جزوه برنامه نویسی زبان سی

ویژه دانش آموزان و دانشجویان و علاقه مندان

گردآورنده : یعقوب عندلیب

رشته : مهندسی فناوری سیستم های سخت افزاری رایانه

پائیز ۹۲

فهرست

فهرست

فهرست

فهرست

فهرست

مقدّمه

جزوه حاضر درباره برنامه نویسی زبان سی و نوشتن الگوریتم ها و همچنین برنامه های ساده و مهّمی که در مدارس و مراکز دانشگاهی بیشتر مورد استفاده قرار می گیرند، اطلاعات کاملی ارائه داده است. همه مثال ها و برنامه ها بطور کامل شرح داده شده اند. سعی بر این است مطالب عاری از هرگونه ابهامات بوده و برنامه ها بطور کاملاً ساده و مفید بیان شوند تا مخاطبان در کمترین زمان ممکن، مطالب را بطور کامل یاد گرفته و مثال های مشابه را به سادگی حل نمایند.

ابتدا درباره نوشتن الگوریتم ها که اساسی ترین مرحله برنامه نویسی می باشند مثال هایی نوشته شده و سپس در رابطه با تعاریف بکار رفته در برنامه نویسی و بررسی تفاوت های آنها سخن گفته و نهایتاً به نوشتن برنامه ها در زبان سی پرداخته و دستورات مهم زبان برنامه نویسی سی؛ نظیر دستورات ورودی و خروجی، دستورات شرطی، دستورات حلقه و تکرار و آرایه های یک بعدی و دو بعدی و... را مورد بررسی قرار داده ایم.

در طول نگارش از منابعی همچون؛ ۵ فصل اول کتاب برنامه نویسی به زبان سی نوشته جعفرنژاد قمی و کتاب برنامه نویسی زبان سی و سی پلاس پلاس نوشته دایتل و دایتل و جزوات دانشگاهی اساتید بزرگوارم استفاده نموده ام.

با توجه به اینکه زبان برنامه نویسی سی به بزرگ و کوچک نوشتن حروف حساس است لذا در استفاده از برنامه ها به بزرگ و کوچک بودن حروف توجه نموده و سایر نکات برنامه نویسی را رعایت نمائید.

از آنجایی که در طول نگارش از حروف فارسی و انگلیسی و کاراکترها و ... استفاده شده لذا ممکن است تغییراتی بوجود آمده باشد. لذا امید است مخاطبان ارجمند در صورت وجود موارد مذکور را از طریق وبلاگ sarabrobo.blogfa.com و یا شماره تلفن۰۹۳۵۶۱۷۹۲۶۴ به ما اطلاع دهند.

آشنایی با کامپیوتر

کامپیوتر: وسیله ای است همانند سایر سیستم ها که از اجزای مختلف و مرتبط با هم تشکیل شده است که این اجزا هدف خاصی را دنبال می کنند. از اجزای یک سیستم می توان به واحد کنترل و پردازش و حافظه ها و ثبات ها و دستگاه های جانبی و دستگاه های ورودی و خروجی و ... اشاره نمود که هر کدام وظیفه خاص خود را دارند. اما برای ایجاد ارتباط بین این اجزا و همچنین نظم بخشیدن به نحوه عملکرد آنها به ابزار هایی نیازمند است که بتوان توسط آن برای انجام عملیات مختلف و مورد انتظار از آنها استفاده نموده و به سهولت به اهداف خویش دست یافت. به همین جهت در طراحی و ساخت کامپیوترها انعطاف پذیری از مهمترین موارد به شمار می رود. امروزه همه کامپیوترها از انعطاف پذیری بالایی برخوردار هستند.

حال که ارتباط بین سخت افزار های مورد نظر را ایجاد کردیم به ابزار هایی نیاز است که بتوان توسط آنها به اهداف مورد نظر دست یافت. قابل ذکر است که هر کامپیوتر عملاً یک دستگاه بی شعوری است که توسط برنامه نویسی آنرا با شعور می کنیم.

مزیّت کامپیوترها: مهمترین مزیت کامپیوترها سرعت و دقت بیشتر حل مسائل نسبت به انسان و همچنین قابلیت ذخیره سازی وسیع و امنیت اطلاعات می باشد.

هدف از برنامه نویسی: هدف از برنامه نویسی این است که بتوان مسائل مختلف را توسط کامپیوتر حل کرده و در زمان و هزینه های انجام آن صرفه جویی نمود.

روش حل مسئله :

۱-تجزیه و تحلیل مسئله(شتافت)

۲-یافتن راه حل مناسب برای مسئله(الگوریتم)

۳-کد نویسی به زبان های برنامه نویسی(مثل سی)

۴- اجرا و آزمایش برنامه برای داده های مختلف(نسخه آزمایشی)

۵- پشتیبانی و به روز رسانی برنامه

کامپیوترها به زبان اسمبلی (زبان ماشین) کار می کنند. لذا برای ایجاد ارتباط مابین انسان و کامپیوترها به زبان مشترکی نیز است که هم برای انسان و هم برای کامپیوترها قابل فهم باشد. زبان های برنامه نویسی به ما این امکان را می دهند که بتوانیم بواسطه آنها با کامپیوترها ارتباط برقرار کنیم. عمده ترین تفاوت زبان های برنامه نویسی در این است که دستورات بکار رفته و کاربرد هر یک از آنها برای انجام کارهای خاصی مانند برنامه نویسی صفحات وب و یا تحت ویندوز و ... می باشد. پس برای شروع برنامه نویسی ابتدا برای مسئله مورد نظر ترتیب بالا را رعایت نموده و در مرحله سوم با استفاده از زبان های برنامه نویسی و دستورات بکار رفته در آنها برنامه نویسی را شروع می کنیم.

الگوریتم: به چگونگی بیان راه و حل یک مسئله بصورت قدم به قدم و با جزئیات کافی که نقطه شروع و پایان آن مشخص باشد الگوریتم گویند.

مثال: الگوریتمی بنویسید که شعاع دایره ای را دریافت کرده و محیط و مساحت آنرا محاسبه کرده و چاپ کند؟

۱)شروع

۲) شعاع را در R بگیر

۳)S= 3.14*R*R

۴) M=2*R*3.14

۵) مساحت و محیط را چاپ کن

۶) پایان

مثال: الگوریتمی بنویسید که مختصات دو نقطه را در صفحه دریافت کرده و فاصله بین آن دورا محاسبه نموده و چاپ کند؟

$$d = \sqrt{(x1-x2)^2 + (y1-y2)^2}$$

۱)شروع

۲) نقاط x1,y1,x2,y2 را دریافت کن

۳) D=sqrt(pow(x1-x2,2) + pow(y1-y1,2)

۴) فاصله را چاپ کن

۵) پایان

مثال: الگوریتمی بنویسید که وزن جسمی را برحسب کیلوگرم دریافت کرده و به پوند تبدیل نموده و چاپ کند؟ (راهنمائی: هر پوند برابر۴۵۲ گرم می باشد.)

۱)شروع

۲) وزن را برحسب کیلوگرم در A بگیر

۳) G= A*1000

۴) p= G/452

۵) وزن بر حسب پوند را چاپ کن

۶) پایان

فلوچارت: فلوچارت نیز همانند الگوریتم مراحل انجام کاری بصورت قدم به قدم و با جزئیات کافی را بیان می کند که شروع و پایان کار مشخص می باشد. با این تفاوت که برای ترسیم فلوچارت از اشکال استاندارد استفاده می شود.

آشنایی با زبان سی

دستورات ورودی و خروجی: در زبان سی برای گرفتن اعداد و... از کاربر از دستورات ورودی و برای نمایش نتیجه عملیات از دستورات خروجی استفاده می شود. فرم استفاده از این دستورات در ادامه نمایش داده شده است.

Cin>>اسم متغیر ;

Cout<<عبارت خروجی ;

در زبان سی ثابت (لیترال ها) به شکل زیر بیان می شوند:

۱)عددی که از نوع صحیح و یا اعشاری می باشد.

۲)کاراکتر یا حروف که شامل تمامی حروف روی صفحه کلید می باشد.

۳)رشته ای که شامل حروف و کلمات می شوند.

انواع داده ها: زبان سی برای نگهداری اطلاعات در برنامه۴ نوع داده در اختیار ما قرار داده است و برنامه نویس باید متغیر های مورد نیاز را از نوع یکی از آنها در نظر بگیرد.

۱) کاراکتر: برای نگهداری و پردازش حروف و علائم مورد استفاده قرار می گیرد و یک بایت حافظه اشغال می کند.
۲) صحیح: برای اعداد طبیعی و صحیح بین۳۲۷۶۷ الی منفی ۳۲۷۶۸ را در بر می گیرد و دو بایت حافظه اشغال می کند.
۳) اعشاری: برای نگهداری و پردازش اعداد اعشاری (ممیز شناور) با دقت ۷ رقم اعشار می باشد که ۴ بایت حافظه اشغال می کند.
۴) دابل: برای نگهداری و پردازش اعداد اعشاری با دقت مضاعف می باشد که ۸ بایت حافظه اشغال می کند.

نکته: می توان پیشوند long را قبل از نوع داده صحیح قرار داده و اندازه حافظه آنرا دو برابر نمود.که در این حالت چهار بایت حافظه اشغال نموده و تقریبا مثبت و منفی دوملیارد را در خود ذخیره می کند.

تعریف متغیر: برای نگهداری و پردازش اطلاعات باید از متغیرها استفاده نمود که در ابتدای برنامه باید آنها را تعریف کنیم. شکل تعریف متغیر بصورت زیر است.

نوع داده نام متغیر ;

نوع داده نام متغیر اول و نام متغیر دوم و نام متغیرسوم ;

که نوع داده می تواند از انواع داده های بیان شده باشد و می توان از یک نام انگلیسی برای متغیرها استفاده نمود. که در ادامه قواعد نامگذاری متغیر ها را بررسی خواهیم نمود.

قواعد نامگذاری متغیرها: از کلمات رضرو شده زبان سی نباشد. حرف اول نام متغیر بایستی یکی از حروف انگلیسی و یا کاراکتر زیر خط (_A) باشد. حداکثر طول آن ۳۲ حرف باشد.سایر حروف آن ترکیبی از حروف انگلیسی و کاراکتر زیر خط و عدد باشد.

مقدار دهی اولیه متغیرها: اگر هنگام تعریف یک متغیر مقداری را برای آن تعیین کنیم اصطلاحاً می گوئیم متغیر ها را مقدار دهی اولیه نموده ایم. که برای این کار نام متغیر را مساوی عدد مورد نظر قرار می دهیم. و یا در طول برنامه می توان برای آنها مقدار مورد نظر را قرار داد.

نکات کلی: در ابتدای همه برنامه ها از شاوساین اینکلود برای الحاق نمودن کتابخانه به برنامه استفاده می شود. کتابخانه (فایل) محلی است که در آن تعدادی تابع و برنامه مخصوصی تعریف شده است.کتابخانه تابع main( ) جریان ورودی و خروجی محلی است که توابع ورودی و خروجی در آن تعریف شده اند.

تابع مین بدنه اصلی برنامه را تشکیل می دهد که باید کدهای برنامه را درون آن بنویسیم. و از کروشه باز برای شروع یک بخش(بلاک) و از کروشه بسته برای پایان آن بخش استفاده می کنیم.

در انتهای همه دستورات کاما سیمیکولن قرار می دهیم. و در یک خط می توان چندین دستور را نوشت. اما باید دقت نمود که زبان سی به بزرگی و کوچکی حروف حساس است.

برای استفاده از توابع دیگر آماده زبان سی باید قبل از تابع مین آنرا معرفی نمود و اغلب توابع ریاضی درون کتابخانه<math.h> تعریف شده اند.

دستورات شرطی: اگر اجرا شدن و یا اجرا نشدن دستوراتی وابسته به برقراری شرطی باشد از دستورات شرطی استفاده می کنیم. که دستورات شرطی بر دو نوع اند.

۱) شرطی ساده: اگر شرط برقرار باشد دستورات اجرا و در غیر این صورت اجرا نمی شوند.

if(شرط){

دستورات

}

۲) شرطی کامل: در شرطی کامل اگر شرط برقرار باشد دستورات برقراری شرط اجرا خواهند شد و در غیر این صورت دستورات عدم برقراری شرط انجام خواهند شد.

if(شرط){

دستورات برقراری شرط

else

دستورات عدم برقراری شرط

}

دستورات شرطی را با (شرط) if می نویسیم. و اگر تعداد دستورات بیش از یک دستور باشد از کروشه ها استفاده می کنیم. و در شرطی کامل از (شرطif دستورات برقراری و else دستورات عدم برقراری استفاده می کنیم.

دستورات شرطی: دستور سوئیچ نوع دیگری از دستورات شرطی است که در ادامه شرح داده شده است.

Switch(عبارت) {

Case مقدار اول ;

دستورات برابری عبارت با مقدار اول

Break;

Case مقدار دوم ;

دستورات برابری عبارت با مقدار دوم

Break;

.

.

.

Default;

دستورات عدم برابری عبارت با هیچ یک از مقادیر

}

عبارت می تواند یک متغیر باشد که در هر مرحله با مقادیر اول و دوم و... مقایسه می شود و با هر یک از آنها برابر باشد دستورات مربوط به آن را اجرا خواهد نمود. با استفاده از دستور Break از آن دستور خارج شده و سطر های بعدی را مرور می کند. قسمت آخرdefault را می توان از برنامه حذف نمود.

دستورات حلقه و تکرار: اگر بخواهیم تعدادی از دستورات به میزان مشخصی تکرار شوند از دستورات حلقه و تکرار استفاده می کنیم. که دستورات حلقه و تکرار بر سه نوع اند.

۱) شرط در ابتدای حلقه: در ابتدا شرط بررسی می شود اگر برقرار باشد دستورات داخل حلقه اجرا می شوند و اگر برقرار نباشد برنامه از حلقه خارج خواهد شد.
۲) شرط در وسط حلقه: در این صورت پس از اجرای دستوراتی برنامه به شرط حلقه می رسد و آنرا بررسی می کند اگر برقرار باشد دستورات پس از شرط را اجرا می کند و در غیر این صورت از حلقه خارج می شود.
۳) شرط در انتهای حلقه: در این حالت تمامی دستورات حداقل یکبار اجرا می شوند و در نهایت به شرط حلقه می رسد و آنرا بررسی می کند اگر برقرار باشد دوباره به ابتدای حلقه می رود و دستورات را اجرا می کند و اگر برقرار نباشد از حلقه خارج می شود. در حالی که در نوع اول یعنی شرط در ابتدای حلقه ممکن است دستورات آن حتی یکبار هم اجرا نشوند.

برای نوشتن دستورات حلقه و تکرار از دستور(; ;)for استفاده می کنیم. که در این دستور:

عبارت اول معمولا به عنوان مقدار دهی اولیه شمارنده حلقه مورد استفاده قرار می گیرد و فقط یکبار هنگام ورود به حلقه اجرا می شود.

عبارت دوم به عنوان شرط حلقه می باشد و تا زمانی که برقرار باشد تکرار خواهد شد و این عبارت قبل از هربار تکرار حلقه ارزیابی می شود.

عبارت سوم معمولا به عنوان گام حلقه بکار می رود و هر بار پس از اجرای دستورات حلقه ارزیابی می شود.

نکته: از هریک از عبارت های سه گانه حلقهfor می توان صرف نظر نمود اما درج سیمیکولن ها بین عبارت ها الزامی است.

دستور حلقه (شرط) while : در این دستور ابتدا از کلمه وایل سپس درون پارانتز شرط را می نویسیم و دستورات را درون کروشه ها قرار می دهیم.

نکته: از دستور حلقه فور اغلب در مواردی استفاده می شود که تعداد تکرار حلقه مشخص باشد. و از دستور حلقه وایل اغلب در مواردی استفاده می شود که تعداد تکرار مشخص نباشد. اما می توانند بجای یکدیگر نیز بکار گرفته شوند. اگر بجای شرط از عدد یک استفاده کنیم به این مفهوم است که شرط همیشه برقرار است

دستور do-while( ) نیز همانند وایل می باشد با این تفاوت که در آن ابتدا از کلمه do و کروشه دستورات حلقه و سپس (شرط) while استفاده می کنیم که در آخر برنامه شرط بررسی می شود. و دارای محدودیت است. مثلا نمی توان برنامه بزرگترین مقسوم علیه مشترک را با دو وایل نوشت زیرا اگر دو عدد ورودی بر یکدیگر بخش پذیر باشند در این صورت خطای تقسیم بر صفر اتفاق می افتد.

حلقه های تو در تو: در یک حلقه تکرار می توان دستورات مختلفی قرار داد که از جمله آن، دستور حلقه دیگری است که به حلقه شامل حلقه تکرار دیگر حلقه تکرار خارجی و به حلقه ای که در داخل حلقه تکرار خارجی قرار گرفته است حلقه تکرار داخلی می گویند. که در این صورت به ازای هربار تکرار حلقه خارجی، حلقه تکرار داخلی بطور کامل از اول تا آخر اجرا خواهد شد.

نکته: در زبان ها یک به معنی درست بودن و صفر به معنای عدم برقراری است لذا گاهاً در برنامه نویسی معمولاً درون حلقه وایل بجای شرط از عدد یک استفاده می شود که نشان دهنده برقراری شرط می باشد.

آرایه یک بعدی یا خطی: آرایه به متغیر های هم نامی گفته می شود که نوع آنها یکسان بوده و در حافظه کنار یکدیگر قرار می گیرند و با اندیس از هم جدا می شوند. و بصورت زیر در برنامه تعریف می شود. عدد داخل کروشه (تعداد عناصر) نیز توسط برنامه نویس بصورت اختیاری وارد می شود.

نوع داده نام آرایه [تعداد عناصر] ;

آرایه دو بعدی یا ماتریس: برای نگهداری و پردازش اطلاعاتی که دارای ماهیت جدولی یا ماتریسی هستند از آرایه دو بعدی استفاده می شود. که همچون آرایه خطی است با این تفاوت که از دو شمارنده یکی برای سطرها و دیگری برای ستون ها مورد استفاده قرار می گیرد.

نوع داده نام آرایه [تعداد سطرها] [تعداد ستونها] ;

تابع یا فانکشن: زبان های برنامه نویسی ساخت یافته این امکان را به کاربر می دهند که برنامه های بزرگ را به برنامه های کوچکتر تقسیم نموده و بین اعضای گروه تقسیم نمود و نهایتا با یک جمع بندی به اهداف مورد نظر دست یافت. مزیت تابع بر این است که می توان آنرا یکبار تعریف نموده و بارها از آن استفاده نمود. معمولا تعدادی از دستورات که وظیفه مشخصی را انجام می دهند و در طول برنامه چندین بار مورد استفاده قرار می گیرند را بصورت تابع تعریف می کنیم.

نوع داده خروجی نام تابع (پارامترهای تابع) {

دستورات تابع

Return مقدار خروجی;

}

نوع داده خروجی: یکی از انواع داده ها در زبان سی می باشد و در صورتی که تابع دارای خروجی نباشد آنرا از نوع void یا پوچ تعریف می کنیم.

پارامتر های تابع: یک تابع می تواند صفر یا چندین پارامتر داشته باشد. که همان متغیر ها می باشند.

اجرای یک تابع با فراخوانی آن شروع شده و با رسیدن به دستورreturn و یا کروشه بسته انتهای آن به پایان می رسد. توابع را قبل از فراخوانی تعریف می کنیم. و باید دقت نمود که تعریف تودر تو توابع مجاز نیست. و باید بطور جداگانه هرکدام را تعریف کنیم.

تابع main() در واقع یک تابعی است که با فراخوانی آن توسط سیستم عامل اجرای برنامه شروع می شود.

نکاتی در زبان سی:

جهت نمایش اطلاعات راهنمایی کاربر متن مورد نظر را درون: " می نویسیم.

برای نمایش نتیجه در خط یا سطر بعدی از cout<<endl<<I; استفاده می کنیم.

در زبان سی 1/2=0 اما 1.0/2=0.5

در زبان سی رادیکال x را به شکل Sqrt(x) نشان می دهند.

در زبان سی x به توان y را به شکل Pow(x,y) نشان می دهند.

نرم افزار مورد استفاده: با توجه به مطالب فوق با اطلاعات کلی برنامه نویسی و زبان سی آشنا شدیم. لذا در ادامه برای کامل شدن بحث اولین برنامه را شروع می کنیم. و در طول هر برنامه توضیحات لازم ارائه خواهند شد. اکثر برنامه ها در نرم افزار Borland c++5.02 تست شده اند لذا برای استفاده از آنها در سایر نرم افزار ها باید خطا های مورد نظر را اصلاح نمود. نرم افزار های متفاوتی برای هر کدام از زبان های برنامه نویسی وجود دارند که در این مجموعه برای برنامه نویسی زبان سی از نرم افزار فوق استفاده شده است. سادگی و سرعت عملکرد و محیط کاملاً ساده این نرم افزار باعث شده تا در امر آموزش بیشتر مورد استقبال قرار گیرد. همچنین این نرم افزار حجم کمتری را داراست که می توان به سادگی آنرا از اینترنت نیز دانلود نمود. محیط نرم افزار شبیه به صفحه نوت پد ویندوز بوده و برای ایجاد یک پروژه جدید می توان از گزینه file/ creat new project استفاده نموده و برای ذخیره نمودن کدهای نوشته شده از گزینه file/ save فایل ها را با پسوند پیش فرض cpp ذخیره نمود. و برای اجرا و یا امتحان نمودن نرم افزار از گزینه debug/ run استفاده می شود.

برای اینکه پس از دریافت عدد مورد نظر، صفحه برنامه قطع نشود، در انتهای همه برنامه ها و قبل از کروشه بسته از دستور زیر استفاده می کنیم. Getch( );

مثال ها و برنامه ها

مثال۱: برنامه ای بنویسید که شعاع دایره ای را گرفته و محیط و مساحت آنرا محاسبه و چاپ کند؟

```
#include<iostream.h>
Main() {
        Float r,a,p;
        Cout<< " enter a radius:";
        Cin>>r;
```

```
a=3.14*r*r;
p=2*r*3.14;
cout<< “area=”<<a<<”perimeter=”<<p;
}
```

توضیحات کلی: از شاوساین اینکلود برای الحاق نمودن کتابخانه به برنامه استفاده می شود. کتابخانه محلی است که در آن تعدادی تابع و برنامه مخصوص تعریف شده است.

کتابخانهiostream.h محلی است که در آن جریان ورودی و خروجی تعریف شده اند. و برنامه اصلی درون تابع main نوشته می شود. برای شروع یک بخش(بلاک) از کروشه باز و برای بستن همان بلاک از کروشه بسته استفاده می کنیم. در انتهای همه دستورات از علامت کاما سیمیکولن استفاده می شود. و در یک سطر می توان چندین دستور نوشت اما باید در انتهای دستورات از کاما سیمیکولن استفاده نمود. دقت کنید که در برخی موارد معمولاً پس از دستورات if ,for,... سیمیکولن قرار داده نمی شود به این علّت که آنها دستور نیستند و دستورات پس از آنها نوشته می شود.

راهنمائی: در همه برنامه ها ابتدا باید متغیرها را تعریف نموده و سپس با نمایش یک پیام اختیاری کاربران برنامه را راهنمائی نموده و سپس عدد ورودی را دریافت کرده و نهایتاً سایر دستورات را بکار ببریم. و در پایان کار مقادیر خروجی ها را با دستورات مربوطه نمایش دهیم.

مثال۲: برنامه ای بنویسید که طول جسمی را بر حسب سانتی متر دریافت کرده و به اینچ تبدیل کند؟

```
#include<iostream.h>
Main() {
        Float  n, m;
        Cout<< " enter a number:";
        Cin>>n;
        m= n / 2.54;
        cout<< "inch="<<m;
       }
```

راهنمائی: با توجه به اینکه هر اینچ برابر ۲/۵۴ (دو و پنجاه و چهار صدم) سانتی متر می باشد. و طول جسم می تواند عددی اعشاری و یا صحیح باشد آنرا از نوع اعشاری تعریف می کنیم. سپس عدد را از کاربر دریافت نموده و بر ۲/۵۴ تقسیم می کنیم حاصل تقسیم جواب را نشان می دهد.

مثال۳ : برنامه ای بنویسید که مختصات دو نقطه در صفحه را گرفته و فاصله بین آن دو را محاسبه و چاپ کند؟

$$d = \sqrt{(x1-x2)^2+(y1-y2)^2}$$

```
#include<iostream.h>
#include <math.h>
Main() {
        Float  x1,x2,y1,y2,d;
       Cout<< “x1 ,x2,y1,y2:”;
       Cin>>x1>>x2>>y1>>y2;
       D=sqrt(pow(x1-x2,  2)+pow(y1-y2, 2));
       cout<< “faseleh=”<<d;
       }
```

راهنمائی: با توجه به اینکه هر نقطه بر روی صفحه مختصات با دو متغیرx,y نشان داده می شود. پس برای فاصله دو نقطه طبق رابطه فوق عمل می کنیم. که در زبان سی رادیکال را با sqrt(x) و توان را با pow(x,y) نمایش می دهند. چون در طول برنامه از توابع ریاضی مثل رادیکال و توان استفاده کرده ایم باید آنها را قبل از استفاده به برنامه معرفی کنیم که این کار با دستورmath.h انجام می گیرد. برای نمایش اندیس در کامپیوتر آن را بعد از عدد می نویسیم. و چون مقادیر می توانند بصورت اعشاری و یا صحیح باشند از نوع اعشار انتخاب شده اند. به دستورات ورودی وخروجی و پرانتز ها نیز توجه شود. که استفاده از چنین دستورات در برنامه نویسی مجاز می باشند.

مثال۴: برنامه ای بنویسید که عدد صحیحی را گرفته و قدر مطلق آنرا محاسبه و چاپ کند؟

```
#include<iostream.h>
```

```
Main() {
        int r ;
Cout<< " enter a number:";
Cin>>r;
If(r >0) {
         r = r ;
   else
         r = - r ;|| r=(-1)*r;
        }
cout<< "gadr motlag="<<r ;
}
```

راهنمائی: خاصیت قدر مطلق این است که علامت اعداد منفی را به عدد مثبت تبدیل می کند. لذا در این برنامه برای جداکردن اعداد مثبت و منفی از یکدیگر عدد صفر را بعنوان یک شرط در نظر می گیریم. اعدادی که از صفر بزرگترند قدر مطلق آنها با خودشان برابر است. و اعدادی که از صفر کوچکتر باشند قدر مطلق شان برابر با منفی عدد برابر است. در برخی موارد برای نوشتن یک دستور راه های زیادی وجود دارند و یا برای استفاده از کلمه یا در فارسی از دو علامت|| استفاده می کنیم.

مثال۵: برنامه ای بنویسید که دو عدد اعشاری را گرفته و ماکسیمم آنها را محاسبه و چاپ کند؟

```
#include<iostream.h>
Main() {
        Float   a, b, max;
Cout<< " enter 2 number :";
Cin>>a>>b;
If(a>b) {
      Max=a;
```

```
Else
        Max=y;
}
cout<< “max=”<<max;
}
```

راهنمائی: چون اعداد مورد نظر اعشاری هستند لذا آنها را از نوع اعشاری انتخاب نموده و با فرض اینکه اولین عدد بیشتر از دومی است برنامه را می نویسیم اگر شرط برقرار بود عدد اول بیشتر است و چاپ می شود. در غیر این صورت عدد دوم بیشتر است و چاپ می شود. دقت کنید که در این برنامه اگر هر دو عدد با هم برابر باشند دومین عدد را به عنوان بزرگترین چاپ می کند.

مثال۶: برنامه ای بنویسید که طول اضلاع مثلثی را گرفته و بررسی کند که آیا مثلث قائم الزاویه است یا نه؟

```
#include<iostream.h>
Main() {
        Float a,b,c;
Cout<< “ enter a,b,c:”;
Cin>>a>>b>>c
If(a*a==b*b+c*c||b*b==a*a+c*c||c*c==a*a+b*b)
cout<< “right angled”;
else
cout<< “ NO right angled”;
}
```

راهنمائی: چون اضلاع می توانند از نوع صحیح و اعشاری باشند پس از نوع اعشاری تعریف می کنیم. در نمایش پیام و دریافت اعداد باید ترتیب را رعایت کنیم. حال شرط قائم الزاویه بودن مثلث این است که مجذور وتر با مجموع مجذور تک تک سایر اضلاع برابر باشد. که در این برنامه هر سه به عنوان یک شرط بررسی می شوند.

اگر یکی از آنها برقرار باشد چاپ می کنیم قائم الزاویه است و در غیر این صورت چاپ می کنیم قائم الزاویه نیست.

مثال۷: برنامه ای بنویسید که ضرایب معادله ی درجه دوم را دریافت نموده و ریشه های آنرا بررسی‌کند؟

$$ax^2 + bx + c = 0$$

```
#include<iostream.h>
#include<math.h>
Main() {
        Float a, b, c, x1, x2, d;
Cout<< " enter a ,b,c:";
Cin>>a>>b>>c;
d=((b*b) – (4*a*c));
if(d<0)
cout<< "rishe hagigi nadarad";
if(d==0){
          x1= - b / (2*a);
          cout<< "x1="<<x1;
          }
If(d>=0){
            x1= (- b +sqrt(d)) / (2*a);
      x2= (- b -sqrt(d)) / (2*a);
cout<<"x1="<<x1<<"x2="<<x2;
           }
 }
```

راهنمائی: برای حل یک معادله درجه دوم ابتدا دلتا را بدست آورده و سپس دلتا را با عدد صفر مقایسه می کنیم اگر کمتر از صفر باشد می نویسیم ریشه حقیقی ندارد. و در صورتی که دلتا برابر صفر باشد می نویسیم فقط یک ریشه حقیقی دارد. و در صورتی که از صفر بیشتر باشد دو ریشه حقیقی دارد که با فرمول های مربوطه آنها را بدست می آوریم.

مثال۸: برنامه ای بنویسید که شماره روز(هفته) را گرفته و نام روز را چاپ کند؟

```
#include<iostream.h>
Main() {
        Int d;
Cout<< " enter a day number :";
Cin>>d;
Switch(d) {
      Case1:  cout<< "shanbeh" ;
      Break;
      Case2:  cout<< " 1 shanbeh" ;
      Break;
      Case3:  cout<< "2 shanbeh" ;
      Break;
      Case4:  cout<< "3 shanbeh" ;
     Break;
     Case5:  cout<< "4 shanbeh" ;
     Break;
     Case6:  cout<< "5 shanbeh" ;
     Break;
    Case7:  cout<< "jomeh " ;
```

```
        Break;
        Defult :cout<<"na motabar";
      }
}
```

راهنمائی: با توجه به مسئله در این برنامه کاربر عددی را وارد می کند و بر اساس برنامه تعریف شده نام آن روز چاپ می شود. چون تعداد روز های یک هفته۷ عدد می باشد لذا اعداد بین یک تا هفت دارای جواب بوده و بقیه اعداد نا معتبر خواهد بود. از دستور سویچ استفاده می کنیم. طوری که در ابتدای برنامه متغیری را از نوع عدد صحیح تعریف نموده و در هر مرحله مقدار آن عدد با تک تک دستورات مقایسه می شود اگر با یکی از دستورات برنامه برابر شود، آنگاه آنرا اجرا نموده و از دستور خارج می شود. و برنامه دوباره کار خویش را از ابتدا دنبال می کند. دقت نمائید که این برنامه را می توان باif نیز نوشت چرا که مقدار مقابل هر یک از کیس ها می تواند از نوع عددی و یا کاراکتری باشد. پس در حالت کلی دستور سوؤیچ از خوانایی بیشتری برخوردار است و دستورif جامع تر است.

مثال۹: برنامه ای بنویسید که دو عدد صحیح و یک کاراکتر را به عنوان یکی از چهار عمل اصلی ریاضی دریافت کرده و مقدارآنرا محاسبه و چاپ کند؟(ماشین حساب ساده)

```
#include<iostream.h>
Main() {
        Int a,b;
        Char d;
  Cout<< " enter 2 number :";
Cin>>a>>b;
Cout<<"enter a operator:";
Cin>>d;
Switch(d) {
          Case '+':
          cout<<a+b;
```

```
            Break;
            Case '-':
            cout<<a-b;
            Break;
      Case '*':
            cout<<a*b;
            Break;
      Case '/ ':
If(b!=0)
            cout<<a/b;

           Else
           Cout<<" error:";
           Break;
          }
}
```

راهنمائی: برای محاسبه چهار عمل اصلی ریاضی بین دو عدد و یا بیشتر، به یک عملوند نیاز داریم که حاصل جمع، تفریق، ضرب و یا تقسیم بین آن دو را بدست آوریم. چون این عملوند ها از نوع کاراکتری بشمار می آیند، لذا متغیری را از نوع کاراکتری در نظر می گیریم که این عملگر ها را مشخص کند. با استفاده از دستور سوئیچ به سادگی این کار را انجام می دهیم. چون نتیجه تقسیم هر عدد بر صفر برابر بی نهایت است، لذا در مرحله تقسیم دو عدد شرط مخالف صفر بودن مخرج را بررسی می کنیم. اگر مخرج صفر بود پیام خطا ظاهر می شود و اگر مخرج کسر مخالف صفر باشد نتیجه تقسیم نمایش داده می شود.

مثال۱۰: برنامه ای بنویسید که یک عدد طبیعی را از کاربر گرفته و از یک تا آن عدد را چاپ کند؟

```
#include<iostream.h>
```

```
Main() {
        Int i, p;
Cout<< " enter  p:";
Cin>>p;
For(i=1;i<=p;i++)
cout<< i;     ||   cout<<endl<<i;
}
```

راهنمائی: اعداد طبیعی اعدادی اند که از مثبت یک تا مثبت بینهایت را شامل می شوند. در این برنامه ابتدا عددی را از کاربر دریافت کرده و سپس اعداد یک تا خود آن عدد را چاپ می کند. از یک شمارنده حلقه استفاده می کنیم تا به تعداد عدد وارد شده حلقه مورد نظر تکرار شود. در نمایش نتایج بدست آمده برای چاپ پشت سرهم و در یک سطر از دستور اول و برای چاپ هر یک در یک سطر جداگانه از دستور دوم استفاده می کنیم.

مثال۱۱: برنامه ای بنویسید که عدد طبیعیN را گرفته و n! را محاسبه کرده و چاپ کند؟

```
#include<iostream.h>
Main() {
        Int i,n;
Long int f=1;
Cout<< " enter n:";
Cin>>n;
for(i=1;i<=n;i=i+1)
f=f*i;
cout<< f;
}
```

راهنمایی: در محاسبه فاکتوریل تمامی اعداد قبلی را بطور متوالی به یکدیگر ضرب می کنیم. پس می توان گفت یک شمارنده که هربار یک واحد به آن افزوده می شود و در هر مرحله این شمارنده در مقادیر قبلی

ضرب می شود. تا نتیجه نهایی بدست آید. در این مسئله از یک حلقه برای شمارنده استفاده شده است و چون جواب بدست آمده بسیار بزرگتر می شود از نوع عدد صحیح بزرگتر استفاده می شود.

مثال۱۲: برنامه ای بنویسیدکه دوعدد طبیعی را گرفته و بزرگترین مقسوم علیه مشترک آن دو رامحاسبه و چاپ کند؟

```
#include<iostream.h>
Main() {
        Int a,b,r;
Cout<< " enter a,b:";
Cin>>a>>b;
While (a%b !=0) {
r =a%b;
a=b;
b=r;
cout<< b;
}
```

راهنمایی: برای بدست آوردن بزرگترین مقسوم علیه مشترک دو عدد فرض می کنیم اولی بزرگتر از دومی است. و عدد کوچکتر را در متغیر دیگری قرار می دهیم. سپس باقیمانده عدد بزرگتر بر متغیر مورد نظر را مقایسه می کنیم اگر صفر بود یعنی هر دو بر عدد دوم(کوچکتر) بخش پذیر هستند و عدد کوچکتر، بزرگترین مقسوم علیه مشترک است. اگر بر آن بخش پذیر نباشد یک واحد از مقدار آنرا کم می کنیم و برای عدد بدست آمده محاسبات را تکرار می کنیم. دقت کنید که در این روش تعداد دفعات تکرار مشخص نیست.

روش دوم: روش نردبانی (پلکانی) در این روش هم چون روش قبلی کوچکترین عدد بین آن دو را پیدا کرده و سپس باقیمانده عدد بزرگتر بر کوچکتر را بدست آورده و با صفر مقایسه می کنیم. اگر برقرار بود عدد کوچکتر جواب مسئله است. در غیر این صورت باقیمانده عدد بزرگتر بر کوچکتر را بدست آورده و عدد کوچکتر را با

بزرگتر جایگزین می کنیم تا در مرجله بعدی مقادیر در شرط ها کمتر شوند. یعنی تقسیم عدد کوچکتر بر باقیمانده را مقایسه کنیم و نهایتا مقدار باقیمانده را بجای عدد کوچکتر قرار می دهیم تا حلقه تکرار شود.

مثال۱۳: برنامه ای بنویسید که عدد طبیعی n را گرفته و مقلوب آنرا به دست آورد؟

```
#include<iostream.h>
Main() {
Int  n, s=0, r ;
Cout<< " enter a number:";
Cin>>n;
While( n>0) {
      r =n% 10;
     n=n/10;
     s=s*10+r;
}
cout<< "maghloob="<<s ;
}
```

راهنمایی: تعریف مقلوب؛ عددی است که مکان رقم های آن برعکس شود.

برای بدست آوردن مقلوب هر عددی آنرا پشت سر هم به عدد ۱۰ تقسیم می کنیم و سپس باقیمانده تقسیم عدد را بدست آورده و باقیمانده را در مجموعه جدید قرار داده و در هر مرحله جمله قبلی را در عدد ۱۰ ضرب و با باقیمانده تقسیم جمع می کنیم. تا زمانی که عدد از صفر بزرگتر است تقسیم را ادامه می دهیم.

مثال۱۴: برنامه ای بنویسید که یک عدد طبیعی را گرفته و حاصلضرب ارقام آنرا بدست آورد؟(با استفاده از دستور دو وایل)

```
#include<iostream.h>
Main() {
```

```
        Int n,r,s=1;
Cout<< " enter a number:";
Cin>>n;
do {
    r=n%10;
    n=n/10;
    s=s*r;
   } while(n>0   ||   n!=0);
cout<< s;
}
```

راهنمائی: جهت بدست آوردن ارقام یک عدد آن را پشت سر هم به۱ تقسیم می کنیم.(چون عدد در مبنای ده دهی است) یعنی ابتدا آن عدد را به ۱۰ تقسیم و سپس باقیمانده آن عدد بر ۱۰ و تا آخر این کار را ادامه می دهیم. برای بدست آوردن باقیمانده ابتدا آنرا بدست آورده و سپس تقسیم بر ده را ادامه می دهیم. چون عدد یک در ضرب بی تاثیر است پس مقدار اولیه متغیر مورد نظر را یک قرار می دهیم و در هر مرحله باقیمانده را به مقدار قبلی متغیر مورد نظر ضرب می کنیم. با توجه به دستور مورد استفاده در این مرحله شرط حلقه در انتهای آن بررسی می شود که آیا عدد وارد شده بزرگتر از صفر و یا مخالف صفر است یا نه؟ تا عمل تقسیم را ادامه دهد یا با سطر بعدی برنامه برود.

مثال۱۵: برنامه ای بنویسید که دو عدد طبیعی را دریافت کرده و جدول ضربی به a سطر و b ستون چاپ کند؟

```
#include<iostream.h>
Main() {
        Int   a,b,I,j ;
Cout<< " enter a, b:";
Cin>>a>>b;
For(i=1;i<=a;++i) {
```

```
                For(j=1;j<=b;++j)
                Cout<< i * j ;
                cout<< endL;
                }
}
```

راهنمائی: چون هر جدول ضرب از یک سطر و یک ستون برای نمایش هر عدد تشکیل شده است لذا برای نمایش تعداد سطرها از متغیر i و برای نمایش ستون ها از متغیر j استفاده می کنیم. چون در مرحله اول اعداد ستون های سطر اول درج شده و سپس به سطر بعدی می رود. حلقه سطرها را حلقه خارجی و حلقه ستون ها را حلقه داخلی می نامیم. و در هر مرحله مقدار سطر را در ستون مورد نظر ضرب نموده و نتیجه را چاپ می کند. هنگامی که به انتهای دستورات رسید به سطر بعدی رفته و حلقه ها را ادامه می دهد.

مثال۱۶: برنامه ای بنویسید که چند عدد طبیعی را گرفته و مقلوب هر یک را محاسبه و چاپ کند؟

```
#include<iostream.h>
Main() {
        Int n,I,x,s=0,r;
Cout<< " enter 2 number:";
Cin>>n
For(i=1;i<=n;i++) {
                Cin>>x;
For( ;x>0;) {
R=x%10;
X=x/10;
S=s*10+r;
}
}
```

```
cout<< s;
}
```

راهنمائی: برای بدست آوردن مقلوب یک عدد آنرا بطور متوالی بر عدد ۱۰ تقسیم کرده و در هر مرحله باقیمانده ها را در ۱۰ ضرب نموده و با مقدار باقیمانده در مرحله بعدی جمع می کنیم. جهت دریافت تعداد اعداد ابتدا عددی را وارد می کنیم که تعداد اعداد مورد نظر را نشان می دهد. سپس با استفاده از یک حلقه خارجی تعداد اعداد را بررسی می کنیم. بعد از آن اولین عدد را از کاربر دریافت کرده و مقلوب آنرا محاسبه و چاپ می کنیم. سپس به ابتدای حلقه رفته و عدد دوم را دریافت و محاسبه می کنیم و این کار را آخرین عدد مورد نظر ادامه می دهیم.

مثال۱۷: برنامه ای بنویسید که عددی را گرفته و مجموع ارقام آنرا محاسبه و چاپ کند؟

```
#include<iostream.h>
Main() {
        Int n,r,s=0;
Cout<< " enter a number:";
Cin>>n;
While(n!=0) {
                R=n%10;
                N=n/10;
                S=s+r ;
                }
cout<< s;
}
```

راهنمائی: برای بدست آوردن ارقام عدد مورد نظر، بطور متوالی آن را بر ۱۰ تقسیم می کنیم. باقیمانده تقسیم همان رقم های آن عدد می باشد. چون عدد صفر در عمل جمع بی تاثیر است پس مقدار اولیه متغیر مورد نظر را صفر کرده و در هر مرحله مقدار باقیمانده را به آن اضافه می کنیم.

مثال۱۸: برنامه ای بنویسید که عددی را گرفته و ماکسیمم رقم های آنرا محاسبه و چاپ کند؟

```
#include<iostream.h>
Main() {
        Int n, r, max=0 ;
Cout<< " enter a number:";
Cin>>n;
While(n>0) {
                R=n%10;
                N=n/10;
                If(r>max)
                Max=r;
               }
cout<< "maximuom="<<max;
}
```

راهنمایی: چون از عمل مقایسه استفاده می کنیم، لذا باید برای متغیر ها مقدار تعریف کنیم که مقدار صفر در مقایسه اولیه بی تاثیر بحساب می آید. برای محاسبه ماکسیمم ابتدا رقم های آنرا با استفاده از تقسیم بر ۱۰ و بدست آوردن باقیمانده جدا کرده و در هر مرحله با مقدار رقم قبلی مقایسه می کنیم درصورتی که بزرگتر از رقم های قبلی باشد بزرگترین رقم است و در غیر این صورت رقم قبلی بزرگتر از بقیه است و چاپ می شود.

مثال۱۹: برنامه ای بنویسید که دو عدد طبیعی a,b را گرفته و a^b را با استفاده از ضرب های متوالی بدست آورده و چاپ کند؟

```
#include<iostream.h>
```

```
Main() {
Int  a, b, i, p=1;
Cout<< " enter a, b :";
Cin>>a>>b;
For(i=1;i<=b;i++)
p=p*a;
        cout<< p;
}
```

مثال۲۰: برنامه ای بنویسید که دما را بر حسب سانتی گراد دریافت نموده و به درجه فارنهایت تبدبل نموده و چاپ کند؟

```
#include<iostream.h>
Main() {
        Float F,C;
Cout<< " damaye celsiuse ra vared konid :";
Cin>>C;
F=C*1.8+32;
cout<< "damaye fahrenheit ="<<F;
}
```

مثال۲۱: برنامه ای بنویسید که برای یک حساب سپرده گذاری موجودی ابتدای دوره و نرخ بهره سالیانه و تعداد سال را دریافت کرده و موجودی نهایی را محاسبه کرده و چاپ کند؟

```
#include<iostream.h>
Main() {
        Float f,i,n,p;
Cout<< " enter f=mojodi,i=darsad,n=sal:";
```

```
Cin>>f>>i>>n;
p=f*pow(1+i,n);
cout<< “mojodi nahaee=”<< p;
}
```

راهنمایی: برای بدست آوردن موجودی نهایی، ابتدا سرمایه اولیه که یک عدد مشخصی است و نرخ بهره را بر حسب درصد مشخص نموده و سپس مطابق فرمول زیر آنرا بدست می آوریم.

(بتوان سال (درصد +۱))*سرمایه اولیه = موجودی نهایی

مثال۲۲: برنامه ای بنویسید که مجموع اعداد فرد دو رقمی را بدست آورده و چاپ کند؟

```
#include<iostream.h>
Main() {
Int i=11,s=0;
If(i<=99) {
  s=s+i;
  i=i+2;
 }
cout<< s;
}
```

راهنمایی: اعداد فرد دو رقمی از ۱۱ شروع و تا ۹۹ ادامه می یابند. برای حل این مسئله یک متغیر برای ذخیره نمودن اولین عدد فرد دورقمی (۱۱) و متغیری هم برای ذخیره مجموع جملات در نظر می گیریم. در این مسئله با هربار تکرار حلقه دو واحد به عدد قبلی اضافه می کنیم تا عدد فرد بعد از آن بدست آید. مجموع جملات نیز باید جملات قبلی را حفظ نموده و جمله جدید را در کنار آن ثبت کند.

مثال۲۳: برنامه ای بنویسید که مجموع اعداد فرد بین۹-۹۹ را چاپ کند؟

```
#include<iostream.h>
```

```
Main() {
Int i=9,s=0;
i=i+2;
If(i<=99)
  s=s+i;
cout<< s;
}
```

راهنمایی: در این مسئله نیز همچون مسئله قبل برای شمارنده حلقه یک متغیر و برای ذخیره مجموع جملات متغیر دیگری را تعریف می کنیم. و این بار شرط را در وسط حلقه بکار برده و مقدار اولیه شمارنده حلقه را برابر ۹ قرار می دهیم.

مثال۲۴: برنامه ای بنویسید که طول سه پاره خط را گرفته و بررسی کند که آیا می توانند با هم یک مثلث تشکیل دهند یا نه؟

```
#include<iostream.h>
Main() {
        Float  a,b,c;
Cout<< " enter a,b,c:";
Cin>>a>>b>>c;
If(a>=b+c||b>=a+c||c=a+b) {
cout<< "mosallas tashkil mishavad";
else
cout<< "mosallas tashkil  nemishavad";
}
}
```

مثال۲۵: برنامه ای بنویسید که چهار عدد حقیقی را گرفته و ماکسیمم آنرا محاسبه کرده و چاپ کند؟

```
#include<iostream.h>
Main() {
        Int a,b,c,d,max ;
Cout<< " enter a 4 number :";
Cin>>a>>b>>c>>d;
A=max;
If(b>max)
            B=max;
If(c>max)
C=max;
If(d>max)
            D=max;
cout<< "max="<<max;
}
```

مثال۲۶: برنامه ای بنویسید که عدد طبیعی n را گرفته و بررسی کند که عدد کامل است یا نه؟

```
#include<iostream.h>
Main() {
Int  n, s=0, i, m;
Cout<< " enter a number:";
Cin>>n;
For(i=1; i<=n/2; i++) {
     m=n%i;
    If(m==0)
```

```
s=s+i;
}
If(s==n)
cout<< “kamel ast=”;
cuot <<”kamel nist=”;
}
```

راهنمایی: تعریف عدد کامل؛ عددی که مقدار آن برابر با مجموع مقسوم علیه های آن عدد بجز خودش باشد. یعنی ابتدا مقسوم علیه های عدد مورد نظر را بدست آورده (بجز خودش) و سپس با هم جمع می کنیم اگر حاصل جمع مقسوم علیه ها با خودش برابر بود عدد کامل است و در غیر این صورت عدد کامل نیست.

مثال۲۷: برنامه ای بنویسید که عدد طبیعی n را گرفته و بررسی کند که عدداول است یا نه؟

```
#include<iostream.h>
Main() {
Int  n, i, a;
Cout<< “ enter a number:”;
Cin>>n;
For(i=2; i<=n/2; i++) {
      a=n%i;
     If(a==0)
Cout<<"aval nist:";
}
cout<< “aval ast=”;
}
```

راهنمایی: تعریف عدد اول؛ عددی است که فقط به یک و خودش بخش پذیر باشد. اگر بجز یک و خودش به هر عددی بخش پذیر باشد اول نیست. در این برنامه مد به معنای باقیمانده تقسیم عدد بر شمارنده را نشان می

دهد و اگر برابر باشد نشان دهنده بخش پذیر بودن آن بر مقدار شمارنده است. و چون در شرط اول بودن هر عددی بر یک و خودش بخش پذیر است مقدار اولیه شمارنده از۲ شروع می شود. و در شرط حلقه تکرار حلقه تا زمانی که مقدار شمارنده از نصف عدد کوچکتر باشد تکرار می شود چون مقسوم علیه های عدد از یک تا نصف آن را شامل می شوند.

مثال۲۸: برنامه ای بنویسید که n عدد طبیعی را گرفته و ماکسیمم اول و دوم آنها را بدست آورده و چاپ کند؟

```
#include<iostream.h>
Main() {
        int  max1, max2, n, x[30];
Cout<< " enter n:";
Cin>>n;
For(i=1;i<=n;++i)
Cin>>x[ i];
Max1=max2=x[ i];
For(i=2;i<=n;++i)
If( x[ i] > max1) {
                Max2=max1;
                Max1=x[ i];
}
Else
    If(x[ i]>max2)
    max2= x[ i];
cout<< max1<<max2;
}
```

راهنمائی: در این برنامه ابتدا تعداد کل اعداد را از کاربر دریافت نموده و با استفاده از یک آرایه اعداد را دریافت می کنیم. در ابتدا فرض می کنیم ماکسیمم اول و ماکسیمم دوم و اولین عدد با هم برابر اند. سپس اعداد بعدی را با ماکسیمم اول مقایسه می کنیم اگر بزرگتر باشد مقدار ماکسیمم اول را در ماکسیمم دوم قرار داده و عدد بزرگتر را در ماکسیمم اول قرار می دهیم. در غیر این صورت اگر عدد وارد شده از ماکسیمم دوم بزرگتر باشد آنرا در ماکسیمم دوم قرار می دهیم. وسپس مقادیر ماکسیمم اول و ماکسیمم دوم را چاپ می کنیم. دقت کنید که در ابتدا هر دو ماکسیمم با هم برابر اند و در مراحل بعدی تغییر می کنند.

مثال۲۹: برنامه ای بنویسید که حاصل ضرب a*b را با استفاده از جبرهای متوالی بدست آورده و چاپ کند؟

```
#include<iostream.h>
Main() {
        Int a, b, I, s=0;
Cout<< " enter a & b:";
Cin>>a>>b;
For(i=1;i<=b;++i)
s=s+a;
cout<< "pow(a,b)="<<s;
}
```

راهنمائی: برای بدست آوردن ۲ به توان ۳ می توان دو را سه بار با خودش جمع کرده و نتیجه را بدست آورد. در این برنامه ابتدا اعداد را از کاربر دریافت نموده و سپس بطور متوالی آن عدد را با خودش جمع می کنیم این عمل را به تعداد عدد دوم تکرار می کنیم.

مثال۳۰: برنامه ای بنویسید که n عدد طبیعی را گرفته و کامل بودن هر یک را بررسی و چاپ کند؟

```
#include<iostream.h>
Main() {
        Int  i, n, x[30], s=0, c;
Cout<< " enter a number:";
```

```
Cin>>n;
For(i=1;i<=n;++i) {
        Cin>>x[i];
        for(c=1;c<=x[ i] / 2;c++) {
            m=x[ i]%c;
            if(m==0)
            s=s+c;
            }
        If(s==x[ i])
            cout<< “kamel ast=”;
        else
            cout<< “kamel nist=”;
        }
}
```

راهنمائی: در این برنامه ابتدا تعداد کل اعداد را دریافت کرده و با استفاده از یک آرایه مادامی که مقدار شمارنده از تعداد کل اعداد کمتر باشد، اعداد را از کاربر دریافت خواهیم کرد. و در هر مرحله مقسوم علیه های آن عدد را محاسبه می کنیم. اگر بر هر عددی بخش پذیر باشد یعنی باقیمانده تقسیم عدد صفر بود آن عدد را به مقسوم علیه های قبلی اضافه خواهد کرد در صورتی که مقدار عدد مورد نظر با مجموع مقسوم علیه ها برابر باشد عدد کامل و در غیر این صورت کامل نیست. در این برنامه از دو شمارنده حلقه یکی برای حلقه خارجی و دیگری برای حلقه داخلی استفاده شده است.

مثال۳۱: برنامه ای بنویسید که عدد طبیعیn را گرفته وn امین جمله دنباله فیبوناتچی را محاسبه و چاپ کند؟

```
#include<iostream.h>
Main() {
        Int I, n, f[20];
```

```
Cout<< " enter n:";
Cin>>n;
F[1]=1;
F[2]=1;
For(i=3;i<=n;++i)
F[ i ]=f[i-1]+ f[i-2];
cout<< f[ i ];
}
```

راهنمائی: دنباله فیبوناتچی از یک سری اعداد بصورت زیر تشکیل یافته است که در این سری دو جمله اول و دوم آن برابر عدد یک و جمله سوم برابر مجموع دو جمله اول و دوم و به ترتیب جمله چهارم برابر است با مجموع جمله سوم و دوم و ادامه جملات به همین ترتیب ادامه می یابند. لذا در این برنامه ابتدا عددی را از کاربر دریافت نموده و جملات دنباله را تا عدد وارد شده تکرار می کنیم. چون دو جمله اول ثابت و برابر یک هستند مقادیر آنها را در برنامه قرار می دهیم. با استفاده از یک آرایه جمله های بعدی را با مجموع دو جمله قبلی محاسبه نموده و چاپ می کنیم.

مثال۳۲: برنامه ای بنویسید که۵ نمره دانشجویی را گرفته و سپس نمرات بیش از معدل را محاسبه و چاپ کند؟

```
#include<iostream.h>
Main() {
        Int i;
        Float n[5], sum=0, avg;
Cout<< " nomarat ra vared konid:";
For(i=0;i<5;++i) {
                    Cin>>n[ i ];
                    Sum=sum+n[ i ];
                   }
```

```
Avg = sum / 5;
For(i=0;i<5;i++)
If(n[ I ]>avg)
cout<< n[ i ];
}
```

راهنمائی: با توجه به این که اعداد ورودی (نمرات) همگی می توانند از نوع اعشاری و یا صحیح باشند. لذا با استفاده از یک آرایه یک بعدی که تعداد عناصر آن۵ عدد می باشد اعداد را دریافت نموده و سپس با هم جمع کرده و بر تعداد کل تقسیم می کنیم تا میانگین نمرات بدست آید. حال نمراتی را که بیشتر از معدل می باشند محاسبه نموده و چاپ می کنیم.

مثال۳۳: برنامه ای بنویسید که n عدد صحیح را گرفته و اعداد کوچکتر از میانگین را محاسبه و چاپ کند؟

```
#include<iostream.h>
Main() {
        Int   i, n, a[ 30] ;
        Float sum=0, avg;
        Cout<< " enter n:";
        Cin>>n;
        For(i=0;i<n;++i) {
                        Cin>>a[ i ];
                        Sum=sum+a[ i ]; || sum +=a[ i ];
                       }
        Avg = sum / n;
        For(i=0;i<n;i++)
        If(a[ i ]<avg)
        cout<< a[ i ];
```

```
}
```

راهنمائی: چون تعداد اعداد ورودی مشخص نیست و از طرفی در نوشتن برنامه نمی توان مقدار آرایه ها را خالی گذاشت لذا به دلخواه عددی مثل ۳۰ قرار می دهیم. در ابتدای برنامه از کاربر تعداد اعداد ورودی را گرفته وبا توجه به این که اعداد ورودی همگی از نوع صحیح می باشند. لذا با استفاده از یک آرایه یک بعدی که تعداد عناصر آن مثلاً ۳۰ عدد می باشد اعداد را دریافت نموده و سپس با هم جمع کرده و بر تعداد کل تقسیم می کنیم تا میانگین بدست آید. حال نمراتی را که کمتر از میانگین می باشند محاسبه نموده و چاپ می کنیم. در طول برنامه برای محاسبه مجموع اعداد ورودی می توان یکی از دو دستور بالا را استفاده کرد. که نتیجه هر دو آنها با هم برابر است.

مثال۳۴: برنامه ای بنویسید که n نمره دانشجویی را گرفته و سپس انحراف از میانگین هریک از آنها را محاسبه و چاپ کند؟

```
#include<iostream.h>
Main() {
  Int I,n;
        Float  f[20], avg, sum=0 ;
Cin>>n;
For(i=1;i<=n;++i) {
                        Cin>>f[ i ];
                        Sum +=f[ i ];
}
Avg=sum / n;
For(i=1;i<=n;++i)
cout<< f[ i ] – avg ;
}
```

راهنمائی: در این برنامه ابتدا تعداد نمرات را از کاربر دریافت کرده و توسط یک آرایه اعداد را که می توانند اعشار و یا صحیح باشند دریافت نموده و مجموع آنها و سپس میانگین نمرات را بدست آورده وسپس توسط یک حلقه تکرار، اختلاف بین نمره و میانگین را بدست می آوریم.

مثال۳۵: برنامه ای بنویسید که عددی رادر مبنای ۱۰ گرفته و با استفاده از آرایه تبدیل به مبنای ۲ کرده و چاپ کند؟

```
#include<iostream.h>
Main() {
        Int i=0, d, b[20];
Cout<< " enter a number:";
Cin>>d;
While(d>0) {
                B[i]=d %2;
                d=d/2;
                i++;
                }
For(i=i-1;i>=0;++i)
cout<< b[ i];
}
```

راهنمائی: با توجه به تبدیل اعداد ده دهی به دودویی عدد دریافت شده را بطور متوالی به ۲ تقسیم می کنیم و باقیمانده آنرا در یک آرایه ذخیره می کنیم. با استفاده از حلقه مادامی که عدد از صفر بیشتر است تقسیم را ادامه می دهیم. حال با استفاده از یک حلقه تکرار اعداد باقیمانده ها را از آخر به اول یا راست به چپ می نویسیم. و به این ترتیب عدد بدست آمده مبنای دوم عدد مورد نظر را نشان می دهد.

مثال۳۶: برنامه ای بنویسید که دو ماتریسa,b را گرفته وسپس ماتریسc ماتریس حاصل جمع را حساب کند؟

```
#include<iostream.h>
```

```
Main() {
        Int I, j, a[3][4], b[3][4], c[3][4];
Cout<< “matris ra vared konid:”;
///دریافت ماتریس اول
For(i=0;i<3,++i)
For(j=0;j<4;++j)
Cin>>a[ i][ j];
///دریافت ماتریس دوم
For(i=0;i<3,++i)
For(j=0;j<4;++j)
Cin>>b[ i][ j];
///محاسبه ماتریس حاصل جمع
For(i=0;i<3,++i)
For(j=0;j<4;++j)
c[ i][ j]= a[ i][ j]+b[ i][ j];
///چاپ مقادیر ماتریس حاصل جمع
For(i=0;i<3,++i) {
                For(j=0;j<4;++j)
                Cout<<c[ i][ j];
                        cout<< endL;
                                }
}
```

راهنمائی: با توجه به تعریف آرایه دو بعدی ابتدا تعداد سطر ها و سپس تون ها را می نویسیم. شرط جمع بستن دو ماتریس این است که تعداد سطر وستون های آنها با هم برابر باشد. در این صورت تعداد سطر وستون های

ماتریس حاصل جمع نیز با ماتریس های قبلی برابر خواهد بود. در این برنامه ماتریس۴*۳ را بررسی کرده ایم. لذا در قسمت های مشخص شده با استفاده از حلقه تکرار خارجی و داخلی عناصر هر یک از ماتریس ها را دریافت نموده و سپس با هم جمع می کنیم. و در نهایت با استفاده از دستورات مربوطه مقادیر را چاپ می کنیم. پس از چاپ سطر اول به سطر بعدی رفته و عناصر آنرا نیز به ترتیب چاپ می کند.

مثال۳۷: برنامه ای بنویسید که دو ماتریس a,b را گرفته و سپس ماتریس c ماتریس حاصل ضرب را حساب کند؟

```
#include<iostream.h>
Main() {
        Int I, j, a[2][3], b[3][4], c[2][4], k;
Cout<< “matris ra vared konid:”;
///دریافت ماتریس اول
For(i=0;i<2,++i)
For(j=0;j<3;++j)
Cin>>a[ i][ j];
///دریافت ماتریس دوم
For(i=0;i<3,++i)
For(j=0;j<4;++j)
Cin>>b[ i][ j];
/// محاسبه ماتریس حاصل ضرب
For(i=0;i<2,++i)
For(j=0;j<4;++j) {
                c[ i][ j]= 0;
                For(k=0;k<3,++k)
               C[ i][ j]= c[ i][ j]+a[ i][ k] * b[ k][ j];
```

```
                }
Cout<<"matrix a*b:"<<c[ i][ j];
        cout<< endL;
        }
```

راهنمائی: در ضرب ماتریس ها نیز همچون عمل جمع ماتریس ها عمل می کنیم با این تفاوت که در ضرب ماتریس ها تعداد سطر های ماتریس حاصل ضرب برابر با تعداد سطر های ماتریس اول و تعداد ستون های آن با تعداد ستون های ماتریس دوم برابر است. و چون در این مرحله سطر ها را در ستون ها ضرب می کنیم لذا از متغیر دیگری برای محاسبه سطر یا ستون هر کدام از ماتریس ها استفاده می کنیم.

تمرینات

تمرین۱: برنامه ای بنویسید که عدد طبیعیn را گرفته و مجموع زیر را حساب کند؟

$$s = 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n}$$

```
#include<iostream.h>
#include<math.h>
Main() {
          int n,i;
 float s;
Cout<< " enter n:";
Cin>>n;
s=0;
for(i=1;i<=n;i=i+1)
s=s+1.0/I;
cout<< s;
```

```
}
```

راهنمائی: با توجه به صورت مسئله همه جملات هم علامت بوده و صورت همه کسر ها برابر با عدد یک می باشد و تنها متغیری که مقدار آن در هر مرحله تغییر می کند مخرج کسرها ست. دقت کنید که حاصل کسرها می تواند بصورت اعشاری باشد و مطابق برنامه متغیر مورد نظر برابر است با مجموع جملات قبلی به اضافه جمله جدید که در آن عدد یک را بر مقدار متغیر تقسیم می کنیم.

نکته: در زبان سی $\frac{1}{2}=0$ و $\frac{1.0}{2}=0.5$ می باشد.

تمرین۲: برنامه ای بنویسید که عدد طبیعیn را گرفته و مجموع زیر را حساب کند؟

$$s=1+\frac{1}{2!}+\frac{1}{3!}+...+\frac{1}{n!}$$

```
#include<iostream.h>
Main() {
        int n,i;
        longint f=1;
  float s=0;
Cout<< " enter n:";
Cin>>n;
for(i=1;i<=n;i=i+1) {
                        f=f*I;
                        s=s+1.0/f;
                         }
cout<< s;
}
```

راهنمائی: با توجه به صورت مسئله همه جملات هم علامت بوده و صورت همه کسر ها برابر با عدد یک می باشد و تنها متغیری که مقدار آن در هر مرحله تغییر می کند مخرج کسرها ست. دقت کنید که حاصل کسرها می تواند بصورت اعشاری باشد و مطابق برنامه متغیر مورد نظر برابر است با مجموع جملات قبلی به اضافه جمله جدید که در آن عدد یک را بر مقدار فاکتوریل تقسیم می کنیم. برای محاسبه فاکتوریل که می تواند یک عدد طولانی و بسیار بزرگ باشد. آنرا از نوع صحیح طولانی و با استفاده از فرمول مربوطه بدست آورده و در مخرج کسر ها قرار می دهیم.

تمرین۳: برنامه ای بنویسید که عدد طبیعیn را گرفته و مجموع زیر را حساب کند؟

$$s = 1 - \frac{1}{3} + \frac{1}{5} - \ldots \pm \frac{1}{n}$$

```
#include<iostream.h>
#include<math.h>
Main() {
        int n ,k=1 ,i;
  float s=0;
Cout<< " enter n:";
Cin>>n;
for(i=1;i<=n;i=i+2) {
                    s=s+k/I;
                    k= -k;
                   }
cout<< s;
}
```

راهنمائی: با توجه به صورت مسئله علامت جملات در هر مرحله عکس مقدار قبلی می شوند. صورت همه کسر ها برابر با عدد یک می باشد و متغیری که مقدار آن در هر مرحله تغییر می کند مخرج کسرها ست. دقت کنید

که حاصل کسرها می تواند بصورت اعشاری باشد و مطابق برنامه متغیر مورد نظر برابر است با مجموع جملات قبلی به اضافه جمله جدید که در آن عدد یک را بر مقدار متغیر تقسیم می کنیم. در این مسئله برای محاسبه علامت ها از متغیر دیگری استفاده شده است. و مقادیر مخرج در هر محله با عدد دو جمع می شوند.

تمرین۴: برنامه ای بنویسید که عدد طبیعیn را گرفته و مجموع زیر را حساب کند؟

$$s = x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots + \frac{x^n}{n!}$$

```
#include<iostream.h>
#include<math.h>
Main() {
        int n ,x ,i;
       longint f=1;
  float s=0 ,p=1;
Cout<< " enter x,n:";
Cin>>x>>n;
for(i=1;i<=n;i=i+2) {
                        p=p*x;
                        f= f*i;
                        s=s+p / f;
                       }
cout<< s;
}
```

راهنمائی: با توجه به صورت مسئله علامت همه جملات مثبت است و تغییر نمی کند. جمله اول برابر مقدار متغیر مورد نظر و صورت همه کسرها با مقدار مشخص شده تغییر می کنند که توان ها را می توان با یک متغیر و شمارنده بدست آورد و در هر مرحله متغیر را به توان شمارنده رسانید. با توجه به فرمول محاسبه فاکتوریل،

مقدار مخرج را نیز بدست آورده و نهایتاً مجموع جملات قبلی را با مقدار جدید جمع کرده و سری مورد نظر را چاپ می کنیم.

تمرین۵: برنامه ای بنویسید که عدد طبیعیn را گرفته و مجموع زیر را حساب کند؟

$$s = \frac{1}{2} + \frac{2}{3} + \frac{3}{4} + \ldots + \frac{n}{n+1}$$

```
#include<iostream.h>
Main() {
        int n ,i=1;
        float s=0 ;
Cout<< " enter n:";
Cin>>n;
while(i <=n) {
                s=s+i / i+1;
                i++;
                }
cout<< s;
}
```

راهنمائی: با توجه به صورت مسئله علامت همه جملات مثبت است و تغییر نمی کند. و صورت همه کسرها در هر مرحله یک واحد افزایش می کندد و مقدار مخرج در هر مرحله یک واحد بیشتر از صورت می باشد. با توجه به دستور حلقه وایل مادامی که شرط برقرار باشد جملات درون آن تکرار خواهند شد. سپس مجموع جملات قبلی را با مقدار جدید جمع کرده و سری مورد نظر را چاپ می کنیم.

تمرین۶: برنامه ای بنویسید که دو عدد را دریافت نموده و مقدار اولی را در عدد دوم و مقدار دومی را در عدد اول قرار دهد؟(بدون استفاده از متغیرهای اضافه)

```
#include<iostream.h>
```

```
Main() {
        int x ,y;
Cout<< " enter x,y:";
Cin>>x>>y;
 x=x+y;
 y=x-y;
  x=x-y;
cout<< x<<y;
}
```

راهنمائی: با توجه با مسئله دو عدد را از کاربر دریافت نموده و مقادیر آنها را با هم برعکس می کنیم.یعنی:

Y=10; , X=50; آنگاه X=10; , Y=50;

برای نوشتن این برنامه ساده ترین راه حل این است که ابتدا مقدار عدد اول را در متغیر دیگری ذخیره نموده و سپس مقدار عدد دوم را در متغیر اول قرار داده و نهایتاً مقدار متغیر سوم را در متغیر دوم قرار دهیم.

روش دوم این است که ابتدا هر دو مقادیر را با هم جمع نموده و سپس در یکی از متغیرها ذخیره کنیم. و مطابق برنامه فوق مقادیر را بدست آوریم.

The first step is the hardest

اولین قدم سخت ترین قدم است.