع long double را به عنوان پارامتر دریافت کرده و مقدار آرک کسینوس آن پارامتر را محاسبه کرده و یک مقدار اعشاری long double را برمیگرداند. این تابع در هدرفایل <math.h> قرار دارد.

۲۲۶) تابع acosl(complex x) یک مقدار مختلط از نوع complex را به عنوان پارامتر دریافت کرده و مقدار آرک کسینوس آن پارامتر را محاسبه کرده و یک مقدار مختلط را برمیگرداند. این تابع در هدرفایل <complex.h> قرار دارد.

۲۲۷) تابع arc(int x,int y,int stangle,int endangle,int radius) برای ترسیم یک نیم دایره می باشد که باید به جای پارامتر int x نقطه طول مرکز نیم دایره را مشخص می کند، و به جای پارامتر int y نقطه عرض مرکز نیم دایره را مشخص می کند، به جای پارامتر int stangle زاویه شروع ترسیم نیم دایره را مشخص می کند، به جای پارامتر int endangle زاویه پایان ترسیم نیم دایره را مشخص می کند، و به جای پارامتر int radius باید میزان شعاع نیم ایره را وارد کنید. سپس برای شما در مختصات و تنظیماتی که انتخاب نموده اید یک نیم دایره برایتان ترسیم خواهد شد.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۲۲۸) تابع asin(double x) یک مقدار اعشاری از نوع double را به عنوان پارامتر دریافت کرده و مقدار آرک سینوس آن پارامتر را محاسبه کرده و یک مقدار اعشاری double را برمیگرداند. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۲۲۹) تابع asin(complex x) یک مقدار مختلط از نوع complex را به عنوان پارامتر دریافت کرده و مقدار آرک سینوس آن پارامتر را محاسبه کرده و یک مقدار مختلط را برمیگرداند. برای استفاده از این تابع باید هدر فایل <complex.h> را به برنامه افزود.

۲۳۰) تابع asinl(long double x) یک مقدار اعشاری از نوع long double را به عنوان پارامتر دریافت کرده و مقدار آرک سینوس آن پارامتر را محاسبه کرده و یک مقدار اعشاری long double را برمیگرداند. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۲۳۱) تابع asinl(complex x) یک مقدار مختلط از نوع complex را به عنوان پارامتر دریافت کرده و مقدار آرک سینوس آن پارامتر را محاسبه کرده و یک مقدار مختلط را برمیگرداند. برای استفاده از این تابع باید هدر فایل <complex.h> را به برنامه افزود.

۲۳۲) زبان C++ به شما اجازه می دهد تا برای راحتی کارتان و باز بودن دست برنامه نویس برای انجام هر کاری از قطعه کدهای اسمبلی در میان قطعه کدهای زیان C++ استفاده کند. برای این کار کافی است شما کلمه کلیدی asm را تایپ کرده و بعد از باز کردن { تا پایان یافتن کدها که } را می بندید قطعه کدهای خود را تایپ کنید. که البته این بلوک مجزا از بلوک اصلی برنامه و بلوکی درون برنامه اصلی به حساب می آید.

۲۳۳) تابع assert(int test) یک مقدار را به عنوان ورودی می پذیرد و آن مقدار را بررسی می کند اگر مقدار ورودی برابر با ۰ باشد ابتدا پیغامی مبنی بر Assertion failed: test سپس تابع abort( ) را فراخوانی می کند و برنامه را به صورت غیر نرمال به پایان می رساند.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۲۳۴) تابع atan(double x) یک مقدار اعشاری از نوع double را به عنوان پارامتر دریافت کرده و مقدار آرک تانژانت آن پارامتر را محاسبه کرده و یک مقدار اعشاری double را برمیگرداند.برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۲۳۵) تابع atan(complex x) یک مقدار مختلط از نوع complex را به عنوان پارامتر دریافت کرده و مقدار آرک تانژانت آن پارامتر را محاسبه کرده و یک مقدار مختلط را برمیگرداند. برای استفاده از این تابع باید هدر فایل <complex.h> را به برنامه افزود.

۲۳۶) تابع atanl(double x) یک مقدار اعشاری از نوع long double را به عنوان پارامتر دریافت کرده و مقدار آرک تانژانت آن پارامتر را محاسبه کرده و یک مقدار اعشاری long double را برمیگرداند.برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۲۳۷) تابع atanl(complex x) یک مقدار مختلط از نوع complex را به عنوان پارامتر دریافت کرده و مقدار آرک تانژانت آن پارامتر را محاسبه کرده و یک مقدار مختلط را برمیگرداند. برای استفاده از این تابع باید هدر فایل <complex.h> را به برنامه افزود.

۲۳۸) تابع atan2(double y, double x) دو مقدار اعشاری از نوع double را به عنوان پارامتر دریافت کرده که در واقع این دو مقدار مختصات یک نقطه هستند و مقدار آرک تانژانت آن نقطه را محاسبه کرده و یک مقدار اعشاری double را برمیگرداند. که برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۲۳۹) تابع atan2l(lomg double y, long double x) دو مقدار اعشاری از نوع double long را به عنوان پارامتر دریافت کرده که در واقع این دو مقدار مختصات یک نقطه هستند و مقدار آرک تانژانت آن نقطه را محاسبه کرده و یک مقدار اعشاری long double را برمیگرداند. که برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۲۴۰) تابع atexit(atexit_t func) این تابع باعث میشود هنگامی که برنامه به طور نرمال در حال بسته شدن می باشد یک تابع را فراخوانی کرده و دستورات آن تابع را اجرا نماید.که نام تابع را ترجیحا نام exit_fn1 انتخاب نمایید.

۲۴۱) البته به تعداد توابع فراخوانی شده عدد گذاری از یک به بالا افزایش می یابد.

۲۴۲) توجه داشته باشید که تابع فوق ۲ مقدار بازگشتی ممکن است داشته باشد که مقدار اول ۰ به معنی موفقیت آمیز بودن اجراو غیر صفر به معنی عدم موفقیت در اجرا می باشد.

۲۴۳) نکته جالب در مورد این تابع این است که در فراخوانی ها برعکس اجرای عادی برنامه ها از آخرین فراخوانی به سمت اولین فراخوانی حرکت میکنید.

۲۴۴) تابع double atof(const char *s) که برای تبدیل یک رشته به عدد اعشاری کاربرد دارد. به این شکل که شما یک عدد اعشاری که رشته می باشد را به عنوان پارامتر ورودی به این تابع می دهید و این تابع برای شما این مقدار رشته ای را به یک مقدار اعشاری عددی تبدیل می نمایید. برای استفاده از این تابع باید هدر فایل های <math.h> یا <stdlib.h> به برنامه افزوده شود.

۲۴۵) تابع int atoi(const char *s) که برای تبدیل یک رشته به عدد صحیح کاربرد دارد. به این شکل که شما یک عدد صحیح که رشته می باشد را به عنوان پارامتر ورودی به این تابع می دهید و این تابع برای شما این مقدار رشته ای را به یک مقدار صحیح عددی تبدیل می نمایید. برای استفاده از این تابع باید هدر فایل <stdlib.h> به برنامه افزوده شود.

۲۴۶) تابع long atol(const char *s) که برای تبدیل یک رشته به عدد صحیح بزرگ کاربرد دارد. به این شکل که شما یک عدد صحیح بزرگ که رشته می باشد را به عنوان پارامتر ورودی به این تابع می دهید و این تابع برای شما این مقدار رشته ای را به یک مقدار صحیح بزرگ

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

عددی تبدیل می نمایید. برای استفاده از این تابع باید هدر فایل <stdlib.h> به برنامه افزوده شود.

۲۴۷) در زبان C++ برای تعریف یک متغیر محلی با طول عمر محلی و معمولی از کلاس حافظه ای با نام auto استفاده میشود.

۲۴۸) البته کلاس auto کلاس پیش فرض تمامی متغیرها می باشد، به این معنا که اگر شما برای متغیرهای تعریفی در برنامه نوع کلاس را مشخص ننمایید کامپایلر به طور اتوماتیک این کلاس را برای متغیر منظور میکند، ولی اگر کلاس خاصی را نام ببرید مسلما همان کلاس در نظر گرفته خواهد شد.

۲۴۹) در زبان C++ برای ترسیم مستطیل از تابع زیر استفاده میشود:

Void far bar(int left , int top , int right , int bottom). با استفاده از این تابع شما قادر به ترسیم یک مستطیل خواهید بود که به جای پارامتر int left طول مختصات چپ ترین نقطه از این مستطیل را قرار میدهیم و به جای پارامتر int top عرض مختصات بالاترین نقطه از مستطیل را قرار میدهیم و به جای پارامتر int right طول مختصات راست ترین نقطه را قرار میدهیم و به جای پارامتر int bottom عرض مختصات پایین ترین نقطه را قرار میدهیم. بعد از اجرای تابع فوق یک مستطیل با مختصات ذکر شده برای شما ترسیم می گردد. برای استفاده از این تابع باید هدر فایل <graphics.h> به برنامه افزوده شود.

۲۵۰) در زبان C++ برای ترسیم مستطیل سه بعدی از تابع زیر استفاده میشود:

Void far bar3d(int left , int top , int right , int bottom , int depth , int topflag). با استفاده از این تابع شما قادر به ترسیم یک مستطیل خواهید بود که به جای پارامتر int left طول مختصات چپ ترین نقطه از این مستطیل را قرار میدهیم و به جای پارامتر int top عرض مختصات بالاترین نقطه از مستطیل را قرار میدهیم و به جای پارامتر int right

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

طول مختصات راست ترسن نقطه را قرار میدهیم و به جای پارامتر int bottom عرض مختصات پایین ترین نقطه را قرار میدهیم و به جای پارامتر int depth عمق (ارتفاع) ترسیم مستطیل را به پیکسل بیان میکنیم و به جای پارامتر int topflag عددی قرار میدهیم که اگر این عدد ۰ باشد بالای مستطیل مارا ترسیم نمیکند واگر غیر صفر باشد حالت سه بعدی آن را ترسیم میکند. بعد از اجرای تابع فوق یک مستطیل سه بعدی با مختصات ذکر شده برای شما ترسیم می گردد. برای استفاده از این تابع باید هدر فایل <graphics.h> به برنامه افزوده شود.

۲۵۱) تابع bcd(number) برای نمایش اعداد به صورت نماد علمی میباشد. برای استفاده از این تابع باید هدر فایل <bcd.h> به برنامه افزوده شود.

۲۵۲) تابع Bcd bcd(int x) که یک عدد صحیح را دریافت کرده و به فرمت bcd تبدیل می کند. برای استفاده از این تابع باید هدر فایل <bcd.h> به برنامه افزوده شود.

۲۵۳) تابع Bcd bcd(double x) که یک عدد اعشاری را دریافت کرده و به فرمت bcd تبدیل می کند. برای استفاده از این تابع باید هدر فایل <bcd.h> به برنامه افزوده شود.

۲۵۴) تابع Bcd bcd(double x , int decimals) که یک عدد اعشاری رادریافت کرده و سپس یک عدد صحیح، عدد اعشاری را با تعداد رقم اعشاری که عدد صحیح مشخص می کند به عدد bcd تبدیل می کند. برای استفاده از این تابع باید هدر فایل <bcd.h> به برنامه افزوده شود.

۲۵۵) تابع int bdos(int dosfun , unsigned dosdx , unsigned dosal) با استفاده از این تابع شما میتوانید توابع dos را فراخوانی کنبد. برای استفاده از این تابع باید هدر فایل <dos.h> به برنامه افزوده شود.

۲۵۶) به جای پارامتر int dosfun شماره تابعی از dos را که قصد فراخوانی داریدرا می نویسید.

۲۵۷) و به جای پارامتر unsigned dosdx مقداری را که میخواهید در رجیستر dx قرار بگیرد را قرار میدهید.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۲۵۸) به جای پارامتر unsigned dosal مقداری را که میخواهید در رجیستر al قرار بگیرد را قرار می دهید.

۲۵۹) مقدار بازگشتی نیز که یک مقدار صحیح میباشد مقداری است که در رجیستر ax قرار میگیرد.

۲۶۰) دستور break; این دستور برای شکستن و خروج از یک ساختار به کار میرود. ساختارهایی که میتوان این دستور را در آنها به کار برد ساختارهای مانند: for و while و do و switch می باشند.

۲۶۱) تابع void *calloc(size_t nitems , size_t size) برای دریافت حافظه پویا از سیستم به کار می رود. در این دستور شما با مشخص کردن تعداد و نوع حافظه مورد نیاز یک بلوک از حافظه را به صورت دستی از سیستم عامل اخذ میکنید و آن را در اختیار برنامه قرار میدهید که این مقدار از ۰ تا 64kb قابل افزایش است. برای استفاده از این تابع باید هدر فایل های <stdlib.h> یا <alloc.h> را به برنامه افزود.

۲۶۲) در پارامترهای مشخص شده به جای size_t nitems شما باید تعداد متغیرهای مورد نیاز را تعیین کنید.

۲۶۳) به جای پارامتر sie_t size نوع متغیر را مشخص نمایید تا میزان حافظه مورد نیازتان برای شما اخذ گردد.

۲۶۴) تابع double ceil(double x) یک عدد اعشاری را به عنوان پارامتر ورودی دریافت می کند و آن عدد را به سمت بالا گرد می کند. و باز هم یک عدد را از نوع اعشاری بازگشت می دهد. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۲۶۵) تابع long double ceil(long double x) یک عدد اعشاری بزرگ را به عنوان پارامتر ورودی دریافت می کند و آن عدد را به سمت بالا گرد می کند. و باز هم یک عدد را از نوع

اعشاری بزرگ بازگشت می دهد. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۲۶۶) تابع char *cgets(char *str) به شما اجازه می دهد که از ورودی یک رشته را خوانده و دخیره نمایید، به این شکل که char *str آدرس شروع رشته ورودی می باشد و خروجی این تابع نیز یک رشته می باشد که آن را در یک رشته درون برنامه ذخیره می کند. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۲۶۷) تابع int chdir(const char *path) توانایی چک کردن یک مسیر را دارد و میتواند چک کند که پارامتر ورودی به تابع آیا یک آدرس با ارزش است یا فاقد ارزش می باشد. برای استفاده از این تابع باید هدر فایل <dir.h> را به برنامه افزود.

۲۶۸) در پارامتر ورودی و به جای const char *path یک آدرس را وارد می کنید و برنامه آدرس ورودی را چک می کند.

۲۶۹) تابع دو مقدار را باز می گرداند، اگر مقدار ۰ را برگرداند به معنای صحت آدرس و در صورتی که 1- را برگرداند به این معنی که آدرس ورودی نا معتبر می باشد و شماره خطا را در errno نگهداری می کند.

۲۷۰) تابع int chmod(const char *path, int amode) برای تغییر صفت فایل به کار می رود و به این صورت به کار می رود. برای استفاده از این تابع باید هدر فایل های <io.h> و <sys\stat.h> را به برنامه افزود.

۲۷۱) به جای پارامتر ورودی const char *path آدرس یک فایل را وارد می نمایید که قرار است صفت آن تغییر یابد.

۲۷۲) به جای قسمت int amode صفت انتخابی را وارد می نمایید.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۲۷۳) صفت هایی را که می توانید به یک فایل نسبت دهید از قرار زیر می باشد:

IWRITE_: صفتی که فایل را فقط نوشتنی می کند.

S_IREAD: صفتی که فایل را فقط خواندنی می کند.

S_IREADIS_IWRITE: صفتی که فایل را هم خواندنی و هم نوشتنی می کند.

۲۷۴) این تابع پس از اجرا ۲ مقدار را بر میگرداند که مقدار ۰ به معنای اجرای موفقیت آمیز تغییر صفت و 1- به معنای اجرای ناموفق برای تغییر صفت می باشد که البته شماره خطای پیش آمده نیز در errno قابل مشاهده می باشد.

۲۷۵) تابع int chsize(int handle, long size) برای افزودن حجم به یک فایل بر اساس بایت به کار می رود. برای استفاده از این تابع باید هدر فایل <io.h> را به برنامه افزود.

۲۷۶) به جای پارامتر ورودی int handle فایلی را مشخص می کنید که قرار است تغییر حجم بر روی آن اعمال شود.

۲۷۷) به جای پارامتر ورودی long size میزان افزایش حجم فایل را بر حسب بایت می نویسیم.

۲۷۸) بعد از اجرای این دستور دو مقدار بازگشت داده می شود که اگر ۰ بازگشت داده شود به معنای اجرای موفقیت آمیز عملیات بوده است.

۲۷۹) اگر 1- بازگشت داده شود به معنای عدم موفقیت در اجرا بوده و شماره خطای مورد نظر نیز در errno ذخیره میشود.

۲۸۰) تابع دو حالت دارد یا شما اطلاعاتی دارید و میخواهید برای اطلاعات مورد نظر افزایش حجم دهید که فایل به میزان مورد نظر افزایش حجم پیدا می کند و اطلاعات را به انتهای خود می افزاید و حالت بعد این است که ما اطلاعاتی نداریم و فقط می خواهیم افزایش حجم به فایل بدهیم . برنامه کاراکترهای 0\ را به تعداد دلخواه شما به انتهای فایل می افزاید و حجم فایل را افزایش می دهد.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۲۸۱) تابع void far circle(int x, int y, int radius) برای ترسیم دایره استفاده می شود. در تابع ذکر شده باید به جای پارامتر ورودی int x طول نقطه مرکز دایره و به جای int y عرض نقطه مرکز دایره و به جای پارامتر ورودی int radius باید شعاع دایره مورد نظر را وارد نمایید. بعد از اجرای این تابع برنامه برای شما در نقطه درخواستی و با شعاع مورد نظر یک دایره ترسیم می کند. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۲۸۲) تابع void far cleardevice (void) برای پاک کردن صفحه نمایش در مود گرافیکی به کار می رود. به این شکل که بعد از فراخوانی تابع صفحه نمایش به طور کامل پاک شده و کرسر نیز در آدرس (0,0) صفحه نمایش قرار می گیرد. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۲۸۳) تابع void far clearviewport (void) برای پاک کردن صفحه نمایش کنونی در مود گرافیکی به کار می رود. به این شکل که بعد از فراخوانی تابع صفحه نمایش به طور کامل پاک شده و کرسر نیز در آدرس (0,0) صفحه نمایش قرار می گیرد. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۲۸۴) تابع clock_t clock(void) شما با استفاده از این تابع می توانید تعداد clock های خورده از زمان شروع برنامه را بدست آورید. و آن را به کاربر نمایش دهید. به طور مثال در سیستم من در هر ثانیه ۱۸ کلاک می شمارد. برای استفاده از این تابع باید هدر فایل <time.h> را به برنامه افزود.

۲۸۵) تابع void closedir(DIR *drip) شما با این تابع میتوانید پوشه مورد نظرتان در آدرس مورد نظر را ببندید. برای استفاده از این تابع باید هدر فایل <dirent.h> را به برنامه افزود.

۲۸۶) به جای پارامتر ورودی DIR *drip آدرس مورد نظر را وارد نمایید .

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۲۸۷) بعد از اجرای این تابع اگر موفقیت آمیز اجرا شود شما مقدار ۰ را دریافت می کنید و اگر با شکست مواجه شود مقدار 1- را بازگشت می دهد.

۲۸۸) تابع void far closegraph(void) برای خروج از مود گرافیکی و بازگشت به مود متنی می باشد، اجرای این تابع باعث می شود که تمامی حافظه ای که برنامه از سیستم برای گرافیک قرض گرفته را به سیستم برگرداند. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۲۸۹) تابع void clreol(void) به شما این اجازه را می دهد که از محل کرسر تا انتهای خط را پاک کنید و اطلاعات را از صفحه نمایش حذف کنید. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۲۹۰) تابع void clrscr(void) به شما این امکان را می دهد که تمام صفحه نمایش را به طور کامل پاک کنید، بعد از انجام این تابع صفحه نمایش کاملا پاک شده و سپس کرسر به (0,0) منتقل می شود. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۲۹۱) تابع complex complex(double real , double image) برای تولید یک عدد مختلط استفاده می شود. برای استفاده از این تابع باید هدر فایل <complex.h> را به برنامه افزود.

۲۹۲) به جای پارامتر ورودی double real یک عدد حقیقی را برای قسمت صحیح عدد مختلط وارد می نمایید.

۲۹۳) به جای پارامتر ورودی double image یک عدد حقیقی را برای قسمت موهومی عدد مختلط وارد کنید.

۲۹۴) تابع با دریافت این اعداد از شما خود یک عدد مختلط ایجاد میکند و خروجی تابع هم همین عدد مختلط خواهد بود. البته متغیر مقصد حتما باید یک عدد مختلط باشد.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۲۹۵) تابع double conj(complex z) به شما کمک می کند تا بتوانید یک عدد مختلط را به عنوان پارامتر ورودی دریافت کرده و سپس برای شما درخروجی قسمت های صحیح وموهومی عدد مختلط را نمایش می دهد.برای استفاده از این تابع باید ه‌در فایل <complex.h> را به برنامه افزود.

۲۹۶) دستور continue در برنامه ها در حلقه ها به کار می رود و کاربرد آن نیز به این شکل است که در صورت برقرار بودن شرط خاصی نخواهیم که تعدادی از دستورات شرط اجرا شوند، یک شرط قرار داده و بدنه آن را دستور continue قرار می دهیم، در صورت برقرار بودن شرط از ادامه اجرای دستورات منصرف شده و اجرای حلقه از سر گرفته می شود و دستورات از ابتدای حلقه شروع به اجرا شدن میکنند.

۲۹۷) تابع double cosh(double x) برای محاسبه کسینوس هاپربولیک استفاده می شود و نحوه کاربرد آن به این شکل است که شما یک عدد حقیقی را به عنوان پارامتر ورودی به برنامه میدهید و برنامه با استفاده از این تابع مقدار کسینوس هایپربولیک را محاسبه کرده و به برنامه باز می گرداند. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۲۹۸) تابع long double cosh(long double x) برای محاسبه کسینوس هاپربولیک استفاده می شود و نحوه کاربرد آن به این شکل است که شما یک عدد حقیقی بزرگ را به عنوان پارامتر ورودی به برنامه میدهید و برنامه با استفاده از این تابع مقدار کسینوس هایپربولیک را محاسبه کرده و به برنامه باز می گرداند. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۲۹۹) تابع int cputs(const char *str) برای نوشتن یک رشته در یک پنجره متنی مورد استفاده قرارمی گیرد. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۳۰۰) تابع int creattemp(char *path, int attrib) برای ساختن یک فایل temp استفاده می شود.

۳۰۱) به جای پارامتر char *path آدرس فایل مورد نظر را باید وارد کرد.

۳۰۲) و به جای پارامتر int attrib صفت فایل temp را مشخص می کنیم.

۳۰۳) به این شکل که فایل یا FA_RDONLY به معنای فقط خواندنی می باشد و یا FA_HIDDEN به معنای نامرئی می باشد و در آخر FA_SYSTEM به معنای فایل سیستمی می باشد.

۳۰۴) تابع بعد از اجرا ۲ مقدار بازگشتی بر می گرداند، که مقدار ۰ به معنای انجام موفقیت آمیز می باشد و 1- به معنای شکست در اجرای تابع می باشد که شماره خطا در errno دخیره می گردد.

۳۰۵) تابع void ctrlbrk(int (*handler)(void)) شبیه سازی فشردن کلیدهای ctrl+break را انجام می دهد که باعث می شود که اجرای برنامه شکسته می شود. و از حالت اجرا خارج می شود. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۳۰۶) تابع void delay(unsigned milliseconds) برای ایجاد تاخیر های زمانی استفاده می شود، این تابع یک مقدار را به عنوان پارامتر ورودی دریافت کرده و به جای unsigned millisecond یک عدد صحیح مثبت را قرار می دهد. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۳۰۷) دستور delete *var برای حذف حافظه های پویا که از حافظه سیستم اخذ می شود به کار می رود.

۳۰۸) به جای var* نام متغیری را که به صورت پویا از سیستم اخذ کرده اید را وارد می نمایید سپس حافظه اشغال شده آزاد می گردد و در اختیار سیستم قرار می گیرد.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۳۰۹) تابع void delline(void) در مکانی که کرسر قرار دارد خط را به طور کامل پاک می کند و خطوط پایین تر را یک خط به بالا شیفت می دهد. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۳۱۰) تابع void far detectgraph(int far *graphdriver , int far *graphmode) برای چک کردن حالت گرافیکی کارت گرافیک به کار می رود. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۳۱۱) دو پارامتر را دریافت میکند.که فقط کافیست دو متغیر با نام های gdriver و gmode از نوع int تعریف کرده و سپس در تابع جایگذاری می کنیم. در صورت آماده بودن گرافیک برنامه از مود متنی به مود گرافیکی وارد می شود.

۳۱۲) تابع double difftime(time_t time2 , time_t time1) برای محاسبه اختلاف زمانی مابین دو زمان time2 و time1 می باشد. به این شکل که شما زمان اولیه و ثانویه را وارد می نمایید و تابع برای شما اختلاف زمان را برحسب ثانیه به شما بر میگرداند. برای استفاده از این تابع باید هدر فایل <time.h> را به برنامه افزود.

۳۱۳) تابع void disable (void) برای غیر فعال کردن اجرای وقفه ها به کار می رود، این تابع زمانی به کار میرود که شما بخواهید در حین اجرای برنامه تان اجرای تمامی وقفه ها را غیر فعال کنید. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۳۱۴) تابع int dosexterr(struct DOSERROR *eblkp) برای بدست آوردن اطلاعات کامل در مورد خطاهای پیش آمده در برنامه به کار می رود. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۳۱۵) به جای پارامتر ورودی که در تابع قرار دارد با نام struct DOSERROR *eblkp آدرس یک ساختمان از نوع مورد نظر را وارد می کنبد سپس می توانید از ساختمان ذکر شده تمامی اطلاعات خطای پیش آمده از قبیل Extended error و Class و Action و Error locus را بدست آورید و تصمیم درست را در مواجهه با خطای پیش آمده بگیرید.

۳۱۶) تابع long dostounix (struct date *d, struct dostime *t) برای تبدیل تاریخ از dos به unix به کار می رود. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۳۱۷) عملکرد این تابع به این شکل است که با استفاده از توابع getdate و gettime ابتدا تاریخ و زمان سیستم را در dos بدست می آوریم، سپس تابع مقدار ثانیه های گذشته از زمان 00:00:00 در تاریخ تاریخ 1970/January/1 تا زمان حال را محاسبه کرده و سپس آن را به فرمت تاریخ قابل خواندن در unix تبدیل می کند.

۳۱۸) تابع void far drawpoly(int numpoints , int far *polypoints) برای ترسیم یک شکل چند ضلعی نامنظم به کار می رود، برای استفاده از این تابع باید پارامترها را به شکل زیر مقدار دهی کرد. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۳۱۹) به جای پارامتر int numpoints باید تعداد نقاط شکل مورد نظر ترسیمی را وارد نمایید، و به جای پارامتر int far *polypoints باید نقاطی را که از قبل تهیه کرده اید که دقیقا باید دو برابر پارامتر قبل باشند وارد می نمایید. بعد از اجرای تابع فوق یک چند ضلعی با نقاطی که مشخص کرده ای برایتان ترسیم می شود.

۳۲۰) تابع void enable (void) جهت فعالسازی وقفه ها به کار می رود. هنگامی که بخواهیم وقفه ها در حین اجرای برنامه ما قابلیت اجرا را داشته باشند این تابع را فراخوانی میکنیم و

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

بعد از آن میتوانیم از وقفه های از پیش تعیین شده استفاده کنیم برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۳۲۱) اگر در برنامه ای با ERROR مواجه شوید و برنامه نتواند به کار خود ادامه دهد اگر بتوانید با Extern int errno کار کنید مسلما خواهید توانست خطا را مدیریت کنید و از پایان یافتن برنامه بدون خواست کاربر جلوگیری کنید، این عمل یکی از اعمال یک برنامه‌نویس حرفه ای می باشد. شما می توانید شماره خطای پیش آمده را در متغیر فوق الذکر پیدا کرده و عملیات خاص آن خطا را انجام دهید. برای استفاده از این تابع باید هدر فایل های <errno.h> یا <stddef.h> یا <stdlib.h> را به برنامه افزود.

۳۲۲) تابع void exit (int status) برای پایان دادن به اجرای برنامه به صورت دستی و سریع استفاده می شود.

۳۲۳) هنگامی که تابع اجرا می شود سه کار را انجام می دهد:

۱)بستن تمامی فایل های باز وابسته به برنامه.

۲)نوشتن بافرها در خروجی هایشان.

۳)تمامی رجیسترهای خروج را برای اتمام کار برنامه فراخوانی می کند.

۳۲۴) اگر به جای پارامتر ورودی int status عدد ۰ را قرار دهیم برنامه به سیستم عامل اعلام می کند که برنامه به صورت کاملا نرمال پایان یافته است و اگر مقداری غیر ۰ بدهیم شماره خطای رخ داده را به سیستم عامل اعلام میکنیم و به سیستم عامل اعلام می کنیم که برنامه به صورت غیر نرمال پایان یافته است.

۳۲۵) تابع double exp (double x) جهت محاسبه e^x به کار می رود، در تابع به جای پارامتر ورودی double x باید یک عدد اعشاری را وارد نمایید، سپس تابع مقدار e^x را

محاسبه کرده و به صورت یک مقدار double بازگشت می دهد. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۳۲۶) تابع long double exp (long double x) جهت محاسبه e^x به کار می رود، در تابع به جای پارامتر ورودی long double x باید یک عدد اعشاری بزرگ را وارد نمایید، سپس تابع مقدار e^x را محاسبه کرده و به صورت یک مقدار long double بازگشت می دهد. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۳۲۷) تابع complex exp (complex z) جهت محاسبه e^z به کار می رود، در تابع به جای پارامتر ورودی complex z باید یک عدد مختلط را وارد نمایید، سپس تابع مقدار e^z را محاسبه کرده و به صورت یک مقدار مختلط بازگشت می دهد. برای استفاده از این تابع باید هدر فایل <complex.h> را به برنامه افزود.

۳۲۸) تابع double fabs(double x) مقدار قدر مطلق را بر می گرداند، به این صورت که شما به جای پارامتر ورودی double x یک عدد اعشاری را وارد کرده و تابع قدر مطلق عدد را محاسبه کرده و به صورت عدد اعشاری بر می گرداند. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۳۲۹) تابع long double fabs(long double x) مقدار قدر مطلق را بر می گرداند، به این صورت که شما به جای پارامتر ورودی long double x یک عدد اعشاری بزرگ را وارد کرده و تابع قدر مطلق عدد را محاسبه کرده و به صورت عدد اعشاری بزرگ بر می گرداند. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۳۳۰) تابع complex fabs(complex zx) مقدار قدر مطلق را بر می گرداند، به این صورت که شما به جای پارامتر ورودی complex z یک عدد مختلط را وارد کرده و تابع قدر مطلق عدد

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

را محاسبه کرده و به صورت عدد مختلط بر می گرداند. برای استفاده از این تابع باید هدر فایل <complex.h> را به برنامه افزود.

۳۳۱) تابع void far *farcalloc(unsigned long nunits , unsigned long unitsz) برای گرفتن حافظه پویا از سیستم در حین اجرای برنامه به کار میرود. برای استفاده از این تابع باید هدر فایل <alloc.h> را به برنامه افزود.

۳۳۲) به جای پارامتر ورودی unsigned long nunits تعداد متغیرهایی را که میخواهیم وارد می کنیم.

۳۳۳) به جای پارامتر ورودی unsigned long unitsz میزان حافظه ای که برای ەر متغیر نیاز داریم را مشخص میکنیم. که در واقع می توان گفت نوع متغیر هایی را که میخواهیم از حافظه اخذ کنیم مشخص می کند.

۳۳۴) اگر تابع به صورت موفقیت آمیز اجرا شود تابع یک اشاره گر بر میگرداند و اگر اجرای تابع با شکست مواجه شود مقدار null را بر میگرداند.

۳۳۵) تابع void farfree(void far *block) برای آزاد کردن حافظه های پویا که از سیستم اخذ شده است استفاده می شود. برای استفاده از این تابع باید هدر فایل <alloc.h> را به برنامه افزود.

۳۳۶) به جای void far *block باید نام اشاده گری که بلوک حافظه از آن آدرس آغاز شده است را بنویسیم.

۳۳۷) بعد از اجرای این تابع میزان حافظه ای که برنامه از سیستم اخذ کرده بود را به سیستم بر میگرداند.

۳۳۸) تابع int fclose(FILE *stream) برای بستن فلیل باز در حال استفاده در برنامه می باشد. برای استفاده از این تابع باید هدر فایل <stdio.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۳۳۹) به جای پارامتر ورودی تابع FILE *stream نام فایل مورد استفاده در برنامه را تایپ کنید.

۳۴۰) بعد از اجرای تابع فوق فایل مورد نظر بسته خواهد شد. این تابع دو مقدار بازگشتی دارد ، اگر ۰ برگرداند به این معنا می باشد که عملیات بستن فایل با موفقیت انجام شده و اگر عملیات بستن فایل با شکست مواجه شود مقدار EOF را بر میگرداند.

۳۴۱) تابع int fcloseall(void) برای بستن تمامی فایل های باز در حال استفاده در برنامه می باشد،با استفاده از این تابع به طور همزمان تمامی فایل های باز مورد استفاده در برنامه بسته خواهند شد. برای استفاده از این تابع باید هدر فایل <stdio.h> را به برنامه افزود.

۳۴۲) این تابع دو مقدار بازگشتی دارد ، اگر ۰ برگرداند به این معنا می باشد که عملیات بستن فایل با موفقیت انجام شده و اگر عملیات بستن فایل با شکست مواجه شود مقدار EOF را بر میگرداند.

۳۴۳) تابع int feof (FILE *stream) برای چک کردن رسیدن به انتهای فایل استفاده می شود. برای استفاده از این تابع باید هدر فایل <stdio.h> را به برنامه افزود.

۳۴۴) به جای پارامتر ورودی FILE *stream نام فایلی را که میخواهید تست کنید را می نویسید، سپس مقدار بازگشتی فایل را چک می کنیم، اگر مقدار غیر صفر بازگشت داده شود به معنای آخر فایل است و اگر مقدار صفر بازگشت داده شود به معنای نرسیدن به انتهای فایل است.

۳۴۵) تابع int ferror (FILE *stream) برای بدست آوردن خطای احتمالی در مورد فایل می باشد. برای استفاده از این تابع باید هدر فایل <stdio.h> را به برنامه افزود.

۳۴۶) به جای پارامتر ورودی نام فایلی را که می خواهید بدانید برایش خطایی رخ داده یا خیر را وارد می کنید.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۳۴۷) مقدار بازگشتی را برای تابع چک میکنید، اگر مقدار بازگشتی ۰ بود به این معناست که هیچ خطایی رخ نداده است و اگر مقدار بازگشتی غیر صفر بود به این معنا می باشد که خطایی رخ داده و مقدار بازگشتی کد خطای اتفاق افتاده می باشد.

۳۴۸) تابع int fflush(FILE *stream) برای تخلیه بافر برنامه در فایل مورد نظر می باشد. برای استفاده از این تابع باید هدر فایل <stdio.h> را به برنامه افزود.

۳۴۹) در برنامه ها یی که از فایل استفاده می شود و اطلاعات در فایل ذخیره می شود و از فایل اطلاعات می خواند اگر بخواهیم کاراکتر به کاراکتر اطلاعات را در فایل نوشته و یا بخوانیم زمان زیادی از cpu هدر می رود، برای اینکه این زمان را کاهش دهیم ، یک حافظه واسط به نام بافر را قرار می دهیم تا اطلاعات را در آن بریزیم وبعد اطلاعات را از بافر درون فایل ذخیره میکنیم، این تابع به شما کمک می کند که در هنگام خاتمه دادن به کار برنامه تمامی اطلاعات بافر را درون فایل ذخیره کنید.تابع دو مقدار بازگشتی دارد، اگر مقدار ۰ را برگرداند به این معنا می باشد که تابع کار خود را با موفقست به پایان رساتده است و اگر کار تابع با شکست مواجه شود مقدار EOF را برمی گرداند.

۳۵۰) تابع int fgetc(FILE *stream) برای خواندن یک کاراکتر از فایل به کار می رود. برای استفاده از این تابع باید هدر فایل <stdio.h> را به برنامه افزود.

۳۵۱) این تابع از فایلی که نام آن را به جای پارامتر FILE *stream نوشته اید کاراکتری را که اشاره گر فایل بر روی آن متوقف شده را می خواند، اگر عملیات خواندن کاراکتر از فایل با موفقیت انجام گیرد، تابع کد اسکی کاراکتری را که خوانده است را برمیگرداند، و اگر عملیات خواندن از فایل با شکیت مواجه شود ویا به انتهای فایل برسیم تابع مقدار EOF را بر می گرداند.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۳۵۲) تابع long filelength(int handle) حجم فایل ورودی را بر می گرداند. برای استفاده از این تابع باید هدر فایل <io.h> را به برنامه افزود.

۳۵۳) به این شکل که به جای پارامتر ورودی int handle شماره فایل مورد نظر را وارد می نمایید، سپس تابع حجم فایل را به بایت بر می گرداند. اگر تابع کار خود را با موفقیت انجام دهد مقدار بازگشتی آن حجم فایل می باشد، و اگر با خطا مواجه شود مقدار 1- را برمی گرداند.

۳۵۴) تابع int fileno(FILE *stream) شماره فایلی را که شما به جای پارامتر ورودی FILE *stream وارد کر ده اید را بر می گرداند، برای استفاده از این تابع باید هدر فایل<stdio.h> را به برنامه افزود.

۳۵۵) تابع double floor(double x) یک مقدار اعشاری را دریافت کرده سپس عدد را به سمت پایین گرد می کند و بر می گرداند، برای استفلده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۳۵۶) تابع long double floorl(long double x) یک مقدار اعشاری بزرگ را دریافت کرده سپس عدد را به سمت پایین گرد می کند و بر می گرداند، برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۳۵۷) تابع int flushall(void) برای تخلیه بافر برنامه در فایل مورد نظر می باشد. دربرنامه ها یی که از فایل استفاده می شود و اطلاعات در فایل ذخیره می شود و از فایل اطلاعات می خواند اگر بخواهیم کاراکتر به کاراکتر اطلاعات را در فایل نوشته و یا بخوانیم زمان زیادی از cpu هدر می رود، برای اینکه این زمان را کاهش دهیم ، یک حافظه واسط به نام بافر را قرار می دهیم تا اطلاعات را در آن بریزیم وبعد اطلاعات را از بافر درون فایل ذخیره میکنیم، این تابع به شما کمک می کند که در هنگام خاتمه دادن به کار برنامه تمامی اطلاعات بافر های فایل های

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

درون برنامه را در فایل ذخیره می کند مقدار بازگشتی تابع نمایانگر تعداد فایل های درون برنامه می باشد. برای استفاده از این تابع باید هدر فایل <stdio.h> را به برنامه افزود.

۳۵۸) تابع double fmod(double x , double y) برای محاسبه باقیمانده تقسیم x بر y استفاده می شود، برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۳۵۹) تابع long double fmodl(long double x ,long double y) برای محاسبه باقیمانده تقسیم x بر y استفاده می شود، برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۳۶۰) تابع FILE *fopen(const char *filename , const char *mode) برای باز کردن فایل استفاده می شود. برای استفاده از این تابع باید هدر فایل <stdio.h> را به برنامه افزود.

۳۶۱) در تابع به جای پارامتر ورودی const char *filename نام فایل را می نویسید، و به جای پارامتر ورودی const char *mode نوع بازکردن فایل را انتخاب می کنید.

۳۶۲) مقدار بازگشتی تابع آدرس یک فایل می باشد.انواع باز کردن فایل حالتهای زیر می باشد:

r-w-a-r+-w+-a+-etc...

تابع دو مقدار بازگشتی دارد، اگر مقدار بازگشتی یک آدرس باشد نشان دهنده موفقیت آمیز بودن اجرای تابع می باشد و اگر اجرای تابع با شکست مواجه شود مقدار بازگشتی null خواهد بود.

۳۶۳) تابع int fputc(int c , FILE *stream) برای نوشتن یک کاراکتر در فایل استفاده می شود. برای استفاده از این تابع باید هدر فایل <stdio.h> را به برنامه افزود.

۳۶۴) کاراکتری را که کد اسکی آن را به جای پارامتر int c قرار داده ایم را در فایلی که به جای پارامتر FILE *stream نوشته ایم ذخیره می شود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۳۶۵) تابع دو مقدار بازگشتی دارد، اگر تابع کار خود را با موفقیت انجام دهد کاراکتری را قصد نوشتن در فایل داریم را بر می گرداند و اگر با شکست مواجه شود مقدار EOF را بر می گرداند.

۳۶۶) تابع int getch(void) برای دریافت یک کاراکتر از ورودی صفحه کلید می باشد،تابع به این صورت عمل می کند که برنامه را در حالت توقف نگه می دارد و تا زمانی که کاربر یک کلید را نفشارد برنامه به کار خود ادامه نمی دهد. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۳۶۷) هنگامی که یک کلید فشرده شود کد اسکی آن را بر میگرداند.این تابع مزیتی که دارد این است که کاراکتری را که از ورودی می خواند را در صفحه نمایش نشان نمی دهد.

۳۶۸) تابع int getchar(void) برای دریافت یک کاراکتر از ورودی صفحه کلید می باشد،تابع به این صورت عمل می کند که برنامه را در حالت توقف نگه می دارد و تا زمانی که کاربر یک کلید را نفشارد برنامه به کار خود ادامه نمی دهد. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۳۶۹) هنگامی که یک کلید فشرده شود کد اسکی آن را بر میگرداند.این تابع مزیتی که دارد این است که کاراکتری را که از ورودی می خواند را در صفحه نمایش نشان می دهد.

۳۷۰) تابع int getche(void) برای دریافت یک کاراکتر از ورودی صفحه کلید می باشد،تابع به این صورت عمل می کند که بنامه را در حالت توقف نگه می دارد و تا زمانی که کاربر یک کلید را نفشارد برنامه به کار خود ادامه نمی دهد. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۳۷۱) هنگامی که یک کلید فشرده شود کد اسکی آن را بر میگرداند.این تابع مزیتی که دارد این است که کاراکتری را که از ورودی می خواند را در صفحه نمایش نشان می دهد.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۳۷۲) تابع int far getcolor(void) برای بدست آوردن رنگ تنظیم شده می باشد، و به این شکل کار می کند که در مود گرافیکی کد رنگ سیستم را دریافت کرده و به صورت عدد صحیح بر میگرداند. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۳۷۳) تابع void getdate(struct date *datep) برای دریافت تاریخ کنونی سیستم می باشد. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۳۷۴) به جای پارامتر ورودی struct date *datep یک اشاره گر از نوع ساختمان تاریخ قرار دهیم تا تابع برای ما اطلاعات تاریخ را از سیستم دریافت کرده و برای شما در متغیر ذخیره کند.

۳۷۵) تابع int getdisk(void) برای بدست آوردن درایو جاری می باشد. برای استفاده از این تابع باید هدر فایل <dir.h> را به برنامه افزود.

۳۷۶) این تابع یک مقدار بازگشتی از نوع int دارد که در واقع شماره درایو جاری می باشد که از شماره۱ شروع می شود و بعد ۲ و ...

۳۷۷) تابع void far getimage(int left,int top,int right,int bottom, void far *bitmap) برای ذخیره یک عکس از صفحه نمایش در حافظه سیستم به کار می رود. به این شکل که تصویر را بایت به بایت در حافظه ذخیره می کند. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۳۷۸) پارامترهای int left و int top گوشه بالا سمت چپ تصویر را در صفحه نشان می دهد و پارامترهای int right و int bottom گوشه پایین سمت راست تصویر را در صفحه نشان می دهد و پارامتر void far *bitmap اشاره گری به آدرسی است که قرار است تصویر در آن ذخیره شود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۳۷۹) تابع int far getmaxcolor(void) در مود گرافیکی بالاترین رنگ معتبر را به کاربر نمایش می دهد، که این مقدار بازگشتی یک عدد خواهد بود که معمولا هم این عدد برابر با ۱۵ می باشد. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۳۸۰) تابع int far getmaxmode(void) در مورد گرافیکی بالاترین مود گرافیکی در درایو جاری را بر می گرداند، کمترین مقدار ۰ خواهد بود ولی برای بدست آوردن بالاترین مقدار باید از این تابع استفاده شود. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۳۸۱) تابع int far getmaxx(void) در مورد گرافیکی بالاترین مقدار x را بر میگرداند، و مشخص کننده تعداد پیکسل های صفحه نمایش (ستون) در مود گرافیکی می باشد. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۳۸۲) تابع int far getmaxy(void) در مورد گرافیکی بالاترین مقدار y را بر میگرداند، و مشخص کننده تعداد پیکسل های صفحه نمایش (سطر) در مود گرافیکی می باشد. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۳۸۳) تابع char * far getmodename(int mode_number) در مود گرافیکی با دریافت شماره مود گرافیکی در حال استفاده به جای پارامتر ورودی تابع ، نام مود در حال استفاده متناظر با شماره ورودی را بر می گرداند. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۳۸۴) تابع void far getmoderange(int graphdriver , int far * lomode , int far *himode) در مود گرافیکی رنج مودهای گرافیکی معتبر در درایور گرافیکی را بر می گرداند. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۳۸۵) به این شکل که شما به جای پارامتر ورودی int graphdriver یک مقدار را وارد می کنید: اگر 1- را قرار دهید تابع به جای پارامتر های خروجی int far *lomode پایین ترین مود گرافیکی و در پارامتر خروجی int far *himode بالاترین مود گرافیکی را قرار می دهد. و اگر شما به جای پارامتر ورودی int graphicdriver شماره یک درایو گرافیکی معتبر را وارد نمایید، تابع به جای دو پارامتر خروجی int far *lomode و int far *himode مقدار 1- را قرار می دهد. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۳۸۶) تابع void far getpalette(struct palettetype far *palette) در مود گرافیکی جعبه رنگ مورد استفاده در سیستم را بدست می آورد. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۳۸۷) به این شکل که شما تابع را با پارامتر خروجی Struct palettetype far *palette یک ساختمان که از نوع جعبه رنگ تعریف کرده ایم قرار میدهیم و تابع برای ما رنگهای موجود در سیستم را در این ساختمان لیست می کند. و ما میتوانیم به آن دست یابی داشته باشیم.

۳۸۸) تابع int far getpalettesize(void) در مود گرافیکی برای بدست آوردن سایز جعبه رنگ سیستم در مود گرافیکی خاص به کار می رود، مقدار بازگشتی این تابع یک عدد می باشد که نشانگر سایز جعبه رنگ می باشد، برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۳۸۹) تابع char *getpass(const char *prompt) برای خواندن پسورد از ورودی به کار می رود. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۳۹۰) به این شکل که شما به جای پارامتر خروجی const char* prompt یک رشته ثابت را برای نمایش پیغام به کاربر وارد می کنید و خروجی تابع که رشته می باشد را می توانید از پارامتر خود تابع دریافت کنید.

۳۹۱) تابع getpid( ) برای بدست آوردن کد پردازشی برنامه مورد نظر در سیستم می باشد، برای استفاده از این تابع باید هدر فایل <process.h> را به برنامه افزود.

۳۹۲) تابع unsigned far getpixel(int x , int y) در مود گرافیکی برای بدست آوردن کد رنگ یک پیکسل به کار می رود. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۳۹۳) به این شکل که شما به جای پارامتر های ورودی int x طول نقطه مورد نظر و به جای int y عرض نقطه مورد نظر را وارد می کنید و تابع برای شما یک مقدار عددی به عنوان کد رنگ نقطه مورد نظر را بر می گرداند.

۳۹۴) تابع unsigned getpsp (void) برای بدست آوردن پریفیک سگمنت سیستم به کار می رود. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۳۹۵) تابع char * gets(char *s) برای خواندن یک رشته از ورودی صفحه کلید به کار می رود. برای استفاده از این تابع باید هدر فایل <stdio.h> را به برنامه افزود.

۳۹۶) شما باید یک متغیر از نوع رشته را در برنامه تعریف کنید و نام آن را به جای پارامتر خروجی char *s قرار دهید تا برنامه رشته را از ورودی صفحه کلید خوانده و آن را در متغیر مورد نظر ذخیره کند.

۳۹۷) تابع int gettext(int left, int top, int right, int bottom, void *destin) برای خواندن و ذخیره کردن متن از صفحه نمایش در بافر می باشد. برای استفاده از این تابع باید هدر فایل <stdio.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۳۹۸) به این شکل که شما صفحه ای را که می خواهید به آن اندازه متن ذخیره گردد را انتخاب می کنید، به طور مثال اگر به جای پارامترهای int left عدد ۱ و به جای Int top عدد ۱ و به جای int right عدد ۸۰ و به جای int bottom عدد ۲۵ و به جای پارامتر خروجی void destin* بافری که اطلاعات را در آن ذخیره میکنیم قرار می دهیم.

۳۹۹) تابع void gettime (struct time *timep) برای گرفتن ساعت کنونی سیستم و ذخیره آن در پارامتر خروجی struct time *timep می باشد. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۴۰۰) تابع int far getx(void) در مود گرافیکی برای بدست آوردن طول موقعیت جاری در صفحه نمایش کاربرد دارد. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۴۰۱) تابع int far gety(void) در مود گرافیکی برای بدست آوردن عرض موقعیت جاری در صفحه نمایش کاربرد دارد. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۴۰۲) تابع struct tm *gmtime(const time_t *timer) برای تبدیل زمان کنونی سیستم به زمان کنونی گرینویچ استفاده می شود. برای استفاده از این تابع باید هدر فایل <time.h> را به برنامه افزود.

۴۰۳) به این صورت که زمان کنونی سیستم را با استفاده از پارامتر خروجی Const time_t timer* دریافت کرده و از خروجی تابع زمان تبدیل شده به گرینویچ را دریافت کرده و ذخیره می کنیم.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۴۰۴) تابع void gotoxy(int x,int y) برای حرکت در صفحه نمایش و پرش از موقعیتی به موقعیت دیگر استفاده می شود. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۴۰۵) در پارامتر های ورودی به جای int x طول نقطه مقصد و به جای پارامتر ورودی int y عرض نقطه مقصد را وارد کنید.

۴۰۶) تابع void far graphdefault(void) در مود گرافیکی برای بازگرداندن کلیه تغییرات در مود گرافیکی به حالت پیش فرض می باشد. بعد از اجرای تابع فوق تمامی تنظیمات دست خورده به حالت پیش فرض باز می گردد. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۴۰۷) تابع char *far grapherrormsg(int errorcode) در مود گرافیکی برای نمایش پیغام خطای به وجود آمده استفاده میشود. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۴۰۸) به این شکل که در پارامتر ورودی به جای int errorcode شماره خطای پیش آمده را وارد می کنیم تا تابع بتواند پیغام خطای مورد نظر را نمایش دهد.

۴۰۹) تابع int far graphresult(void) در مود گرافیکی برای بدست آوردن خطای بوجود آمده استفاده می شود، به این شکل که بعد از اجرای این تابع شما می توانید از مقدار بازگشتی تابع که به صورت یک عدد صحیح می باشد استفاده کنید. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۴۱۰) تابع handler (int errval , int ax , int bp , int si) برای تشخیص خطای پیش آمده در سخت افزارهای سیستم استفاده می شود.برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۴۱۱) به جای پارامتر ورودی int errval کد خطایی که در رجیستر DI قرار دارد ریخته می شود.

۴۱۲) به جای پارامتر int ax دو مقدار قرار می گیرد که اگر خطایی رخ داده باشد کد خطایی که در رجیستر ax قرار دارد و اگر خطایی رخ نداده باشد مقدار 1- قرار می گیرد.

۴۱۳) به جای پارامتر ورودی int bp مقداری که در رجیستر bp وجود دارد قرار می گیرد.

به جای پارامتر ورودی int si مقداری که در رجیستر si است قرار میگیرد.

و مقدار بازگشتی این تابع هم یکی از سه مقدار زیر خواهد بود:

مقدارهای ۰ برای ignore کردن خطا و یا ۱ برای retry کردن خطا و مقدار ۲ برای abort کردن برنامه به خاطر پیش آمدن خطا.

۴۱۴) تابع void harder(int (*handler)( )) برای مدیریت خطا های سخت افزاری پیش آمده در برنامه به کار می رود. به این شکل که شما به جای پارامتر ورودی int (*handler)( ) از نکته ۱۴۵ استفاده می کنید تا بتوانید خطا را کشف کنید و با استفاده از این تابع آن را مدیریت کنید. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۴۱۵) تابع void hardresume(int axert) برای مدیریت کردن خطاهای سخت افزاری مرتبط در حین اجرای برنامه می باشد، شما با مقدار دادن به پارامتر ورودی int axret خطا را به نحوی که میخواهید مدیریت می کنید. مقدارهای معتبر برای این پارامتر به شکل زیر می باشد:

۴۱۶) _HARDERR_ABOR برنامه در حال اجرا را abort می کند.

۴۱۷) _HARDERR_IGNORE از خطای پیش آمده در برنامه چشم پوشی می کند.

۴۱۸) _HARDERR_RETRY برای دست یابی به سخت افزار مورد نیاز در برنامه که با خطا مواجه شده است دوباره تلاش می کند.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۴۱۹) _HARDERR_FAIL پردازش مورد نظر را که درخواست دسترسی به منبع سخت افزاری را داشته است از حالت پردازش خارج می کند.

۴۲۰) تابع void highvideo(void) برای نمایش متن با حداکثر روشنایی استفاده می شود.جهت مشاهده تغییرات این تابع بر روی متن مورد نظر قبل از چاپ متن بر روی صفحه نمایش این تابع را فراخوانی کنید و سپس در دستور بعد متن را چاپ کنید. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۴۲۱) تابع double hypot(double x , double y) برای محاسبه وتر در یک مثلث قائم الزاویه به کار میرود. به این شکل که شما دو ضلع زاویه قائمه را به تابع می دهید و تابع برای شما وتر را محاسبه کرده و باز می گرداند. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۴۲۲) تابع long double hypot(long double x ,long double y) برای محاسبه وتر در یک مثلث قائم الزاویه به کار میرود. به این شکل که شما دو ضلع زاویه قائمه را به تابع می دهید و تابع برای شما وتر را محاسبه کرده و باز می گرداند. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۴۲۳) تابع double image(complex z) برای بدست آوردن ضریب قسمت موهومی یک عدد مختلط به کار می رود. برای استفاده از این تابع باید هدر فایل <complex.h> را به برنامه افزود.

۴۲۴) شما باید به جای عبارت complex z یک عدد مختلط را وارد نمایید و از خروجی تابع ضریب قسمت موهومی عدد مختلط را دریافت نمایید.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۴۲۵) تابع unsigned far imagesize (int left, int top, int right, int bottom) حجم سایز عکس انتخابی در آدرس تعیین شده را بر میگرداند. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۴۲۶) به جای پارامترهای ورودی int left و int top باید آدرس گوشه بالا سمت چپ تصویر را وارد نمایید.

۴۲۷) به جای پارامترهای int right و int bottom آدرس گوشه پایین سمت راست تصویر را وارد می نمایید.

۴۲۸) تابع کار خود را آغاز می کند و در صورت موفقیت آمیز بودن عملیات مقدار بازگشتی تابع حجم فایل می باشد که حداکثر باید تا 64kb باشد و اگر هم تابع با خطا مواجه شود ویا حجم تصویر بیشتر باشد مقدار 1- را به عنوان مقدار بازگشتی به شما بر میگرداند.

۴۲۹) تابع void far initgraph(int far *graphdriver,int far *graphmode , char far *pathdriver)

برای نصب درایور گرافیکی از این تابع استفاده می شود . برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۴۳۰) به جای پارامتر ورودی اول شماره درایور گرافیکی که می خواهیم از آن استفاده کنیم را قرار می دهیم .

۴۳۱) به جای پارامتر ورودی دوم شماره مود گرافیکی که می خواهیم از آن استفاده کنیم را قرار می دهیم .

۴۳۲) به جای پارامتر ورودی سوم که همان پارامتر آخر می باشد باید آدرس درایور گرافیکی که برنامه باید فایل های (*.bgi) را از آن بخواند وارد کنید.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۴۳۳) در صورتی که برنامه بدون بروز خطا بتواند مود گرافیکی را نصب کند و به مود گرافیکی switch کند تابع مقدار ۰ را به برنامه بازگشت می دهد و اگر تابع با خطا مواجه شود یکی از مقدارهای 2- یا 3- یا 4- و یا 5- را بر میگرداند.

۴۳۴) تابع int inp(unsigned portid) این قابلیت را به برنامه شما می افزاید که شما بتوانید یک بایت را از پورتی که آدرس آن را به جای پارامتر ورودی نوشته اید بخواند. و مقداری را که از پورت ورودی خوانده شده را میتوانید از تابع بخوانید و مورد استفاده قرار دهید. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۴۳۵) تابع int inport(int portid) این قابلیت را به برنامه شما می افزاید که شما بتوانید یک کلمه را (دوبایت) از پورتی که آدرس آن به جای پارامتر ورودی نوشته اید بخواند. و مقداری را که از پورت ورودی خوانده شده را می توانید از تابع بخوانید و مورد استفاده قرار دهید. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۴۳۶) تابع unsigned char inportb(int portid) این قابلیت را به برنامه شما می افزاید که شما بتوانید یک بایت را از پورتی که آدرس آن به جای پارامتر ورودی نوشته اید بخواند. و مقداری را که از پورت ورودی خوانده شده را می توانید از تابع بخوانید و مورد استفاده قرار دهید. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۴۳۷) تابع unsigned inpw(unsigned portid) این قابلیت را به برنامه شما می افزاید که شما بتوانید یک کلمه را (دوبایت) از پورتی که آدرس آن به جای پارامتر ورودی نوشته اید بخواند. و مقداری را که از پورت ورودی خوانده شده را می توانید از تابع بخوانید و مورد استفاده قرار دهید. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۴۳۸) تابع void insline (void) برای ایجاد یک سطر خالی در مکانی که کرسر قرار دارد استفاده می شود. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۴۳۹) تابع int far installuserfont(char far *name) برای نصب فونت های کاربر در سیستم میزبان توسط برنامه استفاده می شود. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۴۴۰) به این شکل که در پارامتر ورودی آدرس فونت را وارد می کنید. تابع بعد از اجرا درصورت موفقیت آمیز بودن عملیات شماره فونت را بر میگرداند و در صورتی که با شکست مواجه شود 11- را بر میگرداند. با استفاده از این تابع شما می توانید همزمان تا ۲۰ فونت را نصب کنید.

۴۴۱) تابع int int86(int intno , union REGS *inregs , union REGS *outregs) برای اجرای وقفه های سیستم های مبتنی بر ۸۰۸۶ به کار می رود. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۴۴۲) به این شکل که شما باید در پارامتر ورودی اول شماره وقفه مورد نظر را وارد نمایید و در پارامتر دوم برنامه اطلاعاتی را که قبل از اجرای وقفه نیاز دارد می خواند و ذخیره می کند و در پارامتر سوم نیز اطلاعاتی که بعد از اجرای وقفه وجود دارند در آن ذخیره می شوند.

۴۴۳) تابع int int86x(int intno , union REGS *inregs , union REGS *outregs , struct SREGS *segregs) برای اجرای وقفه های مبتنی بر ۸۰۸۶ به کار می رود. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۴۴۴) به این شکل که شما باید در پارامتر ورودی اول شماره وقفه مورد نظر را وارد نمایید و در پارامتر دوم برنامه اطلاعاتی را که قبل از اجرای وقفه نیاز دارد می خواند و ذخیره می کند و در پارامتر سوم نیز اطلاعاتی که بعد از اجرای وقفه وجود دارند در آن ذخیره می شوند. در پارامتر سوم هم برنامه اطلاعات رجیستر DS را ذخیره می کند که اطلاعات سگمنت رجیستر می باشد.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۴۴۵) تابع int intdos(union REGS *inregs , union REGS *outregs) برای اجرای وقفه های dos به کار می رود. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۴۴۶) به این شکل که شما باید در پارامتر اول نوع و شماره وقفه را مشخص نمایید و سپس بعد از اجرای وقفه برنامه اطلاعات بعد از اجرای وقفه را در پارامتر دوم ذخیره می نماید.

۴۴۷) تابع int intdosx(union REGS *inregs , union REGS *outregs , struct SREGS *segreg) برای اجرای وقفه های dos به کار می رود. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۴۴۸) به این شکل که شما باید در پارامتر اول نوع و شماره وقفه را مشخص نمایید و سپس بعد از اجرای وقفه برنامه اطلاعات بعد از اجرای وقفه را در پارامتر دوم ذخیره می نماید. و در پارامتر سوم هم اطلاعات سگمنت DS ذخیره می شود.

۴۴۹) تابع div_t div(int numer , int denom) برای تقسیم کردن دو عدد بر هم و بدست آوردن خارج قسمت و باقیمانده تقسیم استفاده می شود. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۴۵۰) به این صورت که به جای پارامتر اول عددی را که می خواهیم تقسیم کنیم و به جای پارامتر دوم عددی را که میخواهیم عدد اول را بر آن تقسیم کنیم می نویسیم سپس تابع عدد اول را بر عدد دوم تقسیم کرده و خارج قسمت و باقیمانده را در متغیرس از نوع div_t ذخیره میکند.

۴۵۱) تابع ldiv_t ldiv(long int numer ,long int denom) برای تقسیم کردن دو عدد بر هم و بدست آوردن خارج قسمت و باقیمانده تقسیم استفاده می شود. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۴۵۲) به این صورت که به جای پارامتر اول عددی را که می خواهیم تقسیم‌کنیم و به جای پارامتر دوم عددی را که میخواهیم عدد اول را بر آن تقسیم کنیم می نویسیم سپس تابع عدد اول را بر عدد دوم تقسیم کرده و خارج قسمت و باقیمانده را در متغیرس از نوع ldiv_t دخیره میکند.

۴۵۳) تابع void far line(int x1,int y1,int x2,int y2) در مود گرافیکی برای ترسیم یک خط استفاده می شود. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۴۵۴) پارامتر های ورودی x1 و y1 طول و عرض نقطه شروع ترسیم خط و پارامترهای ورودی x2 و y2 طول و عرض نقطه پایان ترسیم خط می باشد.

۴۵۵) تابع void far linerel(int dx , int dy) در مود گرافیکی برای ترسیم یک خط استفاده می شود. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۴۵۶) پارامتر های ورودی dx و dy طول و عرض نقطه پایان ترسیم خط را مشخص می نمایند.و نقطه شروع ترسم خط نیز مکان فعلی کرسر در نظر گرفته می شود.

۴۵۷) نحوه ترسیم خط به این شکل می باشد که نقطه فعلی کرسر هر چه باشد نقطه x آن را با dx که ما به تابع می دهیم جمع میکند سپس نقطه x پایان خط به دست می آید و نقطه فعلی y را با مقدار dy جمع کرده و نقطه y پایانی ترسیم بدست می آید.

۴۵۸) تابع void far lineto(int x , int y) در مود گرافیکی برای ترسیم یک خط استفاده می شود. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۴۵۹) پارامتر های ورودی x و y طول و عرض نقطه پایان ترسیم خط را مشخص می نمایند.و نقطه شروع ترسم خط نیز مکان فعلی کرسر در نظر گرفته می شود.پس از اجرای تابع برنامه یک خط را از نقطه فعلی کرسر تا نقطه ای که کاربر درخواست کرده است را رسم می نماید.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۴۶۰) تابع int lock(int handle , long offset , long long length) برای قفل کردن یک تابع در برابر share کردن به کار می رود. برای استفاده از این تابع باید هدر فایل <io.h> را به برنامه افزود.

۴۶۱) به این شکل که شما به جای پارامتر اول عددی قرار می دهید که مشخص می کند که‌فایل باز است یا بسته است،که البته این عدد را با استفاده از یک تابع دیگر به نام sopen بدست می آورید.

۴۶۲) در پارامتر دوم مقدار حجمی از فایل را که میخواهید قفل کنید را (به بایت) مشخص می کنید.

۴۶۳) در پارامتر سوم هم طول فایل را مشخص می نمایید.(به بایت).این تابع بعد از اجرا دو مقدار بازگشتی دارد که اگر ۰ را برگرداند به معنای اجرای موفقیت آمیز دارد و اگر مقدار 1- را برگرداند به معنای شکست در اجرای تابع می باشد.

۴۶۴) تابع int locking(int handle , int cmd , long length) برای قفل کردن یا باز کردن قفل یک فایل در مورد share استفاده میشود. برای استفاده از این تابع باید هدر فایل های <io.h> و <sys\locking.h> را به برنامه افزود.

۴۶۵) به جای پارامتر دوم شما مشخص می کنید که تابع قرار است چه کاری انجام دهد، که این عمل با اختصاص ثوابت امکان پذیر است. ثوابت مورد نظر در زیر آمده است:

۴۶۶) LK_LOCK و LK_RLCK برای قفل کردن فایل استفاده میشود و اگر قفل کردن ناموفق باشد، ۱۰ بار دیگر در ۱۰ ثانیه و به ازای هر ثانیه ۱ با ر دیگر تست می کند و در نهایت اگر نشد قفل کردن را رها می کند.

۴۶۷) LK_NBLCK و LK_NBRCLK برای قفل کردن فایل استفاده میشود و اگر قفل کردن ناموفق باشد قفل کردن را رها می کند.

۴۶۸) LK_UNLCK برای باز کردن قفل یک فایل استفاده می شود. البته فایل از قبل حتما باید قفل شده باشد. تابع در صورت اجرای موفقیت آمیز مقدار ۰ را بر میگرداند و در غیر این صورت مقدار 1- را بر می گرداند و شماره خطای رخ داده را در ERRNO قرار میدهد.

۴۶۹) تابع double log(double x) برای محاسبه مقدار لگاریتم طبیعی عدد x به کار می رود. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۴۷۰) تابع complex log(complex z) برای محاسبه مقدار لگاریتم طبیعی عدد مختلط z به کار می رود. برای استفاده از این تابع باید هدر فایل <complex.h> را به برنامه افزود.

۴۷۱) تابع double log10(double x) برای محاسبه مقدار لگاریتم در مبنای ۱۰ عدد x به کار می رود. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۴۷۲) تابع long double logl(long double x) برای محاسبه مقدار لگاریتم طبیعی عدد x به کار می رود. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۴۷۳) تابع long double log10l(long double x) برای محاسبه مقدار لگاریتم در مبنای ۱۰ عدد x به کار می رود. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۴۷۴) تابع complex log10(complex z) برای محاسبه مقدار لگاریتم در مبنای ۱۰ عدد مختلط z به کار می رود. برای استفاده از این تابع باید هدر فایل <complex.h> را به برنامه افزود.

۴۷۵) تابع void lowvideo(void) برای نمایش متن با حداقل روشنایی استفاده می شود.جهت مشاهده تغییرات این تابع بر روی متن مورد نظر قبل از چاپ متن بر روی صفحه نمایش این تابع را فراخوانی کنید و سپس در دستور بعد متن را چاپ کنید. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۴۷۶) تابع char * itoa(int value , char *string, int radix) برای تبدیل یک عدد صحیح به رشته استفاده می شود. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۴۷۷) در پارامتر اول مقداری را که میخواهید تبدیل شود را قرار می دهید و در پارامتر دوم رشته ای را که می خواهید مقدار تبدیل یافته در آن قرار بگیرد را قرار می دهید و در پارامتر آخر مبنای تبدیل عدد را مشخص می کنید که این عدد می تواند از ۲ تا ۳۶ تغییر کند.

۴۷۸) تابع char * ltoa(long value , char *string, int radix) برای تبدیل یک عدد صحیح بزرگ به رشته استفاده می شود. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۴۷۹) در پارامتر اول مقداری را که میخواهید تبدیل شود را قرار می دهید و در پارامتر دوم رشته ای را که می خواهید مقدار تبدیل یافته در آن قرار بگیرد را قرار می دهید و در پارامتر آخر مبنای تبدیل عدد را مشخص می کنید که این عدد می تواند از ۲ تا ۳۶ تغییر کند.

۴۸۰) تابع char * ultoa(unsigned long value , char *string, int radix) برای تبدیل یک عدد صحیح بزرگ بی علامت به رشته استفاده می شود. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۴۸۱) در پارامتر اول مقداری را که میخواهید تبدیل شود را قرار می دهید و در پارامتر دوم رشته ای را که می خواهید مقدار تبدیل یافته در آن قرار بگیرد را قرار می دهید و در پارامتر آخر مبنای تبدیل عدد را مشخص می کنید که این عدد می تواند از ۲ تا ۳۶ تغییر کند.

۴۸۲) تابع void *malloc(size_t size) برای دریافت پویای یک بلوک از حافظه توسط برنامه استفاده می شود. برای استفاده از این تابع باید هدر فایل های <alloc.h> یا <stdlib.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۴۸۳) شما باید به جای پارامتر تابع تعداد متغیرهای مورد نیاز را مشخص کنید. در صورت موفقیت تابع یک ادرس باز میگرداند و در غیر اینصورت مقدار null را باز می گرداند.

۴۸۴) تابع max(a,b) از بین دو مقدار داده شده مقدار بزرگتر را باز میگرداند. برای استفاده از این تابع باید ەدر فایل <stdlib.h> را به برنامه افزود.

۴۸۵) تابع min(a,b) از بین دو مقدار داده شده مقدار کوچکتر را باز میگرداند. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۴۸۶) تابع void *memccpy(void *dest,const void *src, int c, size_t n) برای کپی کردن مقادیر از پارامتر دوم به پارامتر اول استفاده می شود. برای استفاده از این تابع باید هدر فایل های <mem.h> یا <string.h> را به برنامه افزود.

۴۸۷) شما در پارامتر اول محل ذخیره اطلاعات را مشخص می کنید و در پارامتر دوم اطلاعاتی که میخواخید کپی کنید و در پارامتر سوم شرطی می گذارید که به محض رسیدن به آن کپی کردن متوقف شود و در پارامتر چهارم مشخص میکنید که کپی تا چه حجمی صورت گیرد.

۴۸۸) تابع void *memcpy(void *dest,const void *src, size_t n) برای کپی کردن مقادیر از پارامتر دوم به پارامتر اول استفاده می شود. برای استفاده از این تابع باید هدر فایل های <mem.h> یا <string.h> را به برنامه افزود.

۴۸۹) شما در پارامتر اول محل ذخیره اطلاعات را مشخص می کنید و در پارامتر دوم اطلاعاتی که میخواخید کپی کنید و در پارامترسوم مشخص میکنید که کپی تا چه حجمی صورت گیرددر این تابع اگر مقصد از منبع کمتر باشد رفتار تابع غیر قابل پیش بینی است.

۴۹۰) تابع void *memmove(void *dest,const void *src, size_t n) برای کپی کردن مقادیراز پارامتر دوم به پارامتر اول استفاده می شود. برای استفاده از این تابع باید هدر فایل های <mem.h> یا <string.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۴۹۱) شما در پارامتر اول محل ذخیره اطلاعات را مشخص می کنید و در پارامتر دوم اطلاعاتی که میخواخید کپی کنید و در پارامتر مشخص میکنید که کپی تا چه حجمی صورت گیرداگر مقصد از منبع کوپکتر باشد کپی اطلاعات تا زمانی که مقصد جا دارد انجام می پذیرد.

۴۹۲) تابع void *memchr(const void *s , int c , size_t n) برای جستجوی کاراکتر c در n بایت اول از s می باشد. برای استفاده ازاین تابع باید هدر فایل های <mem.h>یا<string.h>رابه برنامه افزود.

۴۹۳) اگر تابع بتواند c را در n بایت اول از s پیدا کند ادرس آن را بر میگرداند و در غیر اینصورت null رابر میگرداند.

۴۹۴) تابع int memcmp(const void *s1,const void *s2,size_t n) برای مقایسه n بایت اول دو رشته S1 و S2 می باشد. برای استفاده از این تابع باید هدر فایل های <mem.h> یا <string.h> را به برنامه افزود.

۴۹۵) اگر رشته اول از دوم کوچکتر باشد مقدار منفی و اگر با هم برابر باشن مقدار ۰ و اگر رشته دوم از اول بزرگتر باشد مقدار مثبت باز می گرداند.در این تابع کوچکی و بزرگی حروف مهم است.

۴۹۶) تابع int memicmp(const void *s1,const void *s2,size_t n) برای مقایسه n بایت اول دو رشته S1 و S2 می باشد. برای استفاده از این تابع باید هدر فایل های <mem.h> یا <string.h> را به برنامه افزود.

۴۹۷) اگر رشته اول از دوم کوچکتر باشد مقدار منفی و اگر با هم برابر باشن مقدار ۰ و اگر رشته دوم از اول بزرگتر باشد مقدار مثبت باز می گرداند. در این تابع کوچکی و بزرگی حروف مهم نیست.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۴۹۸) تابع void *memset(void *s , int c , size_t n) بع تعداد n بایت اول از s را با c پر می کند. برای استفاده از این تابع باید هدر فایل های <mem.h> یا <string.h> را به برنامه افزود.

۴۹۹) تابع int mkdir(const char *path) برای ساختن یک پوشه در آدرسی که در path مشخص شده استفاده میشود. اگر عمل با موفقیت انجام گیرد مقدار ۰ و درغیر اینصورت مقدار 1- را بر میگرداند. برای استفاده از این تابع باید هدر فایل <dir.h> را به برنامه افزود.

۵۰۰) تابع int rmdir(const char *path) برای pbt یک پوشه در آدرسی که در path مشخص شده استفاده میشود. اگر عمل با موفقیت انجام گیرد مقدار ۰ و درغیر اینصورت مقدار 1- را بر میگرداند. برای استفاده از این تابع باید هدر فایل <dir.h> را به برنامه افزود.

۵۰۱) تابع char *mktemp (char *template) برای ایجاد یک فایل temp به کار می رود، نام این فایل هم در پارامتر تابع آمده است. برای استفاده از این تابع باید هدر فایل <dir.h> را به برنامه افزود.

۵۰۲) تابع time_t mktime (struct tm *t) برای دریافت تاریخ کنونی و تبدیل آن به تقویمی می باشد.برای استفاده از این تابع باید هدر فایل <time.h> را به برنامه افزود.

۵۰۳) تابع void far *MK_FP(unsigned seg , unsigned ofs) برای ایجاد یک اشاره گر به دور استفاده می شود که باید شماره سگمنت را در seg و شماره آفست را در ofs قرار داد.مقدار بازگشتی این تابع یک اشاره گر به دور است. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۵۰۴) تابع unsigned FP_OFF(void far *p) برای بدست آوردن شماره آفست یک اشاره گر به دور به کار می رود که اشاره گر به دور را باید در پارانتر تابع قرار داد. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۵۰۵) تابع unsigned FP_SEG(void far *p) برای بدست آوردن شماره سگمنت یک اشاره گر به دور به کار می رود که اشاره گر به دور را باید در پارانتر تابع قرار داد. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۵۰۶) تابع double modf(double x , double *ipart) برای تقسیم یک عدد اعشاری به دو قسمت اعشاری و صحیح می باشد. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۵۰۷) شما باید عدد مورد نظر را به جای x قرار دهید سپس قسمت صحیح آن در ipart* قرار می گیرد و قسمت اعشاری ان هم توسط تابع باز گردانده می شود.

۵۰۸) تابع long double modfl(long double x , long double *ipart) برای تقسیم یک عدد اعشاری بزرگ به دو قسمت اعشاری و صحیح می باشد. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۵۰۹) شما باید عدد مورد نظر را به جای x قرار دهید سپس قسمت صحیح آن در ipart* قرار می گیرد و قسمت اعشاری ان هم توسط تابع باز گردانده می شود.

۵۱۰) تابع void movedata(unsigned srcseg , unsigned srcoff , unsigned destseg , unsigned destoff , size_t n) برای انتقال n بایت اطلاعات از آدرس srcseg:srcoff به destseg:destoff استفاده می شود. برای استفاده از این تابع باید هدر فایل های <mem.h> یا <string.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۵۱۱) تابع void far moverel(int dx,int dy) کرسر را به موقعیت جدید انتقال می دهد،طول محل جدید به صورت جمع طول محل قبل با پارامتر dx و عرض محل جدید به صورت جمع عرض قبل با پارامتر dy بدست می آید. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۵۱۲) تابع void far moveto(int x,int y) کرسر را به موقعیت جدید انتقال می دهد،طول محل جدید x و عرض محل جدید y خواهد بود. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۵۱۳) تابع int movetext(int left , int top , int right , int bottom , int destleft , int desttop) برای انتقال متن از محلی به محلی دیگر استفاده میشود. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۵۱۴) آدرس متن مبدا را با استفاده از پارامترهای اول تا چهارم تعیین میکنیم و آدرس مقصد را با پارامترهای پنجم و ششم مشخص می کنیم.

۵۱۵) تابع void movemem(void *src , void *dest , unsigned length) برای جابجایی اطلاعات حافظه مبدا با آدرس src به آدرس مقصد dest به طول length می باشد. برای استفاده از این تابع باید هدر فایل <mem.h> را به برنامه افزود.

۵۱۶) در برنامه ها به جای مقدار (PI=3.14) از ثابت M_PI استفاده می شود. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۵۱۷) در برنامه ها به جای مقدار 2/ (PI=3.14) از ثابت M_PI _2 استفاده می شود. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۵۱۸) در برنامه ها به جای مقدار PI=3.14) /4 از ثابت M_PI_4 استفاده می شود. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۵۱۹) در برنامه ها به جای مقدار (PI=3.14)/1 از ثابت M_1_PI استفاده می شود. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۵۲۰) در برنامه ها به جای مقدار (PI=3.14)/2 از ثابت M_2_PI استفاده می شود. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۵۲۱) در برنامه ها به جای مقدار (PI=3.14)√/1 از ثابت M_PI استفاده می شود. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۵۲۲) در برنامه ها به جای مقدار (PI=3.14)√/2 از ثابت M_PI استفاده می شود. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۵۲۳) در برنامه ها به جای مقدار عدد نپر (e) از ثابت M_E استفاده می شود. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۵۲۴) در برنامه ها به جای مقدار log(e) از ثابت M_LOG2E استفاده می شود. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۵۲۵) در برنامه ها به جای مقدار log10(e) از ثابت M_LOG10E استفاده می شود. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۵۲۶) در برنامه ها به جای مقدار ln(2) از ثابت M_LN2 استفاده می شود. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۵۲۷) در برنامه ها به جای مقدار ln(10) از ثابت M_LN10 استفاده می شود. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۵۲۸) برای اینکه در برنامه ها بتوانید اشاره گر نزدیک به متغیر تعریف کنید باید به شکل زیر عمل نمایید: <type> near <pointer definition> به جای قسمت اول نوع متغیر را مشخص نمایید سپس کلمه near را تایپ کرده و در آخر نام یک اشاره گر را با علامت * بنویسید. این عمل اشاره گر را برای فضای حافظه 64KB ایجاد می کند.

۵۲۹) برای اینکه در برنامه ها بتوانید اشاره گر نزدیک به تابع تعریف کنید باید به شکل زیر عمل نمایید: <type> near <function definition> به جای قسمت اول نوع تابع را مشخص نمایید سپس کلمه near راتایپ کرده و درآخرنام تابع رابنویسید. این عمل اشاره گربه تابع را برای فضای حافظه 64KB ایجاد می کند.

۵۳۰) برای دریافت حافظه پویا از سیستم در برنامه باید از دستور new به صورت زیر استفاده کرد: <pointer_to_name>=new <name> [<name_initializer>] به این شکل که به جای قسمت اول نام اشاره گر تعریف شده و سپس new سپس نام نوع و در نهایت تعداد مورد نیازاز آن حافظه که گاهی اوقات به صورت متغیری وگاهی اوقات به صورت عدد مستقیم نوشته می شود.

۵۳۱) تابع double norm(complex z) برای محاسبه مجذور عدد مختلط z می باشد. برای استفاده از این تابع باید هدر فایل <complex.h> را به برنامه افزود.

۵۳۲) تابع void normvideo(void) برای نمایش متن با میزان روشنایی نرمال استفاده می شود.جهت مشاهده تغییرات این تابع بر روی متن مورد نظر قبل از چاپ متن بر روی صفحه نمایش این تابع را فراخوانی کنید و سپس در دستور بعد متن را چاپ کنید. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۵۳۳) تابع void nosound (void) برای خاموش کردن اسپیکر سیستم استفاده می شود. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۵۳۴) NULL در زبان C++ نمایانگر پوچ می باشد. که معمولا در اشاره گرها کاربرد دارد.

۵۳۵) تابع int outp(unsigned portid , int value) یک بایت را به پورتی که آدرس ان در پارامتر اول آمده است ارسال می کند. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۵۳۶) تابع int outport(int portid , int value) یک بایت را به پورتی که آدرس ان در پارامتر اول آمده است ارسال می کند. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۵۳۷) تابع int outportb(int portid , unsigned char value) یک بایت را به پورتی که آدرس ان در پارامتر اول آمده است ارسال می کند. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۵۳۸) تابع unsigned outpw(unsigned portid , unsigned value) یک کلمه را به پورتی که آدرس آن درپارامتراول آمده است ارسال میکند. برای استفاده ازاین تابع باید هدرفایل <conio.h> رابه برنامه افزود.

۵۳۹) تابع void far outtext(char far *textstring) برای نمایش متن در مود گرافیکی به کار می رود.برای استفاده ازاین تابع باید هدرفایل <graphics.h> رابه برنامه افزود.

۵۴۰) تابع void far outtextxy(int x,int y,char far *textstring) برای نمایش متن در مود گرافیکی در مکان دلخواه به کار می رود، به این شکل که شما آدرس مکان دلخواه را از

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

طریق x و y مشخص می کنید سپس تابع برای شمامتن رامینویسد.برای استفاده ازاین تابع باید هدرفایل <graphics.h> رابه برنامه افزود.

۵۴۱) تابع int peek(unsigned segment , unsigned offset) برای خواندن یک کلمه از آدرسی که سگمنت و آفست آن مشخص شده استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <dos.h> رابه برنامه افزود.

۵۴۲) تابع char peekb(unsigned segment , unsigned offset) برای خواندن یک بایت از آدرسی که سگمنت و آفست آن مشخص شده استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <dos.h> رابه برنامه افزود.

۵۴۳) تابع void perror(const char *s) برای چاپ پیغام خطای سیتم به همراه پیغامی که به جای پارامتر تابع نوشته ایم به کار می رود. برای استفاده ازاین تابع باید هدرفایل <stdio.h> رابه برنامه افزود.

۵۴۴) تابع void far pieslice(int x , int y , int stangle , int endangle , int radius) در مود گرافیکیبرای ترسیم دایره با مختصات مرکز x و y و زاویه شروع و پایان stangle و endangle و شعاع radius به کار می رود. برای استفاده ازاین تابع باید هدرفایل <graphics.h> رابه برنامه افزود.

۵۴۵) در زبان C++ اگر از نوع داده ای pointer تعریف کنید در واقع متغیری تعریف کرده اید که قادر به نگهداری آدرس یک متغیر از نوع خود می باشد.

۵۴۶) نقطه از دو مختصات x و y تشکیل شده است. اگر این دو عدد را در یک نوع داده ای ذخیره کنیم، آن را pointtype مینامیم که در مود گرافیکی هم کاربرد خواهد داشت.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۵۴۷) تابع void poke(unsigned segment , unsigned offset , int value) برای نوشتن یک کلمه در آدرسی از حافظه که با سگمنت و آفست تعیین شده کاربرد دارد. برای استفاده ازاین تابع باید هدرفایل <dos.h> رابه برنامه افزود.

۵۴۸) تابع void pokeb(unsigned segment , unsigned offset ,char value) برای نوشتن یک بایت در آدرسی از حافظه که با سگمنت و آفست تعیین شده کاربرد دارد. برای استفاده ازاین تابع باید هدرفایل <dos.h> رابه برنامه افزود.

۵۴۹) تابع complex polar(double mag , double angle) برای بدست آوردن یک عدد مختلط کاربرد دارد. برای استفاده ازاین تابع باید هدرفایل <complex.h> رابه برنامه افزود.

۵۵۰) تابع double poly(double x , int degree , double coeff[ ]) برای محاسبه مقدار چند جمله ای x از درجه degree با ضرایب coeff استفاده میشود. برای استفاده ازاین تابع باید هدرفایل <math.h> رابه برنامه افزود.

۵۵۱) تابع long double polyl(long double x , int degree ,long double coeff[ ]) برای محاسبه مقدار چند جمله ای x از درجه degree با ضرایب coeff استفاده میشود. برای استفاده ازاین تابع باید هدرفایل <math.h> رابه برنامه افزود.

۵۵۲) تابع double pow(double x , double y) برای محاسبه x به توان y می باشد. برای استفاده ازاین تابع باید هدرفایل <math.h> رابه برنامه افزود.

۵۵۳) تابع long double powl(long double x ,long double y) برای محاسبه x به توان y می باشد. برای استفاده ازاین تابع باید هدرفایل <math.h> رابه برنامه افزود.

۵۵۴) تابع complex pow(complex x , complex y) برای محاسبه xبه توانy می باشد. برای استفاده از این تابع باید هدر فایل <complex.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۵۵۵) تابع complex pow(complex x , double y) برای محاسبه xبه توانy می باشد. برای استفاده از این تابع باید هدر فایل <complex.h> را به برنامه افزود.

۵۵۶) تابع complex pow(double x , double y) برای محاسبه xبه توانy می باشد. برای استفاده از این تابع باید هدر فایل <complex.h> را به برنامه افزود.

۵۵۷) تابع double pow10(int p) برای محاسبه 10 به توان p می باشد. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۵۵۸) تابع long double pow10l(int p) برای محاسبه 10 به توان p می باشد. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۵۵۹) تابع printf ("const string and formatted" , p1,p2,...) برای چاپ پیغام ها و مقادیر بدست آمده در برنامه ها استفاده می شود. به این صورت که شما می توانید در قسمت اول پیغام ثابت و فرمت متغیرهای برنامه را انتخاب کنید و در قسمت بعد به جای pها به ترتیب نام متغیر ها را وارد نمایید. برای استفاده از این تابع باید هدر فایل <stdio.h> را به برنامه افزود.

۵۶۰) برای تعریف یک کلاس از نوع محلی باید آن را به شکل private تعریف کنیم.

۵۶۱) برای تعریف یک کلاس از نوع سراسری باید آن را به شکل public تعریف کنیم.

۵۶۲) برای تعریف یک کلاس از نوع محافظت شده باید آن را به شکل protected تعریف کنیم.

۵۶۳) تابع int getc(FILE *stream) برای خواندن یک کاراکتر از فایلی است که آدرسش در پارامتر تابع آورده شده است. برای استفاده از این تابع باید هدر فایل <stdio.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۵۶۴) تابع int putc(int c , FILE *stream) برای نوشتن یک کاراکتر در فایلی که ادرسش در پارامتر تابع قرار دارد استفاده می شود. برای استفاده از این تابع باید هدر فایل <stdio.h> را به برنامه افزود.

۵۶۵) تابع int putch(int ch) برای نوشتن کاراکتر ch در خروجی صفحه نمایش به کار می رود. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۵۶۶) تابع int putchar(int ch) برای نوشتن کاراکتر ch در خروجی صفحه نمایش به کار می رود. برای استفاده از این تابع باید هدر فایل <stdio.h> را به برنامه افزود.

۵۶۷) تابع void far putpixel(int x , int y , int color) برای نمایش یک پیکسل در مختصات x و y و با رنگ color می باشد. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۵۶۸) تابع int puts (const char *s) برای نمایش یک رشته در خروجی صفحه نمایش استفاده می شود، بعد از چاپ رشته کرسر به سطر بعد منتقل می شود. برای استفاده از این تابع باید هدر فایل <stdio.h> را به برنامه افزود.

۵۶۹) تابع int puttext(int left , int top , int right , int bottom , void *source) برای نمایش متن در خروجی و در یک قسمت مشخص از صفحه نمایش که آدرس آن با مختصات داده شده از حافظه تعیین میگردد استفاده می گردد. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۵۷۰) تابع int putw(int w,FILE *stream) برای نوشتن یک کلمه در فایلی که آدرس ان در پارامتر تابع تعیین شده است استفاده می شود. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۵۷۱) تابع int raise(int sig) برای ارسال سیگنال نرم افزاری به برنامه استفاده می شود. برای استفاده از این تابع باید هدر فایل <signal.h> را به برنامه افزود.

۵۷۲) تابع int rand(void) برای ایجاد اعداد تصادفی استفاده می شود. که این اعداد معمولا از محدوده ۰ تا ۳۲۷۶۷ می باشند. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۵۷۳) تابع int random(int num) برای اسجاد اعداد تصادفی در بر نامه استفاده می شود که رنج آنها از ۰ تا num-1 میباشند. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۵۷۴) تابع void randomize(void) را قبل از تابع rand به کار می بریم فقط برای اولین بار، برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۵۷۵) ثابت RAND_MAX به شما کمک میکند تا بتوانید بیشترین مقدار تصادفی قابل ایجاد توسط تابع rand را بدست آورید. برای استفاده از این ثابت باید هدر فایل <stdlib.h> را به برنامه افزود.

۵۷۶) تابع double real(complex z) برای بدست آوردن قسمت حقیقی عدد مختلط z به کار می رود. برای استفاده از این تابع باید هدر فایل <complex.h> را به برنامه افزود.

۵۷۷) تابع double real(bcd x) برای تبدیل عددbcd به عدد double به کار می رود. برای استفاده از این تابع باید هدر فایل <bcd.h> را به برنامه افزود.

۵۷۸) تابع void far rectangle(int left , int top , int right , int bottom) در مود گرافیکی برای ترسیم یک مستطیل با مختصات داده شده در تابع استفاده میشود. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۵۷۹) در زبان C++ و در مود گرافیکی به جای رنگ قرمز از ثابت RED میتوان استفاده کرد .

۵۸۰) در هنگام تعریف متغیر میتوانید از کلمه register استفاده کنید تا متغیری را که تعریف میکنید به جای ram از register های cpu انتخاب شود تا در استفاده از آن متغیر سرعت بالایی داشته باشید.

۵۸۱) تابع int remove(const char *filename) برای حذف یک فایل در برنامه استفاده می شود. اگر موفقیت آمیز حذف شود مقدار ۰ و در غیر اینصورت 1- را بر میگرداند. برای استفاده از این تابع باید هدر فایل <stdio.h> را به برنامه افزود.

۵۸۲) در زبان C++ برای بازگشت دادن یک مقدار اصلی از یک تابع از دستور return x استفاده می شود.

۵۸۳) تابع int rmdir(const char *path) برای حذف یک پوشه توسط برنامه استفاده می شود. آدرس پوشه در پارامتر تابع وجو دارد. برای استفاده از این تابع باید هدر فایل <dir.h> را به برنامه افزود.

۵۸۴) تابع int rmtmp(void) برای حذف فایل های temp ساخته شده استفاده می شود. برای استفاده از این تابع باید هدر فایل <stdio.h> را به برنامه افزود.

۵۸۵) تابع scanf("format's" , variable's) برای خواندن مقادیر از ورودی استفاده می شود. به این شکل که ابتدا فرمت ها را در قسمت اول تابع می نویسیم و سپس به ترتیب فرمتها مقادیر متغیر ها را یک به یک از ورودی می خوانیم برای استفاده از این تابع باید هدر فایل <stdio.h> را به برنامه افزود.

۵۸۶) مقدار ثابت SEEK_CUR به معنای seek کردن در فایل از محلی که اشاره گر فایل قرار دارد.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۵۸۷) مقدار ثابت SEEK_END به معنای seek کردن از انتهای فایل می باشد.

۵۸۸) مقدار ثابت SEEK_SET به معنای seek کردن از ابتدای فایل می باشد.

۵۸۹) تابع void setcolor(int color) در مود گرافیکی برای تنظیم رنگ به کار میرود. که رنگ انتخابی را باید به جای پارامتر تابع قرار دهید.برای استفاده ازاین تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۵۹۰) تابع void setdate(struct date *datep) برای تنظیم کودن تاریخ سیستم به کار میرود، به این شکل که شما باید تاریخ را به صورت پارامتر ورودی به تابع دهید تا تابع بتواند تاریخ سیستم را تغییر دهد. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۵۹۱) تابع unsigned far setgraphbufsize (unsigned bufsize) میزان بافر مورد استفاده توسط مود گرافیکی را تعیین میکند، مقدار پیش فرض ۴۰۹۶ بایت است. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۵۹۲) تابع void far setgraphmode(int mode) برای مود گرافیکی به کار میرود. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۵۹۳) تابع void settime(struct time *timep) برای تنظیم ساعت سیستم به کار می رود. ساعت جدید را در پارامتر ورودی به تابع مب دهید تا تابع ساعت جدید را به سیستم اعمال کندو برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۵۹۴) تابع double sin(double x) مقدار sin را برای x محاسبه می کند. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۵۹۵)) تابع long double sinl(long double x) مقدار sin را برای x محاسبه می کند. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۵۹۶) تابع complex sin(complex z) مقدار sin را برای z محاسبه می کند. برای استفاده از این تابع باید هدر فایل <complex.h> را به برنامه افزود.

۵۹۷) تابع double sinh(double x) مقدار sin ه ایپر بولیک را برای x محاسبه می کند. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۵۹۸) تابع long double sinhl(long double x) مقدار sin هایپر بولیک را برای x محاسبه می کند. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۵۹۹) تابع complex sinh(complex z) مقدار sin هایپر بولیک را برای z محاسبه می کند. برای استفاده از این تابع باید هدر فایل <complex.h> را به برنامه افزود.

۶۰۰) عملگر sizeof(type) این قابلیت را به شما میدهد که بتوانید حجم هر نوعی را که بخواهید محاسبه کنید، کافیست نام آن نوع و یا حتی متغیری را به جای type بنویسید عملگر حجم ان را بر می گرداند.

۶۰۱) تابع void sleep(unsigned seconds) باعث توقف اجرای برنامه برای اجرای وقفه ها به مدت تعداد ثانیه هایی که درپارامترتعیین میکنید.برای استفاده ازاین تابع بایدهدر فایل <dos.h> را به برنامه افزود.

۶۰۲) تابع void sound(unsigned frequency) برای ایجاد صوت با فرکانس دلخواه در برنامه می باشد. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۶۰۳) تابع double sqrt(double x) برای محاسبه جذر عدد x به کار می رود. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۶۰۴) تابع long double sqrtl(long double x) برای محاسبه جذر عدد x به کار می رود. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۶۰۵) تابع complex sqrt(complex z) برای محاسبه جذر عدد z به کار می رود. برای استفاده از این تابع باید هدر فایل <complex.h> را به برنامه افزود.

۶۰۶) برای تعریف یک متغیر که بتواند مقدار خود را در رفت و برگشت ها از توابع حفظ کند از کلاس حافظه static استفاده می شود.

۶۰۷) تابع int stime(time_t *tp) برای تظیم تاریخ و ساعت سیستم به کار می رود. شما باید مقدار تاریخ و ساعت کنونی را در پارامتر تابع قرار دهید سپس تابع تاریخ و ساعت سیستم را تنظیم خواهد کرد. برای استفاده از این تابع باید هدر فایل <time.h> را به برنامه افزود.

۶۰۸) تابع char *strcpy(char *dest , const char *src) برای کپی کردن رشته از پارامتر دوم به پارامتر اول استفاده می شود. مقدار بازگشتی هم پارامتر اول می باشد که در واقع src خواهد بود. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۰۹) تابع char *strcat(char *dest , const char *src) برای الحاق رشت دوم به انتهای رشته اول به کار می رود. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۱۰) تابع char *strchr(const char *s , int c) کاراکتر c را درون رشته s جستجو کرده و آدرس اولین وقوع آن را بر میگرداند. در غیر اینصورت مقدار NULL را بر می گرداند. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۶۱۱) تابع int strcmp(const char *s1 , const char *s2) برای مقایسه رشته های s1 و s2 به صورت کد اسکی به کار می رود. و کوچکی و بزرگی حروف برای تابع مهم می باشند. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۱۲) تابع int far _fstrcmp(const far char *s1 , const far char *s2) برای مقایسه رشته های s1 و s2 به صورت کد اسکی به کار می رود. و کوچکی و بزرگی حروف برای تابع مهم می باشند.این تابع برای رشته های با آدرس far استفاده می شود. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۱۳) تابع int strcmpi(const char *s1 , const char *s2) برای مقایسه رشته های s1 و s2 به صورت غیر کد اسکی به کار می رود. و کوچکی و بزرگی حروف برای تابع مهم نمی باشند. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۱۴) تابع int stricmp(const char *s1 , const char *s2) برای مقایسه رشته های s1 و s2 به صورت غیر کد اسکی به کار می رود. و کوچکی و بزرگی حروف برای تابع مهم نمی باشند. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۱۵) تابع int far _fstricmp(const far char *s1 , const far char *s2) برای مقایسه رشته های s1 و s2 به صورت غیر کد اسکی به کار می رود. و کوچکی و بزرگی حروف برای تابع مهم نمی باشند.این تابع برای رشته های با آدرس far استفاده می شود. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۱۶) تابع int strcoll(char *s1 , char *s2) برای مقایسه دو رشته s1 و s2 استفاده می شود. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۶۱۷) تابع char far *_fstrcpy(char far *dest , const char far *src) برای کپی کردن رشته دوم در رشته اول به کار می رود. این تابع برای رشته های با آدرس far استفاده می شود. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۱۸) تابع size_t strcspn(const char *s1 , const char *s2) برای پیدا کردن آدرس کوچکترین کاراکتری از رشته اول که در رشته دوم قرار دارد استفاده می شود. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۱۹) تابع size_t far _fstrcspn(const char far *s1 , const char far *s2) برای پیدا کردن آدرس کوچکترین کاراکتری از رشته اول که در رشته دوم قرار دارد استفاده می شود. این تابع برای رشته های با آدرس far استفاده می شود. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۲۰) تابع size_t strspn(const char *s1 , const char *s2) برای یافتن طول زیر رشته ای در رشته دوم از رشته اول به این صورت که این زیر رشته باید از ابتدای رشته رشته اول و پیوسته باشد. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۲۱) تابع size_t far _fstrspn(const char far *s1 , const char far *s2) برای یافتن طول زیر رشته ای در رشته دوم از رشته اول به این صورت که این زیر رشته باید از ابتدای رشته رشته اول و پیوسته باشد.این تابع برای رشته های با آدرس far استفاده می شود. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۲۲) تابع char *strdup(const char *s) برای کپی کردن مقدار رشته s را در strdup کپی می کند. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۶۲۳) تابع char far *_fstrdup(const char far *s) برای کپی کردن مقدار رشته s را در strdup کپی می کند.این تابع برای رشته های با آدرس far استفاده می شود. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۲۴) تابع char *strerror (int errnum) پیغام خطای پیش آمده را که در پارامتر تابع وجود دارد را به صورت رشته بر می گرداند. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۲۵) تابع char *_strerror (const char *s) برای ایجاد پیغام خطای سفارشی به کار می رود. بیای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۲۶) تابع size_t strlen(const char *s) برای بدست آوردن طول یک رشته به کار می رود. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۲۷) تابع size_t far _fstrlen(const char far *s) برای بدست آوردن طول یک رشته به کار می رود. این تابع برای رشته های با آدرس far استفاده می شود. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۲۸) تابع char *strlwr(char *s) برای تبدیل کلیه حروف رشته به حروف کوچک استفاده می شود. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۲۹) تابع char far * far _fstrlwr(char far *s) برای تبدیل کلیه حروف رشته به حروف کوچک استفاده می شود. این تابع برای رشته های با آدرس far استفاده می شود. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۳۰) تابع char *strupr(char *s) برای تبدیل کلیه حروف رشته به حروف بزرگ استفاده می شود. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۶۳۱) تابع char far * far _fstrupr(char far *s) برای تبدیل کلیه حروف رشته به حروف بزرگ استفاده می شود. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۳۲) تابع char *strncat(char *dest , const char *src , size_t maxlen) برای الحاق رشته دوم به رشته اول استفاده می شود. به این صورت که به تعداد عددی که در پارامتر سوم نوشته ایم کاراکتر از رشته دوم به انتهای رشته اول الحاق می کند. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۳۳) تابع char far * far f_strncat(char far *dest , const char far *src , size_t maxlen) برای الحاق رشته دوم به رشته اول استفاده می شود. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۳۴) به این صورت که به تعداد عددی که در پارامتر سوم نوشته ایم کاراکتر از رشته دوم به انتهای رشته اول الحاق می کند.این تابع برای رشته های با آدرس far استفاده می شود.

۶۳۵) تابع int strncmp(const char *s1 , const char *s2 , size_t maxlen) برای مقاسیه n کاراکتر اول دو رشته مورد استفاده قرار میگیرد. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۳۶) که شما باید طول مورد نظرتان را برای مقایسه در پارامتر آخر وارد نمایید.این تابع برای مقایسه از کد اسکی کاراکتر ها برای مقایسه استفاده می کند.

۶۳۷) تابع int strncmpi(const char *s1 , const char *s2 , size_t maxlen) برای مقاسیه n کاراکتر اول دو رشته مورد استفاده قرار میگیرد. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۶۳۸) که شما باید طول مورد نظرتان را برای مقایسه در پارامتر آخر وارد نمایید. این تابع بدون توجه به کوچکی و بزرگی حروف مقایسه انجام می دهد.

۶۳۹) تابع int strnicmp(const char *s1 , const char *s2 , size_t maxlen) برای مقاسیه n کاراکتر اول دو رشته مورد استفاده قرار میگیرد. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۴۰) که شما باید طول مورد نظرتان را برای مقایسه در پارامتر آخر وارد نمایید. این تابع بدون توجه به کوچکی و بزرگی حروف مقایسه انجام می دهد.

۶۴۱) تابع int far f_strncmp(const char far *s1 , const char far *s2 , size_t maxlen) برای مقاسیه n کاراکتر اول دو رشته مورد استفاده قرار میگیرد. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۴۲) که شما باید طول مورد نظرتان را برای مقایسه در پارامتر آخر وارد نمایید. این تابع برای رشته های با آدرس far استفاده می شود.

۶۴۳) تابع int far f_strnicmp(const char far *s1 , const char far *s2 , size_t maxlen) برای مقاسیه n کاراکتر اول دو رشته مورد استفاده قرار میگیرد. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۴۴) که شما باید طول مورد نظرتان را برای مقایسه در پارامتر آخر وارد نمایید.این تابع بدون توجه به کوچکی و بزرگی حروف مقایسه انجام می دهد.این تابع برای رشته های با آدرس far استفاده می شود.

۶۴۵) تابع char *strncpy(char *dest , const char *src , size_t maxlen) برای کپی کردن تعداد n کاراکتر از رشته مبدا در رشته مقصد استفاده می شود. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۶۴۶) تابع char far *far f_strncpy(char far *dest , const char far *src , size_t maxlen) برای کپی کردن تعداد n کاراکتر از رشته مبدا در رشته مقصد استفاده می شود.این تابع برای رشته های با آدرس far استفاده می شود. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۴۷) تابع char * strnset(char *s , int ch , size_t n) برای پر کردن n کاراکتر اول رشته s با کاراکتر ch می باشد. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۴۸) تابع char far * far _fstrnset(vhar far *s , int ch , size_t n) برای پر کردن n کاراکتر اول رشته s با کاراکتر ch می باشد. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۴۹) تابع char *strpbrk(const char *s1 , const char *s2) رشته اول را برای اولین وقوع یکی از کاراکترهای رشته دوم جستجو می کند. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۵۰) تابع char far *far _fstrpbrk(const char far *s1 , const char far *s2) رشته اول را برای اولین وقوع یکی از کاراکترهای رشته دوم جستجو می کند.این تابع برای رشته های با آدرس far استفاده می شود. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۵۱) تابع char *strrch(const char *s , int c) برای جستجوی کاراکتر c در رشته s از آخر به اول استفاده می شود. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۶۵۲) تابع char far * far strrch(const char far *s , int c) برای جستجوی کاراکتر c در رشته s از آخر به اول استفاده می شود.از این تابع برای رشته های با آدرس far استفاده می شود. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۵۳) تابع char *strrev(char *s) یک رشته را به عنوان ورودی پذیرفته و آن را معکوس می کند. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۵۴) تابع char far * far _fstrrev(char far *s) یک رشته را به عنوان ورودی پذیرفته و آن را معکوس می کند.این تابع برای رشته های با آدرس far استفاده می شود. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۵۵) تابع char *strset(char *s , int ch) برای پر کردن یک رشته با یک کاراکتر خاص استفاده می شود. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۵۶) تابع char far * far _fstrset(char far *s , int ch) برای پر کردن یک رشته با یک کاراکتر خاص استفاده می شود. این تابع برای رشته های با آدرس far استفاده می شود. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۵۷) تابع char *strstr(const char *s1 , const char *s2) برای یافتن زیر رشته دوم در رشته اول به کار می رود. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۵۸) تابع char far * far _fstrstr(const char far *s1 , const char far *s2) برای یافتن زیر رشته دوم در رشته اول به کار می رود.این تابع برای رشته هایی که در آدرس far قرار دارند استفاده می شوند. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۶۵۹) تابع double strtod(const char *s , char **endptr) برای تبدیل یک رشته به double استفاده می شود. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۶۶۰) تابع long strtol(const char *s , char **endpty , int darix) برای تبدیل یک رشته به long استفاده می شود.شما میتوانید به جای پارامتر سوم مبنای این تبدیل را تعیین نمایید. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۶۶۱) تابع long double _strtold(const char *s , char **endptr) برای تبدیل یک رشته به long double استفاده می شود. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۶۶۲) تابع unsigned long strtoul(const char *s , char **endptr , int radix) برای تبدیل یک رشته به unsigned long استفاده می شود.شما میتوانید به جای پارامتر سوم مبنای این تبدیل را تعیین نمایید. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۶۶۳) برای ایجاد یک ساختار جدید در زبان C++ از struct استفاده می شود.

۶۶۴) تابع size_t strxfrm (char *s1,cahr *s2,size_t n) برای کپی کردن رشته دوم در رشته اول و به طول n به کار می رود. برای استفاده از این تابع باید هدر فایل <string.h> را به برنامه افزود.

۶۶۵) تابع void swab(char *from , char *to , int nbytes) رشته ای را که در from قرار دارد را در to ذخیره می کند اما به صورت در هم این کار را انجام می دهد. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۶۶۶) تابع int system(const char *command) برای اجرای فرامین dos استفاده می شود. برای استفاده از این تابع باید هدر فایل های <stdlib.h> یا <process.h> را به برنامه افزود.

۶۶۷) ثابت E2BIG به معنای خطای Arg List too Long می باشد. یعنی لیست Arg ورودی تابع پارامترهایش بیش از حد انتظار بوده است.

۶۶۸) ثابت EACCES به معنای خطای Permission denied می باشد. یعنی دسترسی شما مجاز نمی باشد. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۶۶۹) ثابت EBADF به معنای خطای Bad file number می باشد. یعنی شماره فایلی را که وارد کرده این نامعتبر است. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۶۷۰) ثابت ECONTR به معنای خطای Memory blocks destroyed می باشد. یعنی حافظه ای که میخواهید دسترسی داشته باشید خراب است. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۶۷۱) ثابت ECURDIR به معنای خطای Attempt to remove curdir می باشد. یعنی تلاش برای حذف درایو با شکست مواجه شده است. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۶۷۲) ثابت EDOM به معنای خطای Domain Error می باشد. یعنی آدرس وارد شده با خطا مواجه شده است. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۶۷۳) ثابت EEXIST به معنای خطای File already exists می باشد. یعنی فایلی که قصد باز کردن و یا وارد کردن آن به برنامه را دارید باز است و یا در برنامه وجود دارد. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۶۷۴) ثابت EFAULT به معنای خطای Unknown error می باشد. یعنی خطای ناشناس اتفاق افتاده است. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۶۷۵) ثابت EINVACC به معنای خطای Invalid access code می باشد. یعنی کد دسترسی شما نا معتبر می باشد. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۶۷۶) ثابت EINVAL به معنای خطای Invalid argument می باشد. یعنی آرگومان نامعتبر وارد شده است. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۶۷۷) ثابت EINVADT به معنای خطای Invalid data می باشد. یعنی اطلاعات وارد شده نامعتبر می باشد. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۶۷۸) ثابت EINVDRV به معنای خطای Invalid drive specified می باشد. یعنی درایو مشخص شده نامعتبر می باشد. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۶۷۹) ثابت EINVENV به معنای خطایInvalid environment می باشد. یعنی محیط درخواستی شما نامعتبر است. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۶۸۰) ثابت EINVFMT به معنای خطای Invalid format می باشد. یعنی فرمت مشخص شده نامعتبر می باشد. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۶۸۱) ثابت EINVFNC به معنای خطای Invalid function number می باشد. یعنی شماره تابعی که درخواست کرده اید نامعتبر می باشد. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۶۸۲) ثابت EINVMEM به معنای خطای Invalid memory block address می باشد. یعنی آدرس بلوکی از حافظه که شما آدرسش را تعیین کرده اید نامعتبر می باشد. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۶۸۳) ثابت EMFILE به معنای خطای Too many open fiels می باشد. یعنی تعداد فایل های باز که در برنامه در حال استفاده است از تعداد معمول بیشتر شده است. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۶۸۴) ثابت ENMFILE به معنای خطای No more files می باشد. یعنی برنامه توانایی باز کزدن فایل های بیشتر را ندارد. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۶۸۵) ثابت ENODEV به معنای خطای No such device می باشد. یعنی برنامه قابلیت پشتیبانی از چنین فایل هایی را ندارد. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۶۸۶) ثابت ENOENT به معنای خطای No such file or directory می باشد. یعنی برنامه قابلیت پشتیبانی از چنین فایل ها و پوشه هایی را ندارد. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۶۸۷) ثابت ENOEXEC به معنای خطای Exec format error می باشد. یعنی فرمت اجرایی با خطا مواجه شده است. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۶۸۸) ثابت ENOFILE به معنای خطای No such file or directory می باشد. یعنی برنامه قابلیت پشتیبانی از چنین فایل ها و پوشه هایی را ندارد. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۶۸۹) ثابت ENOMEM به معنای خطای Not enough memory می باشد. یعنی سیستم به میزران کافی حافظه در دسترس ندارد. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۶۹۰) ثابت ENOPATH به معنای خطای Path not found می باشد. یعنی آدرس مورد نظر یافت نشد. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۶۹۱) ثابت ENOTSAM به معنای خطای Not same device می باشد. یعنی برنامه توانایی کار با دستگاه های مشابه را ندارد. برای استفاده از این تابع باید هدر فلیل <stdlib.h> را به برنامه افزود.

۶۹۲) ثابت ERANGE به معنای خطای Result out of range می باشد. یعنی نتیجه برنامه خارج از رنج قابل پشتیبانی می باشد. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۶۹۳) ثابت EXDEV به معنای خطای Cross-device link می باشد. یعنی خطا در تقاطع اتباطی دستگاه با دیگر دستگاه ها. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به برنامه افزود.

۶۹۴) ثابت EZERO به معنای خطای Error 0 می باشد. یعنی خطای ۰ پیش آمده است. برای استفاده از این تابع باید هدر فایل <stdlib.h> را به بونامه افزود.

۶۹۵) تابع double tan(double x) برای محاسبه مقدار تانژانت x میباشد. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۶۹۶) تابع long double tanl(long double x) برای محاسبه مقدار تانژانت x میباشد. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۶۹۷) تابع complex tan(complex x) برای محاسبه مقدار تانژانت x میباشد. برای استفاده از این تابع باید هدر فایل <complex.h> را به برنامه افزود.

۶۹۸) تابع double tanh(double x) برای محاسبه مقدار تانژانت هایپربولیک عدد x می باشد. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۶۹۹) تابع long double tanhl(long double x) برای محاسبه مقدار تانژانت هایپربولیک عدد x می باشد. برای استفاده از این تابع باید هدر فایل <math.h> را به برنامه افزود.

۷۰۰) تابع complex tanh(complex x) برای محاسبه مقدار تانژانت هایپربولیک عدد x می باشد. برای استفاده از این تابع باید هدر فایل <complex.h> را به برنامه افزود.

۷۰۱) تابع long tell(int handle) برای مشخص کردن مکان اشاره گر فایل در فایل جاری می باشد. اگر تابع موفق باشد مقدار بازگشتی آدرس اشاره گر را بر میگرداند و در غیر اینصورت مقدار 1- را بر می گرداند. برای استفاده از این تابع باید هدر فایل <io.h> را به برنامه افزود.

۷۰۲) تابع void textattr(int newattr) برای مشخص کردن صفت متن در خروجی استفاده می شود. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۷۰۳) تابع void textbackground(int newcolor) برای مشخص کردنرنگ پس زمینه متن در خروجی استفاده می شود. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۷۰۴) تابع void textcolor(int newcolor) برای مشخص کردن رنگ متن در خروجی استفاده می شود. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۷۰۵) تابع int far textheight(char far *textstring) در مود گرافیکی ارتفاع متن انتخابی را بر میگرداند، برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۷۰۶) تابع int far textwidth(char far *textstring) در مود گرافیکی عرض متن انتخابی را بر میگرداند، برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۰۷) تابع void textmode(int newmode)برای تغییر مدل نمایش متن در صفحه نمایش به کار می رود. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۷۰۸) ثابت LASTMODE با مقدار 1- برای برگرداندن حالت نمایش متن به حالت قبل می باشد. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۷۰۹) ثابت BW40 با مقدار ۰ برای حالت متن سیاه و سفید در ۴۰ ستون می باشد. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۷۱۰) ثابت C40 با مقدار ۱ برای حالت رنگی در ۴۰ ستون می باشد. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۷۱۱) ثابت BW80 با مقدار ۲ برای حالت متن سیاه و سفید در ۸۰ ستون می باشد. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۷۱۲) ثابت C80 با مدار ۳ برای حالت رنگی در ۸۰ ستون می باشد. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۷۱۳) ثابت MONO با مقدار ۷ برای حالت تک رنگ در ۸۰ ستون می باشد. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۷۱۴) ثابت C4350 با مقدار ۶۴ در دو مود گرافیکی EGA با ۴۳ سطر و در مود گرافیکی VGA با ۵۰ سطر میباشد. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۷۱۵) ثابت LEFT_TEXT برای چپ چین کردن متن در حالت گرافیکی و با مقدار ۰ و به صورت افقی می باشد. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۱۶) ثابت CENTER_TEXT برای وسط چین کردن متن در حالت گرافیکی و با مقدار ۱ و به صورت افقی می باشد. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۱۷) ثابت RIGHT_TEXT برای راست چین کردن متن در حالت گرافیکی و با مقدار ۲ و به صورت افقی می باشد. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۱۸) ثابت BOTTOM_TEXT برای پایین بردن متن در حالت گرافیکی و با مقدار ۰ و به صورت عمودی می باشد. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۱۹) ثابت CENTER_TEXT برای مرکزیت دادن به متن در حالت گرافیکی و با مقدار ۱ و به صورت عمودی می باشد. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۲۰) ثابت TOP_TEXT برای بالا بردن متن در حالت گرافیکی و با مقدار ۲ و به صورت عمودی می باشد. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۲۱) تابع int tolower(int ch) برای تبدیل یک حرف بزرگ به حرف کوچک می باشد. برای استفاده از این تابع باید هدر فایل <ctype.h> را به برنامه افزود.

۷۲۲) تابع int _tolower(int ch) برای تبدیل یک حرف بزرگ به حرف کوچک می باشد.این تابع به صورت ماکرو عمل می نماید. برای استفاده از این تابع باید هدر فایل <ctype.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۷۲۳) تابع int toupper(int ch) برای تبدیل یک حرف کوچک به حرف بزرگ می باشد. برای استفاده از این تابع باید هدر فایل <ctype.h> را به برنامه افزود.

۷۲۴) تابع int _toupper(int ch) برای تبدیل یک حرف کوچک به حرف بزرگ می باشد.این تابع به صورت ماکرو عمل می نماید. برای استفاده از این تابع باید هدر فایل <ctype.h> را به برنامه افزود.

۷۲۵) ثابت DEFAULT_FONT با مقدار ۰ برای استفاده از فونت پیش فرض استفاده می شود. برای استفاده از این ثابت باید هدر فایل <graphics.h> را به برنامه افزود.

۷۲۶) ثابت TRIPLEX_FONT با مقدار 1 برای استفاده از فونت با سایز سه برابر پیش فرض استفاده می شود. برای استفاده از این ثابت باید هدر فایل <graphics.h> را به برنامه افزود.

۷۲۷) ثابت SMALL_FONT با مقدار 2 برای استفاده از فونت کوچک استفاده می شود. برای استفاده از این ثابت باید هدر فایل <graphics.h> را به برنامه افزود.

۷۲۸) ثابت SANS_SERIF_FONT با مقدار 3 برای استفاده از فونت sans-serif استفاده می شود. برای استفاده از این ثابت باید هدر فایل <graphics.h> را به برنامه افزود.

۷۲۹) ثابت GOTHIC_FONT با مقدار 4 برای استفاده از فونتی که زبان گوتیک را پشتیبانی می کند استفاده می شود. برای استفاده از این ثابت باید هدر فایل <graphics.h> را به برنامه افزود.

۷۳۰) هنگامی که بخواهیم درزبان یک نوع داده ای جدید را تعریف کنیم، باید ازدستور typedef استفاده شود.

۷۳۱) برای تعریف متغیری که بتوان آن را بصورت متغیر خارجی استفاده کنیم باید از کلاس حافظه extern متغیر را تعریف کنیم.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۷۳۲) ثابت UCHAR_MAX برای بدست آوردن بیشترین مقدار نوع داده ای unsigned char استفاده می شود. برای استفاده از این ثابت باید هدر فایل <limits.h> را به برنامه افزود.

۷۳۳) ثابت USHORT_MAX برای بدست آوردن بیشترین مقدار نوع داده ای unsigned short استفاده می شود. برای استفاده از این ثابت باید هدر فایل <limits.h> را به برنامه افزود.

۷۳۴) ثابت UINT_MAX برای بدست آوردن بیشترین مقدار نوع داده ای unsigned int استفاده می شود. برای استفاده از این ثابت باید هدر فایل <limits.h> را به برنامه افزود.

۷۳۵) ثابت ULONG_MAX برای بدست آوردن بیشترین مقدار نوع داده ای unsigned long استفاده می شود. برای استفاده از این ثابت باید هدر فایل <limits.h> را به برنامه افزود.

۷۳۶) تابع unsigned umask(unsigned modemask) برای تعیین نوع پوشش فایل استفاده می شود. برای استفاده از این تابع باید هدر فایل <io.h> را به برنامه افزود.

۷۳۷) برای تعریف یک نوع داده ای جدید باید شما یک ساختار را از نوع union تعریف نمایید.

۷۳۸) اگر نوع داده ای جدید را از نوع union تعریف کنید ، در هر زمان فقط می توانید به یکی از زیر مجموعه های این union دسترسی داشته باشید.

۷۳۹) تابع void unixtodos(long time,struct date *d,struct time *t) برای تبدیل زمان از unix به dos می باشد. به این شکل که تاریخ و زمان را در پارامتر های دوم و سوم دهیره خواهد کرد. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۷۴۰) تابع int unlink(const char *filename) برای delete کردن فایل به کار می رود، آدرس فایل را در پارامتر تابع قرار داده و تابع فایل مورد نظر را حذف می کند. برای استفاده از این تابع باید هدر فایل های <io.h> یا <dos.h> یا <stdio.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۷۴۱) تابع int unlock (int handle , long offset , long length) برای باز کردن فایلی که قبلا برای share کردن قفل شده است به کار می رود. برای استفاده از این تابع باید هدر فایل <io.h> را به برنامه افزود.

۷۴۲) ثابت SOLID_LINE با مقدار ۰ در مود گرافیکی برای ترسیم خط یکپارچه استفاده می شود. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۴۳) ثابت DOTTED_LINE با مقدار ۱ برای ترسیم خط های نقطه چین استفاده می شود. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۴۴) ثابت CENTER_LINE با مقدار ۲ برای ترسیم خط های بریده بردیده نابرابر استفاده می شود. به این شکل که خطوط بریده بریده یک د ر میان کوتاه و بلند می باشند. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۴۵) ثابت DASHED_LINE با مقدار ۳ برای ترسیم خط های بریده بریده برابر استفاده میشود. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۴۶) ثابت USERBIT_LINE با مقدار ۴ برای ترسیم خط با نقاط دور از هم استفاده می شود. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۴۷) برای به کار بردن فونتی که کاراکتر ها را با اندازه ۸*۸ در نظر بگیرد به جای ثابت USER_CHAR_SIZE باید مقدار ۱ را قرار دهید.

۷۴۸) برای به کار بردن فونتی که کاراکتر ها را با اندازه ۱۶*۱۶ در نظر بگیرد به جای ثابت USER_CHAR_SIZE باید مقدار ۲ را قرار دهید.

۷۴۹) برای به کار بردن فونتی که کاراکتر ها را با اندازه ۲۴*۲۴ در نظر بگیرد به جای ثابت USER_CHAR_SIZE باید مقدار ۳ را قرار دهید.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۷۵۰) برای به کار بردن فونتی که کاراکتر ها را با اندازه ۳۲*۳۲ در نظر بگیرد به جای ثابت USER_CHAR_SIZE باید مقدار ۴ را قرار دهید.

۷۵۱) برای به کار بردن فونتی که کاراکتر ها را با اندازه ۴۰*۴۰ در نظر بگیرد به جای ثابت USER_CHAR_SIZE باید مقدار ۵ را قرار دهید.

۷۵۲) برای به کار بردن فونتی که کاراکتر ها را با اندازه ۴۸*۴۸ در نظر بگیرد به جای ثابت USER_CHAR_SIZE باید مقدار ۶ را قرار دهید.

۷۵۳) برای به کار بردن فونتی که کاراکتر ها را با اندازه ۵۶*۵۶ در نظر بگیرد به جای ثابت USER_CHAR_SIZE باید مقدار ۷ را قرار دهید.

۷۵۴) برای به کار بردن فونتی که کاراکتر ها را با اندازه ۶۴*۶۴ در نظر بگیرد به جای ثابت USER_CHAR_SIZE باید مقدار ۸ را قرار دهید.

۷۵۵) برای به کار بردن فونتی که کاراکتر ها را با اندازه ۷۲*۷۲ در نظر بگیرد به جای ثابت USER_CHAR_SIZE باید مقدار ۹ را قرار دهید.

۷۵۶) برای به کار بردن فونتی که کاراکتر ها را با اندازه ۸۰*۸۰ در نظر بگیرد به جای ثابت USER_CHAR_SIZE باید مقدار ۱۰ را قرار دهید.

۷۵۷) ثابت EMPTY_FILL با مقدار ۰ در مود گرافیکی برای پر کردن درون شکل گرافیکی به صورت خالی استفاده می شود. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۵۸) ثابت SOLID_FILL با مقدار 1 در مود گرافیکی برای پر کردن درون شکل گرافیکی به صورت کامل و یک پارچه استفاده می شود. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۷۵۹) ثابت LINE_FILL با مقدار 2 در مود گرافیکی برای پر کردن درون شکل گرافیکی به صورت خطهای افقی استفاده می شود. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۶۰) ثابت LTSLASH_FILL با مقدار 3 در مود گرافیکی برای پر کردن درون شکل گرافیکی به صورت /// استفاده می شود. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۶۱) ثابت SLASH_FILL با مقدار 4 در مود گرافیکی برای پر کردن درون شکل گرافیکی به صورت /// ضخیم استفاده می شود. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۶۲) ثابت BKSLASH_FILL با مقدار 5 در مود گرافیکی برای پر کردن درون شکل گرافیکی به صورت \\\ ضخیم استفاده می شود. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۶۳) ثابت BKSLASH_FILL با مقدار 5 در مود گرافیکی برای پر کردن درون شکل گرافیکی به صورت \\\ ضخیم استفاده می شود. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۶۴) ثابت HATCH_FILL با مقدار 7 در مود گرافیکی برای پر کردن درون شکل گرافیکی به صورت شبکه های مربعی استفاده می شود. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود. ثابت XHATCH_FILL با مقدار 8 در مود گرافیکی برای پر کردن درون شکل گرافیکی به صورت شبکه های لوزی استفاده می شود. برای استفاده از این تابع باید هدر فایل <graphics.h>را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۷۶۵) ثابت INTERLEAVE_FILL با مقدار ۹ در مود گرافیکی برای پر کردن درون شکل گرافیکی به صورت بافت ریز استفاده می شود. برای استفاده ازاین تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۶۶) ثابت WIDE_DOT_FILL با مقدار ۱۰ در مود گرافیکی برای پر کردن درون شکل گرافیکی به صورت نقاط بافاصله استفاده می شود.برای استفاده ازاین تابع بایدهدرفایل<graphics.h> رابه برنامه افزود.

۷۶۷) ثابت CLOSE_DOT_FILL با مقدار 11 در مود گرافیکی برای پر کردن درون شکل گرافیکی به صورت نقاط نزدیک استفاده می شود.برای استفاده ازاین تابع بایدهدر فایل<graphics.h>را به برنامه افزود.

۷۶۸) ثابت USER_FILL با مقدار 12 در مود گرافیکی برای پر کردن درون شکل گرافیکی به صورت دلخواه کاربر استفاده می شود. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۶۹) ثابت DETECT با مقدار ۰ برای تعیین درایور گرافیکی به صورت اتوماتیک و خودکار می باشد. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۷۰) ثابت CGA با مقدار 1 برای تعیین درایور گرافیکی از نوع CGA می باشد. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۷۱) ثابت MCGA با مقدار 2 برای تعیین درایور گرافیکی از نوع MCGA می باشد. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۷۲) ثابت EGA با مقدار 3 برای تعیین درایور گرافیکی از نوع EGA می باشد. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۷۷۳) ثابت EGA64 با مقدار 4 برای تعیین درایور گرافیکی از نوع EGA64 می باشد. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۷۴) ثابت EGAMONO با مقدار 5 برای تعیین درایور گرافیکی از نوع EGAMONO می باشد. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۷۵) ثابت IBM8514 با مقدار 6 برای تعیین درایور گرافیکی از نوع IBM8514 می باشد. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۷۶) ثابت HERCMONO با مقدار 7 برای تعیین درایور گرافیکی از نوع HERCMONO می باشد. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۷۷) ثابت ATT400 با مقدار 8 برای تعیین درایور گرافیکی از نوع ATT400 می باشد. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۷۸) ثابت VGA با مقدار 9 برای تعیین درایور گرافیکی از نوع VGA می باشد. برای استفاده از این تابع باید ەدر فایل <graphics.h> را به برنامه افزود.

۷۷۹) ثابت PC3270 با مقدار 10 برای تعیین درایور گرافیکی از نوع PC3270 می باشد. برای استفاده از این تابع باید هدر فایل <graphics.h> را به برنامه افزود.

۷۸۰) در زبان C++ شما می توانید متغیرهایی تعریف کنید که به صورت عادی پوچ هستند و تا قابلیت آن را دارند که هر مقداری را که به آن بدهید خود را همانند سازی کنند. این نوع داده ای void می باشد.

۷۸۱) تابع int wherex(void) طول مکان فعلی کرسر را بدست می آورد. برای استفاده از این تابع بلید هدر فایل <conio.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۷۸۲) ثابت wamb– باعث نمایش پیغام Ambiguous operators need parantheses میشود. به این معنی که استفاده از عملگر مورد نظر بدون پرانتز باعث ابهام در برنامه می شود.

۷۸۳) ثابت wamp– باعث نمایش پیغام Super fluous & with function می شود. به این معنی که آدرس عملگرها (&) است و نیازی به ذکر نام تابع نیست.

۷۸۴) ثابت wasm- باعث نمایش پیغام Unknown assembler instruction می شود. به این معنی که کامپایلر با بیانیه مونتاژ خطی opcode disallowed مواجه شده است.

۷۸۵) ثابت waus- باعث نمایش پیغام ’identifier’ is assigned a value that is never used می شود. به ای معنی که شما متغیری را تعریف کرده اید و آن را هیچگاه استفاده نکرده اید.

۷۸۶) ثابت wbbf- باعث نمایش پیغام Bit fields must be signed or unsigned int می شود. به این معنی که رشته های بیتی ممکن است از نوع کاراکتر با علامت و یا کاراکتر بدون علامت باشد.

۷۸۷) ثابت wbei- باعث نمایش پیغام Initializing ‘enumeration’ with ‘type’ می شود. به این معنی که شما سعی در initialize به متغیر شمارشی از نوع متفاوت هستید.

۷۸۸) ثابت wbig- باعث نمایش پیغام Hexadecimal value contains more than 3 digits می شود. در کامپایلر های قدیمی تر C شما نمیتوانستید به یک مغیر از نوع hexadecimal بیش از ۳ رقم بدهید.

۷۸۹) ثابت wccc- باعث نمایش پیغام Condition Is always true می شود. به این معنی که کامپایلر به شرطی برخورد کرده که مقدار شرط همیشه true خواهد بود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۷۹۰) ثابت wccc- باعث نمایش پیغام Condition is always false می شود. به این معنی که کامپایلر به شرطی برخورد کرده که مقدار شرط همیشه false خواهد بود.

۷۹۱) ثابت wcln- باعث نمایش پیغام Constant is long می شود. به این معنی که کامپایلر با یک ثابت تعریف شده برخورد کرده که مقدار آن بیش از حد بزرگ بوده است.

۷۹۲) ثابت wcpt- باعث نمایش پیغام Nonportable pointer comparison می شود. به این معنی که برنامه شما قصد مقایسه کردن یک اشاره گر و یک غیر اشاره گر را با هم دارد.

۷۹۳) ثابت wdef- باعث نمایش پیغام Possible use of ‘identifir’ befor definition می شود. به این معنی که برنامه قصد استفاده از متغیری را دارد که هنوز مقدار نگرفته است.

۷۹۴) ثابت wdpu- باعث نمایش پیغام Declare type ‘type’ prior to use in prototype می شود. به این معنی که شما میخواهید از نمونه به عنوان نوع استفاده کنید.

۷۹۵) ثابت wdsz- باعث نمایش پیغام Array size for ‘delete’ ignored می شود. زمانی این هشدار داده می شود که شما حجم آرایه ای را که می خواهید حذف کنید را مشخص کرده باشید.

۷۹۶) ثابت wdup- باعث نمایش پیغام Redefinition of ‘macro’ is not identical می شود. به این معنی که تعریف دوباره ای که از ماکرو انجام شده یکسان نمی باشد.

۷۹۷) ثابت weas- باعث نمایش پیغام Assigning ‘type’ to ‘enumeration’ می شود. به این معنی که شما می خواهید یک مقدار را به نوع شمارشی اختصاص دهید.

۷۹۸) ثابت weff- باعث نمایش پیغام Code has no effect می شود. به این معنی که کد وارد شده اثری ندارد.

۷۹۹) ثابت wext- باعث نمایش پیغام ’identifier’ is declared as both external and static می شود. به این معنی که شما یک متغیر را هم با کلاس حافظه extern و هم با کلاس حافظه static همزمان تعریف کرده اید.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۸۰۰) ثابت whid- باعث نمایش پیغام 'functiona1' hides virtual function 'function2' می شود. به این معنی که ممکن است تابع۱ تابع مجازی که به عنوان تابع۲تعریف کرده ایدرا بپوشاند وتابع۲اجرا نشود.

۸۰۱) ثابت wias- باعث نمایش پیغام Array variable 'identifier' is near می شود. به این معنی که کامپایلر نمیتواند تشخیص دهد که آرایه باید far تعریف شود یا near و کامپایلر به طور پیش فرض قصد تعریف near را دارد.

۸۰۲) ثابت wibc- باعث نمایش پیغام Base class 'class' is inaccessible because also in 'class' می شود. به این معنی که کلاس پایه غیر قابل دسترسی می باشد.

۸۰۳) ثابت will- باعث نمایش پیغام ILL-formed pragma می شود. به این معنی که شکل pragma درست نمی باشد.

۸۰۴) ثابت winl- باعث نمایش پیغام Functions containing local destructors are not expanded inline in function 'function' می شود. به این معنی که توابع مخربی در برنامه هستند که ممکن است باعث مشکل شود.

۸۰۵) ثابت winl- باعث نمایش پیغام Functions containing reserved words are not expended inline می شود. توابع حاوی کلمات کلیدی هستند که باعث ایجاد توابع inline میگردد.

۸۰۶) ثابت wlin- باعث نمایش پیغام Temporary used to initialize 'identifier' می شود. Temporary برای initialize شناسه استفاده شده است.

۸۰۷) ثابت wlvc- باعث نمایش پیغام Temporary used for parameter 'number' می شود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۸۰۸) ثابت wlvc- باعث نمایش پیغام Temporary used for parameter ‘number’ in call to ‘function’ می شود.

۸۰۹) ثابت wlvc- باعث نمایش پیغام Temporary used for parameter ‘parameter’ می شود.

۸۱۰) ثابت wlvc- باعث نمایش پیغام Temporary used for parameter ‘parameter’ in call to ‘function’ می شود.

۸۱۱) ثابت wmpc- باعث نمایش پیغام Conversion to ‘type’ will for members of virtual base ‘class’ می شود. به این معنی که اعضای پایگاه مجازی کلاس به نوع تبدیل خواهند شد.

۸۱۲) ثابت wmpd- باعث نمایش پیغام Maximum precision used for member pointer type ‘type’ می شود. به این معنی که حداکثر دقت خود را برای اشاره گر مذکور به کار بندید.

۸۱۳) ثابت wncf- باعث نمایش پیغام Non-constant function ‘function’ called for constatnt object می شود. تابع غیر ثابت مذکور برای اشیای ثابت فراخوانی می شود.

۸۱۴) ثابت wncf- باعث نمایش پیغام Non-volatile function ‘function’ called for volatile object می شود. تابع فرار برای اشیای غیر فرار فراخوانی می شود.

۸۱۵) ثابت wnci- باعث نمایش پیغام Constant member ‘member’ is not initialized می شود.عضو ثابتی برای member در برنامه initialized موجود نیست.

۸۱۶) ثابت wnod- باعث نمایش پیغام No delaration for function ‘function’ می شود. تابع مذکور اعلان نشده است.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۸۱۷) ثابت wnst- باعث نمایش پیغام Use qualified name to access nested type 'type' می شود. برای دسترسی به نوع تو در تو باید نام معتبر تعریف نمایید.

۸۱۸) ثابت wobi- باعث نمایش پیغام Base initialization without a class name is now obsolete می شود. پایه آغازین بدون نام کلاس می باشد و در حال حاضر منسوخ می باشد.

۸۱۹) ثابت wofp- باعث نمایش پیغام Style of function definition is now obsolete می شود. استراتژی تعریف تابع در حال حاضر منسوخ است.

۸۲۰) ثابت wore- باعث نمایش پیغام Overloaded prefix 'operator' used as a postfix operator می شود. عملگر پیشوندی مذکور به جای عملگر پسوندی استفاده می شود.

۸۲۱) ثابت wovl- باعث نمایش پیغام 'overload' is now unnecessary and obsolete می شود. بارگذاری دوباره غیر ضروری بوده و در حال حاضر منسوخ است.

۸۲۲) ثابت wpar- باعث نمایش پیغام Parameter 'parameter' is never used می شود. پارامتر تعریف شده هرگز استفاده نشده است.

۸۲۳) ثابت wpia- باعث نمایش پیغام Possibly incorrect assignment می شود. ارجاع داده شده نادرست می باشد.

۸۲۴) ثابت wpin- باعث نمایش پیغام Initialization is only partially bracketed می شود. شروع برنامه تاحدی bracketed می باشد.

۸۲۵) ثابت wpro- باعث نمایش پیغام Call to function 'function' with no prototype می شود.تابعی را که تعریف نشده فراخوانی کرده اید.

۸۲۶) ثابت wpro- باعث نمایش پیغام call to function with no prototype می شود. فراخوانی تابع بدون تعریف تابع انجام گرفته است.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۸۲۷) ثابت wrch- باعث نمایش پیغام Unreachable code می شود. کد نوشته شده بدون حاصل می باشد.

۸۲۸) ثابت wret- باعث نمایش پیغام Both return and return with value used می شود. return بدون مقدار و return بامقدار هر دو در برنامه وجود دارد.

۸۲۹) ثابت wrng- باعث نمایش پیغام Constant out of range in comparison می شود. ثابت تعریف شده در مقایسه مقدار بیش از به خود گرفته.

۸۳۰) ثابت wrpt- باعث نمایش پیغام Nonportable pointer conversion می شود. تبدیل اشاره گر Nonportable صورت گرفته است.

۸۳۱) ثابت wrvl- باعث نمایش پیغام Function should return a value می شود. تابع باید یک مقدار بازگشتی داشته باشد.

۸۳۲) ثابت wsig- باعث نمایش پیغام Conversion may lose significant digits می شود. در این تبدیل ممکن است مقدار قابل توجهی از بین برود.

۸۳۳) ثابت wstu- باعث نمایش پیغام Undefined structure ‘structure’ می شود. ساختار مذکور تعریف نشده است.

۸۳۴) ثابت wstv- باعث نمایش پیغام Structure passed by value می شود. ساختار با مقدار سپری شده است.

۸۳۵) ثابت wsus- باعث نمایش پیغام Suspicious pointer conversion می شود. تبدیل اشاره گر مشکوک صورت گرفته است.

۸۳۶) ثابت wucp- باعث نمایش پیغام Mixing pointers to signed and unsigned char می شود. مخلوط کردن اشاره گر با مقدارهای علامتدار و بی علامت صورت گرفته است.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۸۳۷) ثابت wuse- باعث نمایش پیغام 'identifier' declared but never used می شود. متغیری را که تعریف کرده اید استفاده نکرده اید.

۸۳۸) ثابت wvoi- باعث نمایش پیغام Void function may not return a value می شود. تابعی که با void تعریف می شود ممکن نمی تواند مقداری را برگرداند.

۸۳۹) ثابت wzdi- باعث نمایش پیغام Division by zero می شود. احتمال تقسیم بر ۰ وجود دارد.

۸۴۰) تابع int wherey(void) عرض مکان فعلی کرسر را بدست می آورد. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۸۴۱) در برنامه ها برای ایجاد حلقه هایی که تعداد تکرار نامعلوم دارند از حلقه while استفاده میشود.

۸۴۲) ثابتWHITE با مقدار ۱۵ در مود گرافیکی به معنای رنگ سفید میباشد. برای استفاده از این ثابت باید هدر فایل <graphics.h> را به برنامه افزود.

۸۴۳) تابع void window(int left , int top , int right , int bottom) برای تعریف یک پنجره فعال متن به کار میرود. برای استفاده از این تابع باید هدر فایل <conio.h> را به برنامه افزود.

۸۴۴) تابع int _write(int handle , void *buf , unsigned len) برای نوشتن در فایل استفاده می شود. شماره فایل را در پارامتر اول مشخص کرده ، سپس در پارامتر دوم آدرس اطلاعات و در پارامتر سوم طول اطلاعات را تعیین میکنید. برای استفاده از این ثابت باید هدر فایل <io.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۸۴۵) تابع int write(int handle , void *buf , unsigned len) برای نوشتن اطلاعات در فال استفاده می شود. شماره فایل را در پارامتر اول مشخص کرده ، سپس در پارامتر دوم آدرس اطلاعات و در پارامتر سوم طول اطلاعات را تعیین میکنید. برای استفاده از این ثابت باید هدر فایل <io.h> را به برنامه افزود.

۸۴۶) ثابتYELLOW با مقدار 14 در مود گرافیکی به معنای رنگ زرد میباشد. برای استفاده از این ثابت باید هدر فایل <graphics.h> را به برنامه افزود.

۸۴۷) ثابتLIGHTMAGENTA با مقدار 13 در مود گرافیکی به معنای رنگ بنفش میباشد. برای استفاده از این ثابت باید هدر فایل <graphics.h> را به برنامه افزود.

۸۴۸) ثابتLIGHTED با مقدار 12 در مود گرافیکی به معنای رنگ قرمز میباشد. برای استفاده از این ثابت باید هدر فایل <graphics.h> را به برنامه افزود.

۸۴۹) ثابتLIGHTCYAN با مقدار 11 در مود گرافیکی به معنای رنگ فیروزه ای میباشد. برای استفاده از این ثابت باید هدر فایل <graphics.h> را به برنامه افزود.

۸۵۰) ثابتLIGHTGREEN با مقدار 10 در مود گرافیکی به معنای رنگ سبز میباشد. برای استفاده از این ثابت باید هدر فایل <graphics.h> را به برنامه افزود.

۸۵۱) ثابتLIGHTBLUE با مقدار 9 در مود گرافیکی به معنای رنگ آبی میباشد. برای استفاده از این ثابت باید هدر فایل <graphics.h> را به برنامه افزود.

۸۵۲) ثابتDARKGRAY با مقدار 8 در مود گرافیکی به معنای رنگ خاکستری تیره میباشد. برای استفاده از این ثابت باید هدر فایل <graphics.h> را به برنامه افزود.

۸۵۳) ثابتLIGHTGRAY با مقدار 7 در مود گرافیکی به معنای رنگ خاکستری روشن میباشد. برای استفاده از این ثابت باید هدر فایل <graphics.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۸۵۴) ثابتBROWN با مقدار 6 در مود گرافیکی به معنای رنگ قهوه ای میباشد. برای استفاده از این ثابت باید هدر فایل <graphics.h> را به برنامه افزود.

۸۵۵) ثابتMAGENTA با مقدار 5 در مود گرافیکی به معنای رنگ بنفش میباشد. برای استفاده از این ثابت باید هدر فایل <graphics.h> را به برنامه افزود.

۸۵۶) ثابتRED با مقدار 4 در مود گرافیکی به معنای رنگ قرمز میباشد. برای استفاده از این ثابت باید هدر فایل <graphics.h> را به برنامه افزود.

۸۵۷) ثابتCYAN با مقدار 3 در مود گرافیکی به معنای رنگ فیروزه ای میباشد. برای استفاده از این ثابت باید هدر فایل <graphics.h> را به برنامه افزود.

۸۵۸) ثابتGREEN با مقدار 2 در مود گرافیکی به معنای رنگ سبز میباشد. برای استفاده از این ثابت باید هدر فایل <graphics.h> را به برنامه افزود.

۸۵۹) ثابتBLUE با مقدار 1 در مود گرافیکی به معنای رنگ آبی میباشد. برای استفاده از این ثابت باید هدر فایل <graphics.h> را به برنامه افزود.

۸۶۰) ثابتBLACK با مقدار 0 در مود گرافیکی به معنای رنگ مشکی میباشد. برای استفاده از این ثابت باید هدر فایل <graphics.h> را به برنامه افزود.

۸۶۱) هنگامی که بخواهیم به یکی از عناصر یک ساختار دسترسی داشته باشیم، از عملگر(.) استفاده می کنیم.

۸۶۲) هنگامی که بخواهیم به صورت اشاره گری به عناصر یک ساختار دسترسی داشته باشیم، از عملگر(<-) استفاده می کنیم.

۸۶۳) برای دسترسی به یک عملوند عمومی از عملگر (::) استفاده میکنیم.

۸۶۴) برای اشاره گر بدون مرجع به یک کلاس از عملگر (*.)استفاده میکنیم.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۸۶۵) برای اشاره گر به اشاره گر بدون مرجع به یک کلاس از عملگر (*<-)استفاده می کنیم.

۸۶۶) هنگامی که بخواهیم یک پارامتر از یک تابع را بدون مقدار گذاریم باید از عملگر (...) استفاده کنیم.

۸۶۷) برای نشان کردن یک عبارت در برنامه از عملگر (:) استفاده میکنیم.

۸۶۸) برای تعریف آرایه از عملگر ([ ]) استفاده میکنیم.

۸۶۹) ثابت E2BIG ، خطای dos می باشد و هنگامی که شما در یک محیط کاری را انجام دهید که مناسب آن محیط نمی باشد که پیغام خطای آن نیز Bad environ می باشد. برای آگاهی از این خطا در برنامه باید یکی از هدر فایل های <dos.h> یا <errno.h> یا <stdlib.h> را به برنامه افزود.

۸۷۰) ثابت EACCES خطای dos می باشد و(۱) هنگامی که شما بخواهید به آدرسی که اجازه دسترسی شماتعریف نشده است دسترسی پیدا کنید که پیغام خطای آن نیز Access denied می باشد. برای آگاهی از این خطا در برنامه باید یکی از هدر فایل های <dos.h> یا <errno.h> یا <stdlib.h> را به برنامه افزود.

۸۷۱) ثابت EACCES خطای dos می باشد و(۲) هنگامی که دسترسی شما به یک آدرس مشکل داشته باشد چنین پیغام خطایی را صادر می کند که پیغام خطای آن نیز Bad access می باشد. برای آگاهی از این خطا در برنامه باید یکی از هدر فایل های <dos.h> یا <errno.h> یا <stdlib.h> را به برنامه افزود.

۸۷۲) ثابت EACCES خطای dos می باشد و(۳) هنگامی که دسترسی شما در آدرس جاری از بین برود چنین پیغام خطایی صادر می کند که پیغام خطای آن نیز Is current dir می باشد.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

برای آگاهی از این خطا در برنامه باید یکی از ەدر فایل های <dos.h> یا <errno.h> یا <stdlib.h> را به برنامه افزود.

۸۷۳) ثابت EBADF خطای dos می باشد و هنگامی که شماره فایل مورد نظر مشکل داشته باشد چنین پیغام خطایی صادر می کند و پیغام خطای آن نیز Bad handle می باشد. برای آگاهی از این خطا در برنامه باید یکی از هدر فایل های <dos.h> یا <errno.h> یا <stdlib.h> را به برنامه افزود.

۸۷۴) ثابت EFAULT خطای dos می باشد و هنگامی که فایل محافظت شده باشد چنین پیغام خطایی صادر می شود و پیغام خطای آن نیز Reserved می باشد. برای آگاهی از این خطا در برنامه باید یکی از هدر فایل های <dos.h> یا <errno.h> یا <stdlib.h> را به برنامه افزود.

۸۷۵) ثابت EINVAL خطای dos می باشد و(۱)هنگامی که اطلاعات وارد شده مشکل داشته باشند اتفاق می افتد و پیغام خطای آن نیز Bad data می باشد. برای آگاهی از این خطا در برنامه باید یکی از هدر فایل های <dos.h> یا <errno.h> یا <stdlib.h> را به برنامه افزود.

۸۷۶) ثابت EINVAL خطای dos می باشد و(۲)هنگامی که تابع فراخوانی شده مشکل داشته باشد اتفاق می افتد و پیغام خطای آن نیز Bad function می باشد. برای آگاهی از این خطا در برنامه باید یکی از هدر فایل های <dos.h> یا <errno.h> یا <stdlib.h> را به برنامه افزود.

۸۷۷) ثابت EINVAL خطای dos می باشد و (۳) هنگامی که تعداد فایل های باز در برنامه بیش از حد باشد اتفاق می افتد و پیغام خطای آن نیز Too many open می باشد. برای آگاهی از

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

این خطا در برنامه باید یکی از هدر فایل های <dos.h> یا <errno.h> یا <stdlib.h> را به برنامه افزود.

۸۷۸) ثابت ENOENT خطای dos می باشد و هنگامی که فایل و پوشه هایی از قبیل فایل و پوشه جاری در برنامه باشد که برای برنامه قابل شناسایی نیست چنین پیغام خطایی را صادر می کند که پیغام خطای آن نیز No such file or directory می باشد. برای آگاهی از این خطا در برنامه باید یکی از هدر فایل های <dos.h> یا <errno.h> یا <stdlib.h> را به برنامه افزود.

۸۷۹) ثابت ENOEXEC خطای dos می باشد و هنگامی که فرمت درخواستی برای برنامه قابل شناسایی نباشد چنین پیغام خطایی صادر می کند که پیغام خطای آن نیز Bad format می باشد. برای آگاهی از این خطا در برنامه باید یکی از هدر فایل های <dos.h> یا <errno.h> یا <stdlib.h> را به برنامه افزود.

۸۸۰) ثابت ENOMEM خطای dos میباشد و (۱) هنگامی که mcb خراب شده باشد چنین پیغام خطایی صادر می شود و پیغام خطای آن نیز Mcb destroyed می باشد. برای آگاهی از این خطا در برنامه باید یکی از هدر فایل های <dos.h> یا <errno.h> یا <stdlib.h> را به برنامه افزود.

۸۸۱) ثابت ENOMEM خطای dos می باشد و(۲) هنگامی که آدرس خواسته شده خارج از حد حافظه باشد پیغام خطای Out of memory صادر می شود. برای آگاهی از این خطا در برنامه باید یکی از هدر فایل های <dos.h> یا <errno.h> یا <stdlib.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۸۸۲) ثابت ENOMEM خطای dos می باشد و(۳)هنگامی که بلوک حافظه درخواستی با مشکل مواجه شود پیغام خطای Bad block را صادر می کند. برای آگاهی از این خطا در برنامه باید یکی از هدر فایل های <dos.h> یا <errno.h> یا <stdlib.h> را به برنامه افزود.

۸۸۳) ثابت EXDEV خطای dos می باشد و(۱) هنگامی که شما آدرس یک درایو مشکل داشته باشد با این خطا مواجه می شوید که پیغام خطای Bad drive را صادر می کند. برای آگاهی از این خطا در برنامه باید یکی از هدر فایل های <dos.h> یا <errno.h> یا <stdlib.h> را به برنامه افزود.

۸۸۴) ثابت EXDEV خطای dos می باشد و(۲) هنگامی که شما به یک درایو مشکل دار درخواست دسترسی داشته باشید چنین پیغامی مبنی بر اشکال در دسترسی به درایو های این چنینی را برنامه با پیغام خطای Not same driveصادر می کند. برای آگاهی از این خطا در برنامه باید یکی از هدر فایل های <dos.h> یا <errno.h> یا <stdlib.h> را به برنامه افزود.

۸۸۵) تابع int close(int handle) برای بستن یک فایل باز در برنامه استفاده می شود. برای استفاده از این تابع باید هدر فایل <io.h> را به برنامه افزود.

۸۸۶) تابع int _close(int handle) برای بستن یک فایل باز در برنامه استفاده می شود. برای استفاده از این تابع باید هدر فایل <io.h> را به برنامه افزود.

۸۸۷) تابع unsigned _dos_close(int handle) برای بستن یک فایل باز در برنامه استفاده می شود. فرق این تابع با دو تابع قبل در درج کردن کاراکتر ctrl+z در پایان فایل می باشد. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۸۸۸) تابع int creat(constant char *path, int amode) برای ایجاد یک فایل در آدرس مورد نظر با مود مورد نظر استفاده می شود. برای استفاده از این تابع باید هدر فایل <io.h> را به برنامه افزود.

۸۸۹) تابع int _creat(constant char *path , int attrib) برای ایجاد یک فایل در آدرس مورد نظر با صفت درخواستی می باشد. برای استفاده از این تابع باید هدر فایل <io.h> را به برنامه افزود.

۸۹۰) تابع unsigned _dos_creat(const char *path , int attrib , int *handlep) برای ایجاد یک فایل در آدرس مورد نظر و با صفت تعیین شده می باشد. این تابع با استفاده از پارامتر سوم یک اشاره گر به handle فایل ایجاد می کند و در اختیار برنامه نویس قرار می دهد. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۸۹۱) تابع int creatnew(const char *path , int mode) برای ایجاد یک فایل در آدرس مورد نظر و با صفت ذکر شده ب کارمی رود. برای استفاده از این تابع باید هدر فایل <io.h> را به برنامه افزود.

۸۹۲) تابع unsigned _dos_creatnew(const char *path , int attrib , int *handle) برای ایجاد یک فایل در آدرس مورد نظر و با صفت تعیین شده استفاده می شود. همچنین تابع با استفاده از پارامتر سوم یک اشاره گر به handle فایل ایجاد می کند و در اختیار برنامه نویس قرار می دهد. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۸۹۳) تابع unsigned _dos_findfirst(const char *pathname , int attrib , struct find_t *ffblk) برای جستجوی فایل یا پوشه مورد نظر در سیستم به کار می رود. نام فایل مورد نظر را در پارامتر اول می نویسیم و صفت ان فایل را در پارامتر دوم قرار می دهیم و

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

پارامتر سوم اشاره گری است به فایلی که ممکن است برنامه برای شما پیدا کند. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۸۹۴) تابع unsigned _dos_findnext(struct find_t *ffblk) در ادامه کار تابع قبل به کار می رود و هنگامی که تابع قبل اولین مورد را پیدا کند این تابع به دنبال موردبعدی از آدرس فایل اول پیدا شده می باشد. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۸۹۵) تابع int findfirst(const char *pathname , struct ffblk *ffblk , int attrib) برای جستجو در سیستم و پیدا کردن فایلی که در پارامتر اول مشخص شده است استفاده می شود و در پارامتر دوم اشاره گری به فایلی که احتمالا پیدا شده است قرار میگیرد. و در پارامتر آخر هم صفت فایل را قرار می دهیم. برای استفاده از این تابع باید هدر فایل <dir.h> را به برنامه افزود.

۸۹۶) تابع int findnext(struct ffblk *ffblk) در ادامه کار تابع قبل به کار می رود و هنگامی که تابع قبل اولین مورد را پیدا کند این تابع به دنبال مورد بعدی از آدرس فایل اول پیدا شده می باشد. برای استفاده از این تابع باید هدر فایل <dir.h> را به برنامه افزود.

۸۹۷) تابع unsigned _dos_freemem(unsigned segx) برای آزاد کردن فضایی که قبلا از حافظه به صورت پویا اخذ شده است به کار می رود. پارامتر تابع آدرس سگمنتی است که قرار است بلوک حافظه آن آدرس آزاد شود می باشد. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۸۹۸) تابع int freemem(unsigned segx) برای آزاد کردن فضایی که قبلا از حافظه به صورت پویا اخذ شده است به کار می رود. پارامتر تابع آدرس سگمنتی است که قرار است

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

بلوک حافظه آن آدرس آزاد شود می باشد. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۸۹۹) تابع void _dos_getdate(struct dosdate_t *datep) برای دریافت تاریخ سیستم به کار می رود. پارامتر تابع حاوی تاریخ کنونی سیستم خواهد شد. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۹۰۰) تابع void getdate(struct date *datep) برای دریافت تاریخ سیستم به کار می رود. پارامتر تابع حاوی تاریخ کنونی سیستم خواهد شد. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۹۰۱) تابع void _dos_setdate(struct dosdate_t *datep) برای تنظیم تاریخ سیستم به کار می رود. پارامتر تابع حاوی تاریخی است که می خواهیم تاریخ سیستم با آن تنظیم شود. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۹۰۲) تابع void setdate(struct date *datep) برای تنظیم تاریخ سیستم به کار می رود. پارامتر تابع حاوی تاریخی است که می خواهیم تاریخ سیستم با آن تنظیم شود. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

۹۰۳) تابع unsigned _dos_getdiskfree(unsigned char drive , struct diskfree_t *dtable) برای بدست آوردن فضای خالی یک درایو استفاده می شود. به این شکل که در پارامتر اول شماره درایو مورد نظر را وارد کرده سپس در پارامتر دوم یک متغیر از نوع تعریف شده قرار می دهید. برای پارامتر اول از درایو A=1 و B=2 و به همین صورت به ترتیب مقدار قرار دهید. برای استفاده از این تابع باید هدر فایل <dos.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۹۰۴) تابع unsigned getdfree(unsigned char drive , struct dfree*table) برای بدست آوردن فضای خالی یک درایو استفاده می شود. به این شکل که در پارامتر اول شماره درایو مورد نظر را وارد کرده سپس در پارامتر دوم یک متغیر از نوع تعریف شده قرار می دهید. برای پارامتر اول از درایو A=1 و B=2 و به همین صورت به ترتیب مقدار قرار دهید.برای استفاده ازاین تابع باید هدرفایل <dos.h> را به برنامه افزود.

۹۰۵) تابع int _chdrive(int drive) برای تنظیم درایو جاری استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <direct.h> را به برنامه افزود.

۹۰۶) تابع void _dos_getdrive(unsigned *drivep) برای بدست آوردن درایو جاری می باشد. برای استفاده ازاین تابع باید هدرفایل <dos.h> را به برنامه افزود.

۹۰۷) تابع unsigned _dos_setdrive(unsigned drive , unsigned *ndrives) برای تنظیم درایو جاری استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <dos.h> را به بونامه افزود.

۹۰۸) تابع int getdisk(void) برای بدست آوردن درایو جاری استفاده می سود. برای استفاده ازاین تابع باید هدرفایل <dir.h> را به برنامه افزود.

۹۰۹) تابع int _getdrive(void) برای بدست آوردن درایو جاری استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <direct.h> را به برنامه افزود.

۹۱۰) تابع int setdisk(int drive) برای بدست آوردن درایو جاری می باشد. برای استفاده ازاین تابع باید هدرفایل <dir.h> را به برنامه افزود.

۹۱۱) تابع int _chmod(const char *path , int func[,int attribp]) برای تنظیم و یا دریافت صفت یک فایل به کار می رود. پارامتر اول آدرس فایل و پارامتر دوم نوع عملیات را

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

مشخص می کند که اگر ۰ باشد به معنی دریافت صفت و اگر ۱ باشد به معنی تنظیم صفت می باشد. و در پارامتر سوم هم صفت مورد نظر را تعیین میکنیم برای استفاده ازاین تابع باید هدرفایل <io.h> را به برنامه افزود.

۹۱۲) تابع int _dos_getfileattr(const char *path , unsigned *attribp) برای دریافت نوع صفت فایل انتخابی که آدرسش در پارانتر اول آمده است استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <dos.h> را به برنامه افزود.

۹۱۳) تابع int _dos_setfileattr(const char *path , unsigned attrib) برای تنظیم صفت به یک فایل که آدرسش در پارامتر اول آمده است استفاده می شود که صفت انتخابی هم در پارامتر دوم تعیین می شود. برای استفاده ازاین تابع باید هدرفایل <dos.h> را به برنامه افزود.

۹۱۴) تابع unsigned _dos_getftime(int handle , unsigned *datep , unsigned *timep) برای بدست آوردن آخرین تاریخ و زمان دستکاری شدن بر روی یک فایل استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <dos.h> را به برنامه افزود.

۹۱۵) تابع unsigned _dos_setftime(int handle , unsigned date , unsigned time) برای تنظیم آخرین تاریخ و زمان دستکاری بر روی فایل به کار می رود. برای استفاده ازاین تابع باید هدرفایل <dos.h> را به برنامه افزود.

۹۱۶) تابع int getftime(int handle , struct ftime *ftimep) برای بدست آوردن آخرین تاریخ و زمان دستکاری شدن بر روی فایل را بدست می آورد. برای استفاده ازاین تابع باید هدرفایل <io.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۹۱۷) تابع int setftime(int handle , struct ftime *ftimep) برای تنظیم آخرین تاریخ و زمان دستکاری بر روی فایل به کار می رود. برای استفاده ازاین تابع باید هدرفایل <dos.h> را به برنامه افزود.

۹۱۸) تابع void _dos_gettime(struct dostime_t *timep) برای دریافت زمان کنونی سیستم به کار می رود. برای استفاده ازاین تابع باید هدرفایل <dos.h> را به برنامه افزود.

۹۱۹) تابع void _dos_settime(struct dostime_t *timep) برای تنظیم زمان سیستم استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <dos.h> را به برنامه افزود.

۹۲۰) تابع unsigned _dos_open(const char *filename , unsigned oflags , int *handlep) برای بازکردن یک فایل برای خواندن و یا نوشتن به کار می رود. در پارامتر اول نام فایل و نوع باز کردن فایل را در پارامتر دوم مشخص کرده و در پارامتر سوم یک‌اشاره گر به handle را در اختیار برنامه نویس قرار می دهد. برای استفاده ازاین تابع باید هدرفایل <dos.h> را به برنامه افزود.

۹۲۱) تابع int _open(const char *filename , int oflags) برای باز کردن یک فایل برای نوشتن و یا خواندن استفاده می شود. پارامتر اول نام فایل را تعیین می کند و در پارامتر دوم تعیین نوع باز کردن فایل را می کند. برای استفاده ازاین تابع باید هدرفایل <io.h> را به برنامه افزود.

۹۲۲) تابع unsigned _dos_read(int handle , void far *buf , unsigned len , unsigned *nread) برای خواندن چندین بایت از یک فایل استفاده می شود. حداکثر تا ۶۵۵۳۵ بایت را می توان خواند. برای استفاده ازاین تابع باید هدرفایل <dos.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۹۲۳) تابع int _read(int handle , void *buf , unsigned len) برای خواندن چندین بایت از یک فایل استفاده می شود. حداکثر تا ۶۵۵۳۴ بایت را می توان خواند. برای استفاده ازاین تابع باید هدرفایل <dos.h> را به برنامه افزود.

۹۲۴) تابع int read(int handle , void *buf , unsigned len) برای خواندن چندین بایت از یک فایل استفاده می شود. حداکثر تا ۶۵۵۳۴ بایت را می توان خواند. برای استفاده ازاین تابع باید هدرفایل <io.h> را به برنامه افزود.

۹۲۵) تابع double abs(complex) برای محاسبه مقدار قدر مطلق یک مقدار مختلط به کار می رود. برای استفاده ازاین تابع باید هدرفایل <complex.h> را به برنامه افزود.

۹۲۶) تابع double cabs(struct complex z) برای محاسبه مقدار قدر مطلق یک عدد مختلط به کار می رود. برای استفاده ازاین تابع باید هدرفایل <complex.h> را به برنامه افزود.

۹۲۷) تابع long double cabsl(struct complex z) برای محاسبه مقدار قدر مطلق یک عدد مختلط به کار می رود. برای استفاده ازاین تابع باید هدرفایل <complex.h> را به برنامه افزود.

۹۲۸) تابع double fabs(double x) برای محاسبه مقدار قدر مطلق یک عدد اعشاری به کار می رود. برای استفاده ازاین تابع باید هدرفایل <math.h> را به برنامه افزود.

۹۲۹) تابع long double fabsl(long double x) برای محاسبه قدر مطلق یک عدد اعشاری بزرگ به کار می رود. برای استفاده ازاین تابع باید هدرفایل <math.h> را به برنامه افزود.

۹۳۰) تابع long int labs(long int x) برای محاسبه قدر مطلق یک عدد صحیح بزرگ به کار می رود. برای استفاده ازاین تابع باید هدرفایل <math.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۹۳۱) تابع int brk(void *addr) برای تغییر دادن آدرس فضایی که در برنامه اخذ شده و آدرس آن در data-segment می باشد به کار می رود. برای استفاده ازاین تابع باید هدرفایل <alloc.h> را به برنامه افزود.

۹۳۲) تابع void *sbrk(int incr) برای تغییر دادن آدرس فضایی که در برنامه اخذ شده و آدرس آن در data-segment می باشد به کار می رود. برای استفاده ازاین تابع باید هدرفایل <alloc.h> را به برنامه افزود.

۹۳۳) تابع unsigned coreleft(void) مقدار فضای استفاده نشده از حافظه را بر میگرداند. برای استفاده ازاین تابع باید هدرفایل <alloc.h> را به برنامه افزود.

۹۳۴) تابع unsigned long coreleft(void) مقدار فضای استفاده نشده را بر می گرداند. برای استفاده ازاین تابع باید هدرفایل <alloc.h> را به برنامه افزود.

۹۳۵) تابع unsigned long far coreleft(void) برای بدست آوردن مقدار فضای استفاده نشده در آدرس far می باشد. برای استفاده ازاین تابع باید هدرفایل <alloc.h> را به برنامه افزود.

۹۳۶) تابع void far *farcalloc(unsigned long units , unsigned long unitsz) برای گرفتن حافظه از آدرس far می باشد. برای استفاده ازاین تابع باید هدرفایل <alloc.h> را به برنامه افزود.

۹۳۷) تابع void free(void *block) برای آزاد کردن بلوکی از حافظه که قبلا از حافظه دریافت شده است به کار می رود. برای استفاده ازاین تابع باید هدرفایل <alloc.h> را به برنامه افزود.

۹۳۸) تابع int heapcheck(void) برای چک کردن پشته به کار می رود. برای استفاده ازاین تابع باید هدرفایل <alloc.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۹۳۹) تابع int farheapcheck(void) برای چک کردن پشته در آدرس far استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <alloc.h> را به برنامه افزود.

۹۴۰) تابع int heapcheckfree(unsigned int fillvalue) برای چک کردن بلوک حافظه آزاد در پشته استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <alloc.h> را به برنامه افزود.

۹۴۱) تابع int farheapcheckfree(unsigned int fillvalue) برای چک کردن بلوک حافظه آزاد در پشته استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <alloc.h> را به برنامه افزود.

۹۴۲) تابع int farheapchecknode(void *node) برای چک کردن یک نود در پشته با آدرس far استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <alloc.h> را به برنامه افزود.

۹۴۳) تابع int heapchecknode(void *node) برای چک کردن نود در پشته استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <alloc.h> را به برنامه افزود.

۹۴۴) تابع int heapfillfree(unsigned int fillvalue) برای خالی کردن حافظه پشته استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <alloc.h> را به برنامه افزود.

۹۴۵) تابع int farheapfillfree(unsigned int fillvalue) برای خالی کردن حافظه پشته با آدرس far استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <alloc.h> را به برنامه افزود.

۹۴۶) تابع int heapwalk(struct heapinfo *hi) برای قدم زدن در حافظه پشته گره به گره استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <alloc.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۹۴۷) تابع int farheapwalk(struct farheap info *hi) برای قدم زدن در پشته در آدرس far گره به گره استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <alloc.h> را به برنامه افزود.

۹۴۸) تابع void far *farmalloc(unsigned long nbytes) برای دریافت حافظه پویا از سیستم در آدرس far به مقدار nbytes استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <alloc.h> را به برنامه افزود.

۹۴۹) تابع void *realloc(void *block , size_t size) برای گرفتن دوباره حافظه پویا از سیستم استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <alloc.h> را به برنامه افزود.

۹۵۰) تابع void far *farrealloc(void far *oldblock , unsigned long nbytes) برای گرفتن حافظه پویا از سیستم در آدرس far استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <alloc.h> را به برنامه افزود.

۹۵۱) تابع void assert(int test) برای تست کردن یک شرط استفاده می شود که در صورت نیاز برنامه را abort خواهد کرد. برای استفاده ازاین تابع باید هدرفایل <assert.h> را به برنامه افزود.

۹۵۲) تابع int biosdisk(int cmd , int drive , int head , int track , int sector , int nsects , void *buffer) برای استفاده از عملکرد مستقیم دیسک در BIOS استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <bios.h> را به برنامه افزود.

۹۵۳) تابع unsigned _bios_disk(unsigned cmd , struct diskinfo_t *dinfo) برای استفاده از عملکرد مستقیم دیسک در BIOS استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <bios.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۹۵۴) تابع int biosequipment(void) برای چک کردن تجهیزات سیستم به کار می رود. برای استفاده ازاین تابع باید هدرفایل <bios.h> را به برنامه افزود.

۹۵۵) تابع unsigned _biod_equiplist(void) برای چک کردن تجهیزا سیستم به کار می رود. برای استفاده ازاین تابع باید هدرفایل <bios.h> را به برنامه افزود.

۹۵۶) تابع int bioskey(int cmd) برای استفاده از ویژگی های صفحه کلید که در bios تعریف شده اند به کار می رود. برای استفاده ازاین تابع باید هدرفایل <bios.h> را به برنامه افزود.

۹۵۷) تابع unsigned _bios_keybrd(unsigned cmd) برای استفاده از ویژگی های صفحه کلید که در bios تعریف شده اند به کار می رود. برای استفاده ازاین تابع باید هدرفایل <bios.h> را به برنامه افزود.

۹۵۸) تابع int biosmemsize(void) برای بدست آوردن سایز ram استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <bios.h> را به برنامه افزود.

۹۵۹) تابع unsigned _bios_memsize(void) برای بدست آوردن سایز ram استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <bios.h> را به برنامه افزود.

۹۶۰) تابع int biosprint(int xmd , int abyte , int port) برای نمایش حالات گوناگون پرینتر با استفاده از توابع BIOS می باشد. برای استفاده ازاین تابع باید هدرفایل <bios.h> را به برنامه افزود.

۹۶۱) تابع unsigned _bios_printer(int cmd , int port , int abyte) برای نمایش حالات گوناگون پرینتر با استفاده از توابع BIOS می باشد. برای استفاده ازاین تابع باید هدرفایل <bios.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۹۶۲) تابع int bioscom(int cmd , char abyte , int port) برای کار کردن با پورت RS-232 به کار می رود. برای استفاده ازاین تابع باید هدرفایل <bios.h> را به برنامه افزود.

۹۶۳) تابع unsigned _bios_serialcom(int cmd , int port , char abyte) برای کار کردن با پورت RS-232 به کار می رود. برای استفاده ازاین تابع باید هدرفایل <bios.h> را به برنامه افزود.

۹۶۴) تابع long biostime(int cmd , long newtime) برای دریافت و یا تنظیم زمان سیستم استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <bios.h> را به برنامه افزود.

۹۶۵) تابع unsigned _bios_timeofday(int cmd , long *timep) برای دریافت و یا تنظیم زمان سیستم استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <bios.h> را به برنامه افزود.

۹۶۶) تابع int cprintf(const char *format [,argument,...]) برای نمایش اطلاعات در خروجی صفحه نمایش می باشد. برای استفاده ازاین تابع باید هدرفایل <conio.h> را به برنامه افزود.

۹۶۷) تابع int fprintf(FILE *stream , const char *fornat [,argument,...]) برای نوشتن اطلاعات بر روی فایل استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <conio.h> را به برنامه افزود.

۹۶۸) تابع int sprintf(char *buffer , const char *format [,argument,...]) برای ذخیره اطلاعات در یک بافر استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <conio.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۹۶۹) تابع int vfprintf(FILE *stream , const char *format , va_list arglist) برای ذخیره اطلاعات در فایل با استفاده از arglist می باشد. برای استفاده ازاین تابع باید هدرفایل <conio.h> را به برنامه افزود.

۹۷۰) تابع int vprintf(const char *format , va_list arglist) برای خروجی اطلاعات بر روی stdin با استفاده از arglist میباشد. برای استفاده ازاین تابع باید هدرفایل <conio.h> را به برنامه افزود.

۹۷۱) تابع int vsprintf(char *buffer , const char *format , va_list arglist) برای ذخیره اطلاعات بر روی بافر با استفاده از arglist میباشد. برای استفاده ازاین تابع باید هدرفایل <conio.h> را به برنامه افزود.

۹۷۲) تابع int cscanf(char *format [,address , ...]) برای خواندن اطلاعات از ورودی صفحه کلید می باشد. برای استفاده ازاین تابع باید هدرفایل <conio.h> را به برنامه افزود.

۹۷۳) تابع int fscanf(FILE *stream ,const char *format [,address , ...]) برای خواندن اطلاعات از ورودی فایل می باشد. برای استفاده ازاین تابع باید هدرفایل <stdio.h> را به برنامه افزود.

۹۷۴) تابع int sscanf(const char *buffer [,address , ...]) برای خواندن اطلاعات از ورودی بافر می باشد. برای استفاده ازاین تابع باید هدرفایل <stdio.h> را به برنامه افزود.

۹۷۵) تابع int vfscanf(FILE *stream ,const char *format , va_list arglist) برای خواندن اطلاعات از ورودی فایل با استفاده از arglist می باشد. برای استفلده ازاین تابع باید هدرفایل <stdio.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۹۷۶) تابع int vscanf(const char *format , va_list arglist) برای خواندن اطلاعات از ورودی صفحه کلید با استفاده از arglist می باشد. برای استفاده ازاین تابع باید هدرفایل <stdio.h> را به برنامه افزود.

۹۷۷) تابع int vsscanf(const char *buffer , const char *format , va_list arglist) برای خواندن اطلاعات از ورودی بافر با استفاده از arglist می باشد. برای استفاده ازاین تابع باید هدرفایل <stdio.h> را به برنامه افزود.

۹۷۸) تابع void _setcursortype(int cur_t) برای تنظیم کردن خاصیت کرسر استفاده می شود. برای استفاده ازاین تابع باید هدرفایل <conio.h> را به برنامه افزود.

۹۷۹) ثابت _NOCURSOR برای خاموش کردن حالت کرسر استفاده می شود.

۹۸۰) ثابت _SOLIDCURSOR برای روشن کردن حالت کرسر استفاده می شود.

۹۸۱) ثابت _NORMALCURSOR برای نرمال کردن حالت کرسر استفاده می شود.

۹۸۲) تابع int islower(int c) برای چک کردن اینکه کاراکتر c کوچک باشد. برای استفاده ازاین تابع باید هدرفایل <ctype.h> را به برنامه افزود.

۹۸۳) تابع int isalpha(int c) برای چک کردن اینکه کاراکتر c الفبا باشد. برای استفاده ازاین تابع باید هدرفایل <ctype.h> را به برنامه افزود.

۹۸۴) تابع int isascii(int c) چک می کند که c مقدار بین ۰ تا ۱۲۷ را دارا باشد. برای استفاده ازاین تابع باید هدرفایل <ctype.h> را به برنامه افزود.

۹۸۵) تابع int isspace(int c) چک می کند که c کاراکتر space باشد. برای استفاده ازاین تابع باید هدرفایل <ctype.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۹۸۶) تابع int isdigit(int c) چک می کند که c عددی مابین ۰ تا ۹ باشد. برای استفاده ازاین تابع باید هدرفایل <ctype.h> را به برنامه افزود.

۹۸۷) تابع int isupper(int c) چک می کند که کاراکتر c حرف بزرگ باشد. برای استفاده ازاین تابع باید هدرفایل <ctype.h> را به برنامه افزود.

۹۸۸) تابع int isxdigit(int c) چک می کند که کاراکتر c عدد مبنای ۱۶ باشد. برای استفاده ازاین تابع باید هدرفایل <ctype.h> را به برنامه افزود.

۹۸۹) ثابت _IS_SP بیانگر کاراکتر فضای خالی می باشد. برای استفاده از این ثابت باید هدر فایل <ctype.h> را به برنامه افزود.

۹۹۰) ثابت _IS_DIG بیانگر کاراکتر عدد از ۰ تا ۹ می باشد. برای استفاده از این ثابت باید هدر فایل <ctype.h> را به برنامه افزود.

۹۹۱) ثابت _IS_UPP بیانگر کاراکتر حرف بزرگ می باشد. برای استفاده از این ثابت باید هدر فایل <ctype.h> را به برنامه افزود.

۹۹۲) ثابت _IS_LOW بیانگر کاراکتر حرف کوچک می باشد. برای استفاده از این ثابت باید هدر فایل <ctype.h> را به برنامه افزود.

۹۹۳) ثابت _IS_HEX بیانگر کاراکتر عدد مبنای ۱۶ می باشد. برای استفاده از این ثابت باید هدر فایل <ctype.h> را به برنامه افزود.

۹۹۴) ثابت _IS_CTL بیانگر کاراکتر فضای خالی می باشد. برای استفاده از این ثابت باید هدر فایل <ctype.h> را به برنامه افزود.

۹۹۵) ثابت _IS_PUN بیانگر کاراکتر فضای خالی می باشد. برای استفاده از این ثابت باید هدر فایل <ctype.h> را به برنامه افزود.

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610

۹۹۶) برای تنظیم کردن رجیستر DS در برنامه باید از دستور loadds_ استفاده شود.

۹۹۷) تابع unsigned long _lrotl(unsigned long val , int count) برای چرخش بیت های پارامتر اول تابع به تعداد پارامتر دوم به سمت چپ می باشد. برای استفاده از این تابع باید هدر فایا <stdlib.h> را به برنامه افزود.

۹۹۸) تابع unsigned long _lrotr(unsigned long val , int count) برای پرخش بیت های پارامتر اول تابع به تعداد پارامتر دوم به سمت راست می باشد. برای استفاده از این تابع باید هدر فایا <stdlib.h> را به برنامه افزود.

۹۹۹) تابع unsigned _rtol(unsigned val , int count) برای پرخش بیت های پارامتر اول به تعداد پارامتر دوم به سمت چپ می باشد. برای استفاده از این تابع باید هدر فایا <stdlib.h> را به برنامه افزود.

۱۰۰۰) تابع unsigned _rtor(unsigned val , int count) برای پرخش بیت های پارامتر اول به تعداد پارامتر دوم به سمت راست می باشد. برای استفاده از این تابع باید هدر فایا <stdlib.h> را به برنامه افزود.

۱۰۰۱) برنامه نویسی هنر است ، هنر استفاده از ابزار محدود در خلق موجودیتی نا محدود.

"استاد پویا لعل بخش"

یا حق

تالیف صابر عزیزپوریان
saber_azizpoorian@yahoo.com 09169742610