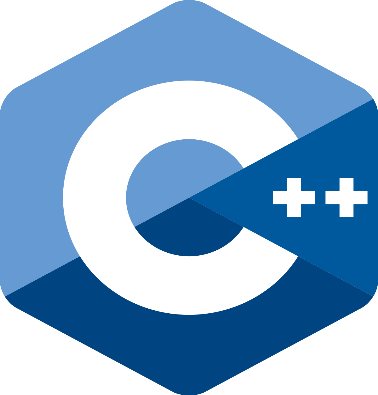


**طرح شده توسط:**

**بهزاد شایق بروجنی**

**معین ورکیانی**



Extra Solutions and Codes



44 + 45

**توضیحات جزوه در زیر آمده است:**

* این جزوه، جزوه‌ی مکمل کلاس و جزوه‌ی استاد کریمیان می‌باشد. با توجه به تکمیلی بودن جزوه، در درسنامه‌ها به اشارات مختصر و مفید بسنده شده و از تکرار خودداری شده است.
* بررسی پاسخ مسائل به کلاس‌های حل تمرین محول شده است.
* پاسخ مسائل شماره دار را در قسمت کدهای جزوه جست‌وجو کنید. پاسخ مسائل با علامت ❖ در جزوه موجود نمی‌باشند و این مسائل صرفاً برای تمرین بیشتر در جزوه قرار داده شده اند. کامنت‌ها و توضیحات در درون پاسخ‌ها قرار داده نشده اند. مسائل شماره‌دار دارای نکته‌ای آموزشی خاص می‌باشند یا از سطح دشوارتری برخوردارند.
* این جزوه، با هدف تکمیل سطح مقدماتی برنامه‌نویسی نوشته شده است، به همین منظور به مقوله‌ی برنامه‌نویسی شیء گرا و کلاس‌ها، علی‌رقم وجودشان در تیترهای درسی ارائه شده وارد نمی‌شود.
* در مقایسه با سرفصل‌های حزوه‌ی استاد کریمیان، این جزوه دارای یک سرفصل اضافه است که حاصل جمع‌آوری نکات از چند بخش جزوه‌ی استاد کریمیان می‌باشد. علاوه بر نکات بالا، سرفصلهای زیر را دربر نمی‌گیرد:

Introduction To Programming

Introduction To C++

Variable Scope

Formatted I/O

Structures

Pointers

علت این امر آن است که دو فصل اول کاملاً تئوری هستند، دو فصل دوم حاوی مطالب مستقل نمی‌باشد و جزوه‌ی اصلی برای این موضوع کاملاً کافیست. مطالب فصل پنجم نیز در فصل Files کنجانده شده، و فصل آخر، در سطح مقدماتی کاربردی نداشته و در سطوح پیشرفته فعالیت دارد.

موفق باشید

مؤلف

***فصل اول***

***Data TypeS***

* **مسائل این قسمت مربوط به بخش Data Types جزوۀ استاد کریمیان می‌باشند و به ترتیب آسان به سخت و همینطور ترتیب مباحث جزوه مرتب شده اند.**
* **4 کادر آموزشی**
* **7 سوال با کد پاسخ**
* **5 سوال بدون کد پاسخ**

***1-1***

*برای نوشتن هر برنامه‌ای لازم است کامپایلر را با دستوراتی آماده‌ی دریافت کد کنید. هر کدام از این دستورات معانی خاصی دارند که درآینده با بعضی از آن‌ها آشنا می‌شوید و به معنای برخی از آن‌ها نیز در سطح مقدماتی نیازی ندارید. به عنوان نمونه برای نوشتن یک برنامه‌ی ساده، مثل برنامه‌ی سوال 1، لازم است عبارات زیر را وارد کنید.*

*#include <iostream>*

*using namespace std;*

*int main(){*

*YOUR CODE*

*return 0;*

*}*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*می‌توان متغیرهایی از هر نوعی که نیاز است (int برای اعداد صحیح، double برای اعداد اعشاری، bool برای صحیح و غلط، char برای کاراکتر و انواع داده‌ای دیگر) قبل از استفاده، یا در اول کد اعلان کرد. برای مثال در زیر، متغیری از نوع double به نام doub تعریف شده است:*

*double doub;*

*می‌توان متغیر‌ها را با عبارت = مقدار دهی کرد:*

*doub = 25.37;*

*همچنین می‌توان چند متغیر از نوع یکسان را با یک دستور اعلان کرد و همچنین می‌توان متغیرها را به هنگام اعلان مقداردهی کرد:*

*bool Ok = true;*

*int a, b, c=12, d;*

*به شیوه‌ی زیر می‌توان مقدار متغیر d را به b نسبت داد یا بصورت زنجیره‌ای چند متغیر را مقداردهی کرد:*

*b = d;*

*a = b = c = d = 12;*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*عبارت #include <iostream> به شما این اجازه را می‌دهد که از دستورات ورودی و خروجی استفاده کنید. دستور خروجی که عبارت یا عباراتی را در نمایشگر سیستم چاپ می‌کند، به صورت زیر عمل می‌کند:*

*cout << WHAT YOU WANT;*

*عبارتی که برای چاپگر فرستاده می‌شود می‌تواند یک متغیر (فرض شده با متغیرها آشنایی دارید.)، یک عدد، یک کاراکتر یا یک عبارت باشد. اگر متغیر باشد، مقدار درون متغیر چاپ می‌شود. اگر یک عدد باشد، خود آن عدد چاپ می‌شود. اگر یک کاراکتر باشد، باید آن را درون ‘ ‘ قرار داد و در کد وارد کرد. مثل دستور زیر که حرف A را چاپ می‌کند:*

*cout << ‘A’;*

*اگر هم قرار بر چاپ یک عبارت باشد، باید آن را درون “ “ قرار داد تا به صورت عینه چاپ شود:*

*cout << “Behzad Shayegh”;*

*با فرستادن عبارت تعریف شده‌ی endl به دستور خروجی، نشانه‌ی نویسنده به اول خط بعد منتقل می‌شود و عبارات بعدی در خط بعد چاپ می‌شوند. می‌توان چند قسمت خروجی را در یک دستور خروجی قرار داد:*

*cout << “Behzad” << endl << “Shayegh” << 92;*

*برخی عبارات که درون “ “ قرار می‌گیرند معانی خاصی دارند. مهم ترین آن‌ها عبارات “\n” و “\t” هستند که اولی با endl یکسان است و دومی یک tab فاصله چاپ می‌کند. برای چاپ خود این عبارات یا حتی کاراکتر «”» کافیست قبل از آن‌ها یک \ قرار دهیم:*

*cout << “my name is:\t \”Behzad\” \nand nothing.”;*

*این دستور عبارت زیر را به نمایشگر می‌فرستد:*

*my name is: “Behzad”*

*and nothing.*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*دستور زیر عبارتی را از ورودی صفحه کلید گرفته و در متغیر name ذخیره می‌کند. متغیر می‌تواند از هر نوعی باشد اما باید با ورودی مطابقت داشته باشد؛ در غیر این صورت برنامه درست کار نمی‌کند.*

*cin >> name;*

*باز هم می‌توان با یک دستور چند عبارت دریافت کرد:*

*cin >> name >> last\_name;*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*در زیر چند دستور و معنای ریاضی آنها به ترتیب تقدم اجرایی لیست شده است:*

*() پرانتز با همان معنای ریاضی*

*\* ضرب*

*/ تقسیم*

*% باقیمانده تقسیم صحیح*

*+ جمع*

*- تفریق*

*عبارت = برای مقداردهی از چپ به راست استفاده می‌شود (درست مثل نوشتار ریاضی، مقدار سمت راست درون متغیر سمت چپ کپی می‌شود.)*

*چند عبارت نیز برای خلاصه نویسی وجود دارد که در زیر آمده است:*

*a++ a=a+1*

*++a a=a+1*

*a-- a=a-1*

*--a a=a-1*

*a+=b a= a+b*

*a-=b a= a-b*

*a\*=b a= a\*b*

*a/=b a= a/b*

*a%=b a= a%b*

*تفاوت دو دستور اول (و همینطور دو دستور دوم) در پیش‌افزاینده یا پس‌افزاینده بودن آنهاست. یعنی اگر از این عبارات در یک دستور دیگر به عنوان جزئی غیرمستقل استفاده کنید، عبارت اول، ابتدا مقدار اولیه‌ی a را استفاده می‌کند و سپس مقدار a را یکی افزایش می‌دهد، اما دومی این دو عمل را با ترتیب برعکس انجام می‌دهد.*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*به مجموعه‌ی دستوراتی که درون آکولاد ({ }) قرار می‌گیرند (معمولاً در چند خط متوالی) یک بلوک گفته می‌شود. می‌توان از بلوک‌بندی برای خوانایی بیشتر برنامه استفاده کرد. یک بلوک برای کامپایلر به منزله‌ی یک دستور واحد خواهد بود و می‌توان هرجایی که اجازه‌ی قرار دادن یک دستور داریم، یک بلوک از دستورات را قرار دهیم (مانند درون ساختارهای if و ... که در ادامه خواهیم خواند. درون بلوک‌ها نیز استفاده از هر دستور و ساختاری مجاز است. نکته‌ی مهم درمورد استفاده از بلوک‌ها، استفاده از متغیر‌های محلی است که از حوصله‌ی این جزوه فراتر می‌رود و مطالعه درمورد آن را به خودتان می‌سپاریم.*

1. برنامه‌ای بنویسید که یک عدد پنج رقمی را بگیرد، ارقام آن را جدا کند و هر یک از ارقام را با سه فاصله بین هرکدام چاپ نماید.

(How to program, Deitel Deitel)

***1-2***

*گفته شد عبارت #include <iostream> اجازه استفاده از دستورات cin و cout را به ما می‌دهد. به iostream یک کتابخانه‌ی c++ گفته می‌شود که با دستور بالا اعلان می‌شود و حاوی توابعی است که با نام و فرم استفاده‌ای خاص، هرکدام اعمال مشخصی را انجام داده و فرایند برنامه نویسی را آسان می‌کنند. این تنها کتابخانه نیست و کتابخانه‌های دیگری نیز برای استفاده‌های مشخص دیگری آماده شده اند. همینطور می‌توان کتابخانه‌هایی شخصی برمبنای نیاز خود نوشت و با دستوراتی دیگر به برنامه لینک کرد که موضوع بحث ما نیست.*

*یکی از کتابخانه‌های پرکاربرد برنامه نویسی، کتابخانه‌ی cmath حاوی توابع ریاضی مانند سینوس و کسینوس است. توجه کنید که این توابع برحسب رادیان عمل می‌کنند. چند تابع پرکاربرد در زیر آمده است:*

*sin(x) = sin x*

*cos(x) = cos x*

*tan(x) = tan x*

*asin(x) = Arcsin x*

*sinh(x) = sinh x*

*exp(x) =*

*log(x) = log x*

*pow(x,y) =*

*sqrt(x) =*

1. برنامه‌ای بنویسید که طول سه ضلع مثلثی بدون نوع خاص (مثل قائم الزاویه) را تحت نام‌های A,B,C دریافت کرده و زاویه‌ی مقابل هر ضلع را تحت نام حرف کوچک متناظر بدست آورده و نمایش دهد.

راهنمایی:

(تألیفی)

* برنامه‌ای بنویسید که مختصات چهار نقطۀ یک چهارضلعی را از کاربر دریافت کرده و مساحت و محیط این چهارضلعی را محاسبه و چاپ نماید.

**راهنمایی**:

این فرمول برای مساحت مثلث به فرمول «هرون» معروف است. می‌توان با کمک گرفتن از پاسخ مسئله‌ی 2 از فرمول روبرو نیز استفاده کرد: که در آن A زاویه بین اضلاع b,c است.

(تمرینات برنامه نویسی با زبان C++، آمادگی امتحان دانشگاه تهران)

***1-3***

*گفته شد نوع داده‌ای وجود دارد به نام bool که می‌تواند حاوی داده‌ی درست (true) یا نادرست (false) باشد که متناظرند با مقادیر 1 و 0. عملگرهای مقایسه، عملگرهایی هستند هوشمند که با استفاده از آنها می‌توان مقادیری از نوع bool دریافت کرد. این عملگرها عبارتند از:*

*A==B که اگر A و B با هم مساوی باشند مقدار true و درغیر این صورت مقدار false را برمی‌گرداند.*

*A!=B که اگر A و B با هم مساوی نباشند مقدار true و درغیر این صورت مقدار false را برمی‌گرداند.*

*A>B که اگر مقدار A از مقدار B بزرگتر باشد (اگر از نوع کاراکتر باشد، کد متناظر بررسی می‌گردد.) مقدار true و درغیر این صورت مقدار false را برمی‌گرداند.*

*A<B که اگر مقدار A از مقدار B کوچکتر باشد (اگر از نوع کاراکتر باشد، کد متناظر بررسی می‌گردد.) مقدار true و درغیر این صورت مقدار false را برمی‌گرداند.*

*A>=B که اگر A بزرگتر یا مساوی B باشد مقدار true و در غیر این صورت مقدار false را برمی‌گرداند.*

*A<=B که اگر A کوچکتر یا مساوی B باشد مقدار true و در غیر این صورت مقدار false را برمی‌گرداند.*

*سه عبارت && , || , ! نیز در ساختن عبارات شرطی پرکاربردند. این عبارات برای به دست دادن یک مقدار bool از مقادیر bool دیگر استفاده می‌شوند. معنای این عبارات به این صورت است:*

*&& = and || = or ! = not*

*که معانی زیر را منتقل می‌کنند.*

*true && true = true*

*true && false = false*

*false && true = false*

*false && false = false*

*true || true = true*

*true || false = true*

*false || true = true*

*false || false = false*

*!(true) = false*

*!(false) = true*

*با روش استفاده از عبارات شرطی در سوالات آشنا می‌شوید.*

***نکته:*** *به تفاوت بین عملگر انتساب ( = ) و عملگر شرط تساوی ( == ) دقت کنید.*

1. برنامه‌ای بنویسید که دو عدد صحیح را دریافت و در متغیرهای a,b ذخیره کند، مقدار بزرگتر و کوچکتر را تشخیص دهد و در متغیرهای max,min ذخیره کند. سپس اگر مقدار بزرگتر بر مقدار کوچکتر بخش‌پذیر بود، پیغام بخش‌پذیر بودن را چاپ کند و درغیر این صورت، علاوه بر چاپ پیغام بخش‌پذیر نبودن باقی‌مانده حاصل تقسیم را چاپ کند.

**گامی به جلو(کادر 1-2):‌** در تمام مراحل کدنویسی این سوال با تصمیم‌گیری های درست یا غلط روبرو هستیم که این مطلب مربوط به قسمت بعدی جزوه است. در همین حد که با نوشتن قطعه‌ی زیر، فقط و فقط درصورت true بودن مقدار bool که عبارت شرطی دستور if می‌باشد، دستورهای درون کروشه‌ها اجرا می‌شوند. پایان هر دستور درون کروشه‌ها به “;” نیاز است.

if (bool) { }

این مسأله از آن جهت در این قسمت مطرح شده که در مسائل ادامه‌ی این فصل به دستور if (حداقل ساده‌ترین صورت آن) نیاز پیدا خواهیم کرد.

(تألیفی)

1. برنامه‌ای بنویسید که یک عدد چهار رقمی را دریافت کند، ارقام آن را از هم جدا کرده و در چهار متغیر جدا از هم ذخیره کند. سپس با دستورالعمل زیر آن مقدار را رمزگذاری کند:

برای هر رقم، اختلاف آن را با عدد 9 درنظر گرفته (بعنوان مثال 3 بجای 6) و سپس با عدد 7 جمع کند. سپس مقدار یکان عدد حاصل را به عنوان رقم جدید ذخیره کند. مقدار رمز گذاری شده عبارت خواهد بود از سری این ارقام جدید با ترتیب برعکس.

(تألیفی)

* برنامه‌ای بنویسید که عدد حاصل از برنامه‌ی سوال 4 را رمزگشایی کند.

راهنمایی: در این مسأله به دستور if که در سؤال 3 توضیح مختصری درمورد آن داده شد نیاز پیدا خواهید کرد.

(تألیفی)

1. جناس قلب، عدد یا متنی است که از هر دو طرف به طور یکسانی خوانده شود. برای مثال هر یک از اعداد پنج رقمی زیر جناس قلب هستند:

45554 , 55555 , 12321 , 11611

برنامه‌ای بنویسید که یک عدد صحیح پنج رقمی را بگیرد و جناس قلب بودن آن را معلوم کند.

(How to program, Deitel Deitel)

* برنامه‌ای بنویسید که سه عدد float غیر صفر را بخواند و معلوم کند آیا این سه عدد می‌توانند نشان دهندۀ اضلاع یک مثلث باشند یا خیر.

(How to program, Deitel Deitel)

1. برنامه‌ای بنویسید که یک عدد صحیح را بخواند و مشخص کند این عدد فرد است یا زوج و نتیجه را چاپ کند.

**گامی رو به جلو(کادر 2-2)**: برای تکرار پذیر کردن برنامه‌تان، می‌توانید قسمتی از کد را که باید تا زمانی که کاربر می‌خواهد تکرار شود (برای این برنامه، این قسمت کد شامل دریافت عدد، تشخیص فرد یا زوج بودن و چاپ نتیجه می‌باشد. توجه کنید اعلان ها در حلقه تکرار قرار نمی‌گیرند.) در ساختار زیر (در بین دو کروشه) قرار دهیم.

do { } while (bool);

توجه کنید بجای bool باید عبارتی شرطی برای ادامه‌ی حلقه قرار دهیم. به عنوان مثال اگر بجای bool عبارت num != 0 را قرار دهیم که در آن num مقدار ورودی در ابتدای آخرین تکرار حلقه باشد، فقط و فقط درصورتی که کاربر مقدار 0 را وارد کند، حلقه برای آخرین بار اجرا می‌شود، پیغام زوج بودن را چاپ می‌کند و برنامه به پایان می‌رسد. این مطلب مربوط به فصل آینده است و مفصل به آن پرداخت خواهد شد اما دراین حد موضوع برای تکرار پذیر کردن برنامه های ساده، لازم و ساده است.

(تألیفی)

1. برنامه‌ای بنویسید که تا زمانی که کاربر مقدار صفر را وارد نکرده، بی وقفه مقادیر ورودی را دریافت کند و درنهایت، پس از وارد شدن مقدار صفر، مقدار max , min این مقادیر را چاپ کند.

**راهنمایی:** از حلقه‌ی do while که در سؤال 6 مطرح شد استفاده کنید و مقدار max , min را با مقدار ورودی در هر حلقه مقایسه کنید و درصورت لزوم مقدار آن‌ها را تغییر دهید. قسمت چاپ خروجی را بیرون حلقه، بعد از آن قرار دهید.

(تألیفی)

* برنامه‌ای بنویسید که تا زمانی که کاربر مقدار صفر را وارد نکرده، بی وقفه مقادیر ورودی را دریافت کند. فرض کنید مقادیر ورودی فقط اعداد صحیح 1 تا 5 هستند. برنامه را طوری بنویسید که درصورت وارد شدن مقداری غیر از این مقادیر، پیغام خطا چاپ کند و این مقدار را بررسی نکند. سپس تعداد تکرار هر یک از این اعداد را بین مقادیر ورودی چاپ کند.

**راهنمایی:** از حلقه‌ی do while که در سؤال 6 مطرح شد استفاده کنید. درون این حلقه به شش دستور if نیاز دارید. یکی برای چاپ پیغام خطا و 5 دستور دیگر برای تشخیص مقدار ورودی بین مقادیر 1 تا 5. برای نادیده گرفتن مقادیر ورودی غیر مجاز نیازی نیست دستوری اضافه کنید به شرطی که تمام دستورات در بلوک‌های if ها جای داده شود. برای تعداد تکرار هر مقدار نیز به متغیری جداگانه نیاز دارید.

(تألیفی)

***1-4***

*گاهی به نوع داده‌ی خاصی نیاز داریم که حاوی مقادیر خاصی باشد. برای ساختن انواع داده از دستور enum استفاده می‌شود. به عنوان مثال می‌خواهیم نوع داده‌ای تعریف کنیم که حاوی اسامی بازیکنان یک تیم 8 نفره‌ی والیبال باشد. می‌نویسیم:*

*enum player {Ali, Ehsan, Amin, Reza, Shayan, Mohsen, Majid, Farhad}*

*از این پس می‌توانیم از نوع داده‌ی جدید player برای اعلان متغیر‌های پذیرای این مقادیر استفاده کنیم:*

*player p1, p2=Amin;*

*p1=p2;*

*اغلب از دستور enum به عنوان یک نوع شمارشی یاد می‌شود چراکه به هر یک از ثابت هایی که تعریف می‌کند (در این مثال اسامی بازیکنان) که با نام شمارشگر از آنها یاد می‌شود یک مقدار صحیح نسبت می‌دهد. این مقادیر به صورت پیشفرض از 0 و به صورت متوالی و به ترتیب به شمارشگر ها تخصیص داده می‌شود اما می‌توان به صورت دستی آنها را مقداردهی کرد:*

*enum player {Ali=3 , Ehsan=5 , Amin, Reza, Shayan, Mohsen, Majid, Farhad}*

*از جایی که مقداردهی متوقف شود، مقداردهی متوالی از آخرین مقدار ادامه پیدا می‌کند.*

* برنامه‌ای بنویسید که برنامه‌ی قبل را برای ورودی روزهای هفته تکرار کند. از دستور enum استفاده کنید. (ورودی به شکل نوشتاری نام روزهای هفته به انگلیسی، با حروف کوچک خواهد بود.)

(تألیفی)

***فصل دوم***

***Control StructureS***

* **مسائل این قسمت مربوط به بخش Control Structures جزوۀ استاد کریمیان می‌باشند و به ترتیب آسان به سخت و همینطور ترتیب مباحث جزوه مرتب شده اند.**
* **5 کادر آموزشی**
* **13 سوال با کد پاسخ**
* **14 سوال بدون کد پاسخ**

***2-1***

*در فصل قبل به طور مختصر با دستور شرطی if آشنا شدید. این دستور به شکل زیر استفاده می‌شود:*

*if (bool) code;*

*روند کار دستور if اینگونه است که ابتدا عبارت شرطی (bool) را بررسی می‌کند و فقط درصورتی که حاصل true بود، دستور code اجرا می‌شود. همانطور که گفته شد می‌توان با استفاده از آکولاد و قرار دادن یک بلوک بجای code چند دستور را با یک ساختار if اجرا کرد.*

*دستور درون ساختار if خود نیز می‌تواند حاوی یک ساختار if دیگر برای ساختن شرایط ویژه باشد. همچنین می‌توان این موضوع را به تمام ساختارهای مورد بحث دراین فصل تعمیم داد به طوری که تمام دستورات تمام ساختار‌ها خود می‌توانند حاوی دستوراتی دیگر یا مشابه باشند.*

*به ساختار زیر دقت کنید:*

*if (bool) code1;*

*else code2;*

*دستور code2 فقط زمانی اجرا می‌شود که حاصل bool نادرست (false) باشد و دستور code1 اجرا نشود. هر ساختار if می‌تواند فقط یک عبارت else را شامل بشود یا نشود. همچنین هر عبارت else فقط به دستور if ماقبل خود اختصاص دارد.*

*با توجه به نکته‌ی بالا و اینکه می‌توان بجای دستور درون if از یک ساختار if دیگر استفاده کرد، می‌توان ساختاری چند دستوری و چند شرطی به صورت زیر نوشت:*

*if (bool1) code1;*

*else if (bool2) code2;*

*else if(bool3) code3;*

*else …*

*else if (bool) code;*

*else code0;*

*بدیهی است که ساختار بالا با همان قوانین کار دستور if else به نحوی کار‌می‌کند که بین عبارات شرطی، از بالا به دنبال عبارت صحیح می‌گردد و اگر پیدا شد، دیگر موارد را بررسی نکرده و دستور را اجرا می‌کند و از ساختتار خارج می‌شود. همچنین اگر هیچکدام از شرط ها درست نبود، دستور code0 اجرا می‌شود. از این دستور معمولاً به نحوی استفاده می‌کنند که عبارات شرطی نقطه‌ی اشتراکی نداشته باشند، چون فقط اولین عبارت درست درنظر گرفته می‌شود.*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*برای عملی که ساختار if else های پیاپی انجام می‌دهند، ساختار دیگری وجود دارد. به ساختار زیر توجه کنید:*

*switch (choice){*

*case choice\_A: code1; break;*

*case choice\_B: code2; break;*

*case …*

*default: code0;*

*}*

*روند کار این دستور اینگونه است که مقدار درون متغیر choic (که می‌تواند از هر نوع داده‌ای باشد) را با مقادیر choic\_A , choic\_B , … (که باید با choic همنوع باشند) مقایسه می‌کند و به محض رسیدن به اولین تساوی بین این مقادیر (ترتیب بررسی از بالا به پایین است)، دستورات را درست از همان نقطه ادامه می‌دهد. به عنوان مثال اگر choic == choic\_B باشد، ابتدا دستور code2 و سپس دستورات بعد از آن اجرا می‌شوند. توجه کنید که اگر دستور break را که بعداً درمورد آن بحث خواهد شد، در پایان دستورات هر case قرار ندهید، دستورات case های بعدی نیز تا آخر اجرا می‌شوند! وظیفه‌ی دستور break این است که روند اجرای دستورات را به بیرون از ساختار switch منتقل می‌کند.*

*می‌توان در این ساختار یک case پیشفرض نیز برای زمانی که هیچ case مناسبی پیدا نشد تعریف کرد. برای اینکار کافیست درست همانند بالا از واژه‌ی default استفاده کرد. توجه کنید که در پایان دستورات این case به break نیازی نیست چون این case آخر است و بعد از آن، ساختار به اتمام می‌رسد.*

1. داده‌های زیر را درنظر بگیرید.

A B E H J Q

C F I K

D G L R S T X

M N O P Z

توضیح: فلش های رو به چپ 10 امتیاز، رو به راست 20 امتیاز و فلش های مورب 35 امتیاز دارند.

با استفاده از تعداد زیادی دستور if و else تودرتو، برنامه‌ای بنویسید که کاربر را در ابتدا در نقطه A درنظر بگیرد و با ارائه قدرت انتخاب مسیر در هر مرحله به کاربر، مسیری را طی کند و به محض رسیدن به نقطه‌ای بدون خروجی، به این ساختار انتخابی پایان دهد. سپس مجموع امتیاز کسب شده توسط کاربر را چاپ کند. (حرکت در خلاف جهت فلش‌ها ممکن نیست.)

(تألیفی)

* برنامه سؤال 8 را به گونه‌ای بازنویسی کنید که کاربر درهر نقطه‌ی دلخواهی توانایی پایان دادن به بازی و مشاهده‌ی نتیجه را داشته باشد.

**راهنمایی:** برای این کار، باید در هر نقطه قدرت انتخاب پایان را قرار دهید و درصورت انتخاب این گزینه دستورهای چاپ نتیجه و پایان برنامه (return 0;) اجرا شوند. هدف این سؤال استفاده از چند دستور پایان برنامه درون دستور های if می‌باشد که اصلاً در برنامه نویسی ساخت یافته پیشنهاد نمی‌شود. در ادامه، این مسئله را از روش های ساخت یافته یا ساخت نیافته دیگری نیز مطرح خواهیم کرد.

(تألیفی)

1. برنامه‌ای بنویسید که 5 نمره در محدودۀ 0 تا 20 دریافت کند و در متغیرهای m1 , m2 , m3 , m4 , m5 ذخیره کند. هربار درصورت وارد شدن عددی خارج از محدوده، پیغام خطا چاپ کند و دوباره عدد را دریافت کند. سپس درصورت دوباره وارد شدن مقدار خارج از محدوده پیغام خطای دیگر چاپ کرده و به برنامه پایان دهد (این دستور بررسی را باید پنج بار کپی کنید. درادامه با استفاده از حلقه‌های تکرار می‌توان این کار را ساده‌تر انجام داد.). هربار پس از دریافت هر نمره، ضریب آن نمره را نیز دریافت و در متغیرهای n1 , n2 , n3 , n4 , n5 ذخیره کند. این ضرایب نیز تنها می‌توانند اعداد صحیح از 0 تا 4 باشند. برای مقادیر خارج از محدوده این ضرایب نیز مراحل قبل را اجرا کنید.

سپس میانگین وزن‌‌دار (معدل) آنها را محاسبه کرده و در متغیر w ذخیره کنید. با استفاده از ساختار چند انتخابی else if نمره معدل را به صورت زیر نمایش دهید:

0 تا 3.99 حرف E، 4 تا 7.99 حرف D، 8 تا 11.99 حرف C، 12 تا 15.99 حرف B، 16 تا 19.99 حرف A، 20 حرف A+.

(تألیفی)

* برنامه‌ی سؤال 9 را با ساختار چند انتخابی switch بازنویسی کنید.

**راهنمایی:** برای اینکار، مقدار w را تقسیم بر 5 کرده و مقدار صحیح (قدرمطلق) آن را مورد سنجش در ساختار switch قرار دهید.

**گامی رو به جلو:** برای بدست آوردن مقدار صحیح یک مقدار اعشاری، می‌توانید با تغییر نوع داده آن را به نوع int برگردانید تا مقدار اعشاری حذف شود. با دستور زیر، W مقدار صحیحِ مقدار اعشاری w خواهد بود.

int W = int(w);

(تألیفی)

***2-2***

*در برنامه نویسی ساختار‌هایی برای تکرار دستورات وجود دارد. یکی از این ساختارها ساختار while است که شکل کلی آن به صورت زیر می‌باشد:*

*while (bool)*

*code;*

*که همانطور که قبلاً هم توضیح داده شد، میتوان به جای code بلوکی از دستورات را قرار داد که می‌تواند حاوی ساختارهای دیگر یا ساختار while دیگری باشد. این حلقه به این صورت کار می‌کند که در هربار استفاده، اول عبارت شرطی bool را بررسی می‌کند و فقط درصورت true بودن نتیجه، دستورات درون ساختار را اجرا می‌کند. پس از پایان این دستورات، بار دیگر شرط را بررسی می‌کند و تا زمانی که مقدار true باشد، تکرار ادامه می‌یابد. بدیهی است که با قرار دادن شرطی همواره درست (یا عبارت true) به جای bool، یک حلقه‌ی بینهایت به دست خواهیم آورد. توجه کنید که برای به دست آوردن یک حلقه‌ی محدود، باید عبارت شرطی را با متغیرهایی تعریف کنیم که درون دستورات تغییر کنند تا سرانجام به حالتی برسند که شرط مقدار نادرست را برگرداند.*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*یک ساختار تکرار دیگر در c++ دستور do while است که بسیار شبیه while کار می‌کند. به ساختار زیر توجه کنید:*

*do*

*code;*

*while(bool);*

*که می‌توان به جای code یک بلوک از دستورات را قرار داد. این ساختار نیز مانند ساختار while تا زمانی که مقدار bool مقدار true باشد، تکرار را ادامه می‌دهد. تنها تفاوت این دو ساختار این است که ساختار do while برای بار اول دستورات، شرط را بررسی نمی‌کند و به این صورت، دستورات حداقل یکبار انجام می‌شوند، حتی اگر شرط از ابتدا نادرست باشد. بقیه‌ی ویژگی ها و قابلیت‌های این دو ساختار درست مثل یکدیگر است.*

1. برنامه‌ای بنویسید که بدون استفاده از کتابخانه cmath مقدار x به توان y را حساب کند. در این برنامه از ساختار تکرار while استفاده کنید. (مقدار y را صحیح دریافت کنید.)

(How to program, Deitel Deitel)

* برنامه‌ای بنویسید که حاصل ضرب اعداد صحیح فرد از 1 تا 15 را محاسبه و چاپ کند.

(How to program, Deitel Deitel)

1. برنامه‌ای بنویسید که با استفاده از حلقه تکرار while، یک مقدار صحیح مثبت را دریافت کرده و مشخص کند که آن مقدار یک عدد اول است یا خیر. با ساختار do while به این برنامه قابلیت تکرار بدهید.

(تألیفی)

1. با استفاده از حلقه تکرار while، تابع فاکتوریل را بازسازی کنید.

(تألیفی)

1. با استفاده از سری نامحدود زیر، مقدار را محاسبه نمایید:

مقدار را با تقریب n جمله از این سری نشان دهید که n توسط کاربر وارد شود.

(How to program, Deitel Deitel)

* کد سؤال 13 را به نحوی بازنویسی کنید که کاربر بجای وارد کردن تعداد جملات، مقدار اختلاف دو جملۀ آخر را وارد کند.

**راهنمایی:** باید در هر حلقه، مقدار قبلی نیز ذخیره شود و در عبارت شرطی حلقه while، اختلاف مقدار قبلی و مقدار جدید را کوچکتر از مقدار ورودی قرار داد.

(تألیفی)

***2-3***

*یکی دیگر از ساختارهای تکرار، حلقه‌ی for است. برای آشنا شدن با روند کار این ساختار به مثال زیر توجه کنید:*

*for (int i = 1; i <= 10; i += 2)*

*cout << i << endl;*

*این حلقه، اعداد فرد کوچکتر از 10 را چاپ می‌کند. همانطور که می‌بینید، این حلقه سه ورودی دریافت می‌کند. اولین آنها برای اعلان و مقداردهی یک شمارشگر است که می‌توان در بلوک دستورات داخل ساختار نیز از آن استفاده کرد. می‌توانیم اگر می‌خواهیم متغیری از پیش اعلان شده را استفاده کنیم، این بخش را خالی بگذاریم اما به ; نیاز خواهد بود. دومین آنها شرط ادامه‌ی حلقه است که درست مانند شرط حلقه‌ی while عمل می‌کند و سومین آنها دستوریست که در آخر هر دور اجرای حلقه، اجرا می‌شود و برای رساندن شرایط به شرط توقف از آن استفاده می‌شود. اگر قرار است این دستور تکرار شونده نیز در میان دیگر دستورات اجرا شود، باز هم می‌توان جای آن را خالی گذاشت (توجه کنید که به ; نیاز نیست). بدیهیست با نوشتن for (;;) به یک حلقه‌ی نامحدود دست پیدا می‌کنیم.*

*نیازی به یادآوری نیست که می‌توان به جای یک دستور اجرایی در این حلقه نیز از یک بلوک استفاده کرد و بدین وسیله می‌توان حلقه‌های تودرتو آفرید.*

1. اعداد اول از 2 تا N که N توسط کاربر وارد می‌شود را با استفاده از ساختار for جمع کنید. برای تشخیص اعداد اول نیز از یک حلقه for دیگر، یک مقدار bool که درون این حلقه for مقدار دهی می‌شود و یک عبارت شرطی if با همان مقدار bool قبلی به عنوان عبارت شرطی درون حلقه اصلی استفاده کنید. با ساختار do while به این برنامه قابلیت تکرار بدهید.

(تألیفی)

* با استفاده از حلقه while نوشته شده در سؤال 11 و یک حلقۀ for، برنامه‌ای بنویسید که یک مقدار را دریافت کرده و اعداد اول کوچکتر از آن را به ترتیب از کوچک به بزرگ در یک سطر و یک tab فاصله بین آنها چاپ کند.

(تألیفی)

* کد سؤال‌های 10 تا 13 را با استفاده از حلقه for بازنویسی کنید.

(تألیفی)

1. با استفاده از توابع توان و فاکتوریل که در سؤال قبل نوشتید، برنامه‌ای بنویسید که مقدار را تا n رقم اعشار که توسط کاربر وارد می‌شود، با استفاده از فرمول زیر محاسبه کند:

**راهنمایی:** باید دو بار از تابع توان استفاده کنید؛ یکبار درون حلقه به عنوان توان x و یکبار درون عبارت شرطی حلقه که اختلاف دو جمله را کوچکتر از . از آنجایی که هنوز با ساختن کد توابع آشنا نشدید و مجبورید مستقیماً ساختار for را کپی کنید، این ساختار را برای اولین استفاده توان اجرا کنید و برای عبارت شرطی، از تابع توان درون کتابخانه‌ی cmath استفاده کنید. (تابع فاکتوریل نیز در این کتابخانه موجود است.)

(تألیفی)

* برنامه‌ای بنویسید که N و x را از کاربر دریافت کرده و مقدار sin(x) را با جمع N حمله اول سری تیلور زیر محاسبه نمایید.

(تمرینات برنامه نویسی با زبان C++، آمادگی امتحان دانشگاه تهران) (امتحان دانشگاه تهران)

1. با استفاده از حلقه‌های for تودرتو برنامه‌ای برای حل دستگاه 3 معادله 3 مجهول با استفاده از روش گاوس جردن بنویسید.

**راهنمایی:** روش گاوس جردن:

دستگاه معادلات را به شکل ماتریسی مقابل می‌نویسیم: CX = B

آنگاه ماتریس زیر را بازسازی می‌کنیم:

سپس با دستورات سطری مقدماتی آن را به یک ماتریس بالا مثلثی تبدیل می‌کنیم. اگر R1 , R2 , R3 را نماد سطرها بدانیم، اینگونه عمل می‌کنیم:

و با داده های جدید:

سپس با روش مشابه، آن را به یک ماتریس قطری مانند زیر تبدیل می‌کنیم:

پس جواب‌ها به صورت زیر خواهد بود:

برای مشکل تقسیم بر صفر نیز راه حلی پیدا کنید. (تعویض سطر مجاز است.)

(تألیفی)

***2-4***

*دو دستور خاص برای ساختن حلقه‌های کامل‌تر و با قابلیت‌های بیشتر وجود دارد که در زیر به آنها می‌پردازیم:*

*1. دستور break; که هرجایِ بدنه‌یِ بلوکِ دستوراتِ یک حلقه قرار بگیرد و اجرا شود، دستورات بعدی بدنه نادیده گرفته شده و حلقه بدون درنظر گرفتن شرط ادامه‌ی حلقه، خاتمه می‌یابد و اولین دستور بعد از حلقه اجرا خواهد شد. لذا این دستور را معمولاً در یک ساختار if قرار می‌دهند. می‌توان حلقه‌های بینهایت را به این روش تحت حالتی خاص خاتمه داد. (حال به استفاده از این دستور در ساختار switch دقت کنید.)*

*2. دستور continue; که هرجایِ بدنه‌یِ بلوکِ دستوراتِ یک حلقه قرار بگیرد و اجرا شود، دستورات بعدی بدنه نادیده گرفته شده و دوباره شرط ادامه‌ی حلقه بررسی می‌شود؛ اگر شرط هنوز برقرار بود، حلقه ادامه می‌یابد و در غیر اینصورت از آن خارج می‌شود. این یعنی بازگشت زودهنگام.*

*این دو دستور قادرند انعطاف بسیار زیادی به حلقه‌ها بدهند ازجمله چند شرطی کردن آنها.*

* کد سؤال اضافه‌ی پس از سؤال 13 را با استفاده از یک حلقه for یا while بینهایت بازنویسی کنید و شرط پایان حلقه را درون یک ساختار if درون حلقه با دستور break قرار دهید.

(تألیفی)

* بازی هوپ: این بازی چند نفره‌ی خانوادگی به این صورت است که در ابتدا یک مقدار پرش دلخواه برای آن انتخاب می‌کنند. سپس افراد به ترتیب، اعداد را از یک شمرده و بجای ضرایب صحیح مقدار پرش، عبارت هوپ را به زبان می‌آورند تا زمانی که یک نفر خطا کند. به عنوان مثال با مقدار پرش 4، بازیکنان به ترتیب اعداد را اینگونه می‌شمارند:

1 , 2 , 3 , hop , 5 , 6 , 7 , hop , 9 , 10 , 11 , hop , 13 , …

با استفاده از یک حلقه‌ی بینهایت for یا while، برنامه‌ای بنویسید که خروجی این بازی را با دریافت مقدار پرش و آخرین عدد بازی (دلخواه توسط کاربر) چاپ کند. برای چاپ hop از دستور if و continue استقاده کنید و برای خاتمه‌ی بازی درزمان رسیدن به مقدار آخرین عدد، از break استفاده کنید. هر عدد یا کلمه را در یک سطر چاپ کنید.

(تألیفی)

***2-5***

*دستور goto یک دستور با انعطاف بسیار بالاست که برنامه‌ها را از حالت ساخت یافته خارج می‌کند. یعنی برای توسعه‌ی برنامه‌ها مشکلاتی ایجاد می‌کند و بنابراین استفاده از آن پیشنهاد نمی‌شود اما این یک دستور عالی برای برنامه‌های کوتاه است. قبل از توضیح دادن این دستور باید اخطار داده شود که این دستور می‌تواند برای برنامه‌ی شما مهلک باشد پس در استفاده از آن دقت کنید.*

*شما می‌توانید قسمت‌های مختلف برنامه‌ی خودتان را به صورت زیر مارک (mark) کنید:*

*Dog:*

*code*

*code*

*Cat:*

*code*

*code*

*به این صورت، قسمت‌های مختلف برنامه‌ی شما با اسامی دلخواه خودتان مارک می‌شود. با دستوراتی که تا کنون یاد گرفتیم، این امر مشکلی برای برنامه‌ی شما ایجاد نمی‌کند.*

*شما می‌توانید با قرار دادن دستور goto Mark; در هر جای برنامه (درون هر ساختاری و تحت هر شرایطی) و نوشتن نام یک مارک در برنامه‌ی خودتان بجای Mark، روند اجرای دستورات را به آن قسمت (سطر) برنامه منتقل کنید. این دستور می‌تواند برنامه را به جلو یا عقب ببرد. گاهی اوقات لازم است در شرایطی، درون ساختارهای تودرتو، از چند ساختار سری خارج شویم. اینکار با کمک دستور break بسیار دشوار خواهد بود درصورتی که با دستور goto به سادگی انجام خواهد شد.*

1. شکل زیر را درنظر بگیرید:

برنامه‌ای بنویسید که کاربر را با 0 امتیاز در نقطۀ A قرار دهد. سپس به کاربر اختیار انتخاب مسیر (فقط از بین گزینه‌های موجود با درنظر گرفتن جهت فلش‌ها) بدهید و در هر مرحله، امتیازی معادل اندیس نقطه‌ای که بدان داخل می‌شود به امتیاز وی بیافزایید. توجه کنید که تقاطع بین خطوط بین تمامی مسیرها مشترک و قابل استفاده باشد. برنامه باید زمانی به پایان برسد که کاربر دیگر حق انتخابی نداشته باشد؛ یعنی در 9 نقطه. (اجازه‌ی برگشت فوری بر روی مسیرهای دوطرفه بلامانع است.)

**راهنمایی:** از تعداد زیادی ساختار go to استفاده کنید.

(تألیفی)

* برنامه مسئله 17 را به گونه‌ای تکمیل کنید که درصورت بازگشت فوری بر روی مسیرهای دوطرفه، امتیاز کسب شده دوبار کم شود. همچنین اختیار پایان بازی درهرنقطه دلخواه و نمایش نتیجه را نیز اضافه کنید.

**راهنمایی:** برای قسمت اول سؤال به چند ساختار if برای هر نقطه نیاز دارید. برای قسمت دوم نیز می‌توانید به روش سؤال اضافه‌ی بعد از سؤال 8 عمل کنید یا اینکه از یک ساختار go to که به پایان برنامه می‌رود استفاده کنید.

(تألیفی)

* برنامه مسئله 17 را به شکل یک بازی با امتیاز اولیه‌ی 25 درآورید به صورتی که با وارد شدن به هرنقطه، امتیازی متناظر با اندیس آن نقطه از امتیاز کاربر کم شود و کاربر فقط و فقط درصورت رسیدن به یکی از 9 نقطه‌ی بدون خروجی برنده‌ی بازی شده و پیغام پیروزی را مشاهده کند. در غیر این صورت، اگر در میان راه امتیاز کاربر تمام شد، کاربر پیغام شکست را مشاهده کرده و برنامه پایان یابد.

(تألیفی)

* برنامه‌ی سؤال 9 را بدون کپی کردن قسمت‌های غیر ضروری و جایگزین کردن حلقه‌های تکرار بجای آنها بازنویسی کنید.

(تألیفی)

* برنامه‌ای بنویسید که از میان چند عدد صحیح، کوچکترین آنها را پیدا کند. فرض کنید که اولین مقداری که خوانده می‌شود تعداد اعدادی را که باید وارد شوند نشان می‌دهد.

**راهنمایی:** از پاسخ سوال بعد کمک بگیرید.

(How to program, Deitel Deitel)

1. برنامه‌ای بنویسید که ابتدا مقدار N را از کاربردریافت کرده و سپس به تعداد N عدد را از کاربر دریافت و سومین بزرگترین عدد بین این N عدد را چاپ نماید.

(تمرینات برنامه نویسی با زبان C++، آمادگی امتحان دانشگاه تهران) (امتحان دانشگاه تهران)

1. برنامه‌ای بنویسید که ابتدا مقدار N را از کاربر دریافت کرده و سپس N عدد از کاربر دریافت و واریانس آنها را محاسبه و چاپ نماید.

**راهنمایی:** به دلیل اینکه شما هنوز با آرایه‌ها آشنایی ندارید، مجبورید این N مقدار را دو بار از کاربر دریافت کنید. این موضوع در این مثال ایرادی ندارد. می‌توانید از تابع توان در کتابخانه‌ی ریاضی استفاده کنید.

(تمرینات برنامه نویسی با زبان C++، آمادگی امتحان دانشگاه تهران)

* اعداد دوم به این صورت تعریف می‌شوند: اعدادی که دقیقاً دو مقسوم علیه کوچکتر از خودشان و بزرگتر از 1 دارند. برنامه‌ای بنویسید که یک عدد به عنوان ورودی دریافت کند و چنانچه این عدد «عدد دوم» بود، پیغام مناسب چاپ کند.

(امتحان دانشگاه تهران، 22/8/1393)

1. برنامه‌ای بنویسید که اندازۀ یک ضلع از مربعی را بخواند و با استفاده از ستاره و فاصله، یک مربع توخالی به اضلاعی با همان اندازه چاپ کند. برنامۀ شما باید برای اضلاع با اندازه‌های بین 1 تا 20 کار کند.

(How to program, Deitel Deitel)

***فصل سوم***

***ArrayS***

* **مسائل این قسمت مربوط به بخش Arrays جزوۀ استاد کریمیان می‌باشند و به ترتیب آسان به سخت و همینطور ترتیب مباحث جزوه مرتب شده اند.**
* **3 کادر آموزشی**
* **11 سوال با کد پاسخ**
* **9 سوال بدون کد پاسخ**

***1-3***

*در بسیاری مواقع پیش می‌آید که ما به لیستی از متغیرها با نام مشترک برای سهولت کار نیاز پیدا می‌کنیم. گاهی نیز موضوع از بحث سهولت خارج شده و به اجبار می‌رسیم. برای حل این موضوع، بهتر است با آرایه‌ها در برنامه نویسی آشنا شویم. آرایه‌ها برای اعلان چند متغیر تحت نامی یکسان ایجاد شده‌اند. طرز کار آرایه‌ها مثل کار با متغیر‌های اندیس‌دار در ریاضیات است. به عنوان مثال به عبارت زیر توجه کنید:*

*int Ark[10];*

*این عبارت آرایه‌ای از متغیر‌ها از نوع int را تعریف می‌کند. اندازه‌ی آرایه، درون دو کروشه نوشته شده است، یعنی 10. این به معنی اعلان 10 متغیر می‌باشد. می‌توانیم به این متغیر‌ها به صورت Ark[0], Ark[1], Ark[2], … , Ark[9] دست پیدا کنیم. توجه کنید که شماره‌ی این متغیر‌ها که درون کروشه نوشته می‌شود و حکم همان اندیس را دارد از صفر شروع می‌شود و به صورت متوالی، تا یک واحد کمتر از اندازه‌ی آرایه، ادامه پیدا می‌کند.*

*می‌توان اعضای آرایه را یک به یک مقداردهی کرد یا اینکه همه‌ی آنها را به صورت زیر در زمان اعلان مقداردهی کرد:*

*int Ark[10] = {0,1,2,2,1,0,0,1,2,2};*

*با این دستور، اعضای آرایه، به ترتیب با مقادیر داده شده، از سمت چپ مقداردهی می‌شوند. اگر تعداد مقادیر داده شده از اندازه‌ی آرایه کمتر باشد، از آن عضو آرایه که دیگرمقداری برای دریافت ندارد به بعد، همه مقادیر 0 را اخذ می‌کنند. پس می‌توان به صورت زیر همه‌ی اعضای یک آرایه را برابر صفر قرار داد:*

*int A[200000] = {0};*

*به دو دستور زیر که هم معنی‌اند توجه کنید:*

*int Ark[10] = {0,1,2,2,1,0,0,1,2,2};*

*int Ark[ ] = {0,1,2,2,1,0,0,1,2,2};*

*این بدان معناست که اگر اندازه‌ی آرایه را وارد نکنید اندازه‌ی آن برابر با تعداد مقادیر ورودی در مقداردهی زمان اعلان خواهد بود.*

*توجه کنید چه زمان اعلان به عنوان اندازه‌ی آرایه و چه زمان استفاده از اعضای آرایه‌ها، اندیس آرایه می‌تواند یک متغیر مداردهی شده باشد که دراین صورت هم‌ارز با مقدار آن عمل خواهد کرد. به این صورت می‌توان تعداد دلخواهی از متغیر‌ها را درمیان برنامه اعلان کرد یا با یک دستور، به متغیر‌های مختلفی دسترسی پیدا کرد. یکی از بزرگترین استفاده‌های این ویژگی، استفاده از آنها در حلقه‌ها و استفاده از شمارشگر حلقه به عنوان اندیس آرایه می‌باشد که به این صورت، در هربار تکرار حلقه، به یک متغیر دیگر دسترسی پیدا می‌کنیم. مثال اینگونه استفاده‌ها را در کد‌های پاسخ مسائل بعدی خواهید دید.*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*می‌توان آرایه‌ای از آرایه تعریف کرد و به این صورت یک آرایه‌ی دوبعدی ساخت. می‌توان آرایه‌ی دوبعدی را به صورت یک جدول تصور کرد. نحوه‌ی اعلان یک آرایه‌ی دوبعدی درست مثل یک آرایه‌ی تک بعدی با دو اندیس (کروشه) است:*

*int A[5][3];*

*یک آرایه‌ی 5 در 3 تشکیل می‌دهد. روند دسترسی به اعضا نیز درست مثل قبل اما با دو اندیس است. تنها نکته‌ای که درمورد آرایه‌های دوبعدی وجود دارد، مقداردهی زمان اعلان است. به سه مقدار دهی زیر که هم معنی‌اند دقت کنید:*

*int A[3][2]; A[0][0]=1; A[0][1]=A[1][0]=1; A[1][1]=A[2][1]=2; A[2][0]=3;*

*int A[3][2]={ {0,1},{1,2},{0,2} };*

*int A[3][2]={0,1,1,2,0,2}*

*همینطور می‌توان به نحو مشابه، آرایه‌های سه بعدی یا بیشتر ساخت و استفاده کرد.*

1. برنامه‌ای بنویسید که سوال 19 را به گونه‌ی دیگری حل کند. ابتدا آرایه‌ای با اندازه‌ی N ایجاد و سپس همه‌ی داده ها را دریافت کند. سپس با داده های موجود در آرایه، واریانس را حساب کند.

(تألیفی)

1. برنامه‌ای بنویسید که با استفاده از حلقه‌ی do while به گونه‌ای که در سؤال 6 آمده پیوسته نمرات دانشجویان را از نوع double دریافت کند و با وارد کردن نمره‌ی -1 به برنامه پایان دهد. با استفاده از ساختار switch (یا هر روش دیگری) برای هر ورودی، فراوانی نمرات را در محدوده‌ی هر عدد صحیح مشخص کنید (جزء صحیح نمرات را بررسی کنید. برای این کار از تبدیل نوع استفاده کنید.) و درصورت وارد شدن نمره‌ای خارج از محدوده‌ی 0 تا 20 و مقدار خاص -1، پیغام خطا چاپ کرده و حلقه را تکرار کنید. درنهایت فراوان ترین دسته را به همراه مقدار فراوانی آن چاپ کرده و سپس لیستی از محدوده ها و دربرابر آنها، مقدار فراوانیشان چاپ کنید. در برابر هر فراوانی در همان خط، با استفاده از علامت \* به تعداد فراوانی آن دسته، شبیه‌ساز نمودار میله‌ای آنها را نیز رسم کنید.

**راهنمایی:** برای ذخیره‌ی مقادیر فراوانی باید از آرایه‌ها استفاده کنید.

(تألیفی)

1. برنامه‌ای بنویسید که تعداد N عدد صحیح (که N توسط کاربر وارد می‌شود) را دریافت کرده و در آرایه‌ای ذخیره کند. سپس مد و تعداد تکرار مد را مشخص کند.

**راهنمایی:** به دو آرایه نیاز دارید.

(تألیفی)

1. (مرتب کردن حبابی) فرض کنید می‌خواهیم آرایه‌ای از اعداد را از کوچک به بزرگ در همان آرایه مرتب کنیم. یک روش برای این کار مرتب کردن حبابی است. به اینگونه که عضو اول و دوم مقایسه شده و اگر عضو دوم کوچکتر بود، جای آن دو را تعویض می‌کنیم. سپس این کار را برای عضو دوم و سوم و همینطور الی آخر انجام می‌دهیم. بدین صورت بزرگ‌ترین عضو به خانه‌ی آخر منتقل می‌گردد. کافیست به تعداد اندازه‌ی آرایه (یا حتی یکبار کمتر) این عملیات را انجام دهیم تا مقادیر مرتب شوند. برنامه‌ای برای این کار بنویسید. (فرض کنید آرایه 10 عضوی است و مقادیر آن توسط کاربر وارد می‌گردد.)

**راهنمایی:** برای تعویض دو عضو به یک متغیر جدید برای حمل موقت یکی از مقادیر نیاز دارید.

(How to program, Deitel Deitel)

* مرتب کردن حبابی که در سؤال 24 مطرح شد، برای آرایه‌های بزرگ کارآمد نیست. تغییرات سادۀ زیر را اعمال کنید تا کارایی برنامه بهبود یابد:

الف) پس از عبور اول، مطمئناً بزرگترین عدد در عنصرِ با بزرگترین اندیس قرار می‌گیرد؛ پس از عبور دوم، دو عددی که از همه بزرگترند «در جای مطلوب» قرار می‌گیرند و الی آخر. برنامۀ مرتب کردن حبابی را به گونه‌ای تغییر دهید که به جای انجام 9 مقایسه درهر عبور، در عبور دوم 8 مقایسه، در عبور سوم 7 مقایسه و ... انجام دهد.

ب) داده‌های موجود در آرایه ممکن است از قبل کاملاً یا تقریباً مرتب بوده باشند. در این صورت چرا 9 عبور انجام دهیم درحالی که تعداد کمتری کفایت می‌کند؟ برنامه را طوری تغییر دهید که در انتهای هر عبور بررسی کند که آیا در طول این عبور، جابجایی انجام گرفته است یا خیر. اگر جابجایی صورت نگرفته است دیگر نیازی به ادامۀ مرتب سازی نیست و داده ها مرتب‌اند.

**راهنمایی:** برای قسمت (ب) به یک متغیر از نوع bool درون شرط تشخیصِ نیاز به جابجایی نیاز دارید.

(How to program, Deitel Deitel)

* سؤال 18 را با ذخیرۀ مقادیر ورودی در یک آرایه و کمک گرفتن از روش مرتب کردن حبابی (سؤال 24) حل کنید.

(تألیفی)

* برنامه‌ای بنویسید که n عدد از کاربر دریافت کرده و k امین بزرگترین عدد را مشخص کرده و چاپ کند. (k و n توسط کاربر وارد می‌شوند.)

(امتحان دانشگاه تهران، 22/8/1393)

* برنامه‌ای بنویسید که 21 عدد از کاربر دریافت کرده و عدد میانه‌، max و min این 21 عدد را مشخص کرده و چاپ کند.

**راهنمایی:** داده‌ها را مرتب کنید.

(تمرینات برنامه نویسی با زبان C++، آمادگی امتحان دانشگاه تهران)

* با استفاده از یک آرایۀ تک اندیسی، مسألۀ زیر را حل کنید:

20 عدد بین 10 و 100 را بخوانید (خود 10 و 100 نیز مجازند) و پس از هر بار خواندن، فقط اگر این عدد قبلاً خوانده نشده آن را چاپ کنید. این مسأله را برای بدترین حالت که در آن هر 20 عدد متفاوتند حل کنید. برای حل این مسأله از کوچکترین آرایۀ ممکن استفاده کنید.

(How to program, Deitel Deitel)

1. (غربال اِراتوستن) یک عدد صحیح اول عدد صحیحی است که فقط بر خود و بر 1 قابل تقسیم باشد. غربال اراتوستن روشی برای یافتن اعداد اول است. این روش به صورت زیر کار می‌کند.

الف) آرایه‌ای درنظر بگیرید و مقدار اولیۀ همۀ عناصرش را برابر با 1 (true) قرار دهید. درنهایت عناصر با اندیس اول، 1 باقی خواهند ماند ولی بقیۀ عناصر آرایه صفر خواهند شد.

ب) با شروع از اندیس 2 (اندیس 1 حتماً اول است.) هربار که عنصری با مقدار 1 یافت می‌شود در بقیۀ آرایه عناصری که اندیسشان مضربی از اندیس عنصر با مقدار 1 است صفر می‌شوند. برای اندیس 2، همۀ عناصر بعد از 2 که اندیسشان مضربی از 2 است (اندیسهای 4، 6، 8، 10، ...) برابر با صفر قرار داده می‌شوند؛ برای اندیس 3، همۀ عناصر بعد از 3 که اندیسشان مضربی از 3 است (اندیسهای 6، 9، 12، 15، ...) صفر قرار داده می‌شوند و به همین ترتیب الی آخر.

در پایان عملیات، عناصری از آرایه که هنوز یک هستند اندیسهایی را نشان می‌دهند که عدد اولند. برنامه‌ای بنویسید که با استفاده از یک آرایۀ 1000 عنصری، اعداد اول از 1 تا 999 را بدست آورد و چاپ کند. عنصر 0 آرایه را نادیده بگیرید.

(How to program, Deitel Deitel)

* برنامه‌ای بنویسید که عدد N را از کاربر دریافت کرده و سپس به تعداد N مختصات به شکل x و y از کاربر دریافت و ذخیره کند. سپس مساحت حاصل از اتصال به ترتیب این نقاط را بدست آورد.

**راهنمایی:** برای x ها و y ها دو آرایه‌ی جداگانه تعریف کنید. به مسئله اضافه بعد از سؤال 2 نگاه کنید.

(تألیفی)

1. برنامه‌ای بنویسید که از کاربر 20 عدد دریافت کرده و بزرگترین توالی مرتب صعودی موجود در این لیست اعداد را چاپ نماید.

**مثال:** درصورتی که کاربر اعداد زیر را وارد نماید:

3 , 6 , 23 , 87 , 45 , 50 , 63 , 71 , 90 , 103 , 139 , 76 , 30 , 31 , 34 , 56 , 79 , 80 , 21 , 25

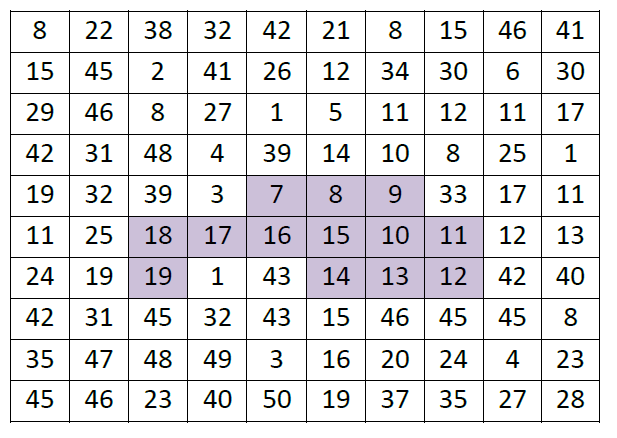
خروجی برنامه باید توالی زیر باشد زیرا طولانی ترین توالی صعودی موجود است:

45 , 50 , 63 , 71 , 90 , 103 , 139

(امتحان دانشگاه تهران، 22/8/1393)

1. جدولی با ابعاد 10 در 10 از اعداد صحیح داریم. اعداد به صورت تصادفی در این صد سلول قرار گرفته‌اند. (فعلاً برنامه را به نحوی بنویسید که کاربر اعداد را وارد کند زیرا هنوز تولید اعداد تصادفی مطرح نشده است.) تابعی بنویسید که این جدول را به عنوان ورودی دریافت کرده و طولانی ترین دنباله از اعداد سریالی را در آن جست و جو کرده و به همراه مکان آنها چاپ نماید. اعداد در چهار جهت اصلی به دنبال هم قرار می‌گیرند.

**مثال:** فرض کنید جدول به صورت زیر باشد، آنگاه طولانی‌ترین دنباله سریالی اعداد جدول از 7 تا 19 خواهد بود که با رنگ تیره مشخص شده است:



(امتحان دانشگاه تهران، 17/10/1394)

***3-2***

*یکی دیگر از کتابخانه‌های کاردبردی، کتابخانه‌ی iomanip است که حاوی توابعی برای قالب بندی می‌باشد. در این بخش، دو تابع از این کتابخانه را معرفی می‌کنیم:*

*1- تابع setw که فضای خاصی را از سمت چپ برای خروجی بعدی رزرو می‌کند. خروجی بعد نیز این فضای رزرو شده را از سمت راست پر می‌کند و درصورت اتمام فضای رزرو شده، تابع setw نادیده گرفته می‌شود. بهتر است روند کارکرد این تابع را با امتحان کردن آن مشاهده کنید. روند استفاده از این تابع مثل عبارت endl می‌باشد که درون دستور خروجی قرار می‌گیرد. مقدار طول فضای دلخواه (عرض میدان) را با یک عدد صحیح مثبت به عنوان ورودی تابع مشخص می‌کنیم. به مثال زیر دقت کنید:*

*cout << “My name is” << setw(12) << “Behzad”;*

*این دستور، مقدار 12 فضای خالی بعد از عبارت My name is را برای خروجی بعدی رزرو می‌کند. خروجی بعدی با طول 6 حرف، این فضای رزرو شده را از سمت راست پر می‌کند 6 فضای خالی باقی مانده، خالی می‌مانند. درواقع این دستور خروجی زیر را تولید می‌کند:*

*My name is Behzad*

*2- تابع setprecision که بر روی نحوه‌ی نمایش تمام خروجی‌های عددی* ***اعشاری*** *بعد از خود تأثیر می‌گذارد. این تابع که نحوه‌ی استفاده از آن مانند setw درون دستور خروجی است، یک ورودی عدد صحیح مثبت دریافت می‌کند که این مقدار، نشان دهنده‌ی تعداد ارقامی است که باید از مقادیر عددی اعشاری خروجی های بعدی نشان داده شود. اگر مقدار ورودی تابع از تعداد ارقام خروجی (هم قبل از اعشار و هم بعد از آن) بیشتر یا مساوی با آن باشد، این تابع کار خاصی انجام نمی‌دهد. اگر این مقدار از تعداد ارقام قبل از اعشار بیشتر و تفاوت این دو از تعداد ارقام بعد از اعشار کمتر باشد، یا اینکه این مقدار با تعداد ارقام قبل از اعشار برابر باشد، مقدار عددی هنگام نمایش، به نحوی گرد می‌شود که تعداد ارقام نمایشی با مقدار تابع برابر شود. اگر این مقدار از تعداد ارقام قبل از اعشار نیز کوچکتر باشد، باز هم مقدار عددی گرد شده و به صورت نمایش علمی، نمایش داده می‌شود. به سه مثال زیر توجه کنید:*

*cout<< setprecision(4)<< 12.2475 << endl<< 12.5 << endl<< 1257843.0;*

*خروجب به صورت زیر خواهد بود:*

*12.25*

*125*

*1.258e+006*

*توجه کنید که توابع دیگری نیز درقالب بندی کاربرد دارند که از حوصله‌ی این جزوه خارج اند و تحقیق درمورد آنها را به خودتان می‌سپاریم.*

* برنامه‌ی سوال 20 را با استفاده از دستور setw بازنویسی کنید.

(تألیفی)

* برنامه سؤال 27 را به نحوی بازنویسی کنید که نتایج را به شکل جدول‌واره‌ی زیر نمایش دهد:

C| x y N

و این سطر را برای هر نقطه تکرار کند. C شماره‌ی نقطه است (ردیف) و «|» صرفاً برای مرتب بودن جدول می‌باشد. x شماره ستون نقطه و y شماره سطر آن است و N نیز مقدار درون خانه‌ی مورد نظر. (فاصله‌ی بین پارامتر ها را با setw با عرض میدان 10 پیاده کنید.)

(تألیفی)

* سؤال 2 و همچنین سؤال اضافه‌ی پس از آن را به نحوی بازنویسی کنید که جواب را به ترتیب با سه رقم و دو رقم اعشار نمایش دهند.

(تألیفی)

***3-3***

*یک تابع کاربردی در کتابخانه‌ی استاندارد <stdlib.h> وجود دارد که ترجیح داده شده در این قسمت توضیح داده شود. خروجی این تابع یک عدد تصادفی بین 0 و 32767 است و ورودی آن هیچ پارامتری ندارد. برای استفاده از این مقدار تصادفی باید آن را در یک متغیر ذخیر کنید:*

*i = rand ( );*

*در طول اجرای برنامه، با هربار رسیدن به عبارت rand ( ) یک مقدار جدید تولید می‌شود. گاهی ما لازم داریم مقدار تولید شده در بازه‌ی خاصی قرار گیرد، در این صورت می‌توانیم از عملگر % استفاده کنیم:*

*i = rand ( ) % 36;*

*دستور بالا، عددی تصادفی بین 1 تا 35 تولید می‌کند. همینطور می‌توانیم برای تولید اعداد منفی، یک -1 در مقدار حاصل ضرب کنیم و به همین صورت، با دیگر اعمال ریاضی می‌توانیم اعدادی تصادفی به صورت دلخواه تولید کنیم.*

*نکته‌ای وجود دارد که باید به آن دقت شود؛ تابع rand درواقع اعداد شبه تصادفی تولید می‌کند. فراخوانی مکرر rand دنباله‌ای تولید می‌کند که تصادفی بنظر می‌رسد اما این دنباله هربار که برنامه اجرا می‌شود عیناً تکرار می‌شود. این خصوصیت به اشکال زدایی برنامه کمک فراوانی می‌کند زیرا بررسی روند کار برنامه را تسهیل می‌بخشد. درنهایت، پس از اطمینان از کارکرد درست برنامه، می‌توان این ایراد را برطرف کرد به طوری که اعداد کاملاً تصادفی باشند. این کار بوسیله‌ی تابعی دیگر بنام srand در کتابخانه‌ی استاندارد صورت می‌گیرد. تابع srand یک مقدار صحیح از نوع unsigned به عنوان ورودی دریافت می‌کند و باعث می‌شود توابع rand به کار رفته در برنامه در هربار اجرای برنامه مقادیر کاملاً تصادفی تولید کنند. توجه کنید که باز هم دنباله‌ی تصادفی حاصل به مقدار ورودی تابع srand بستگی دارد پس مقدار این ورودی باید به نحوی وارد شود که در هربار اجرای برنامه مقداری متفاوت داشته باشد. نحوه استفاده از srand در زیر آمده است. این یک مثال ساده است:*

*#include <iostream>*

*#include <stdlib.h>*

*using namespace std;*

*int main(){*

*unsigned seed;*

*cin >> seed;*

*srand (seed);*

*for (int i=0; i<10; i++)*

*cout << rand ( ) << endl;*

*return 0;*

*}*

1. برنامه‌ای بنویسید که ریختن دو تاس را شبیه‌سازی کند. این برنامه باید برای ریختن اولین تاس از rand و برای ریختن تاس دوم نیز بار دیگر از rand استفاده کند. سپس مجموع این دو مقدار محاسبه شود.

**توجه:** چون هر یک از تاس‌ها مقدار صحیحی بین 1 تا 6 را نشان می‌دهد پس مجموع این دو مقدار عددی بین 2 تا 12 است که مجموع 7 بیشترین دفعات و مجموع 2 و 12 کمترین دفعات پیش می‌آیند. برنامۀ شما باید دو تاس را 36000 بار بریزد. با استفاده از یک آرایۀ تک اندیسی، تعداد دفعاتی را که هر یک از مجموع‌های ممکن، پیش می‌آیند ثبت و نتایج را به شکل جدولی چاپ کنید. سپس تحقیق کنید که آیا تعداد مجموع‌های حاصله، معقول است یا خیر (مثلاً به شش طریق، مجموع 7 حاصل می‌شود پس تقریباً یک ششم همۀ پیشامد ها باید 7 باشد).

**نکته:** نیازی به استفاده از srand نیست. برنامه را با rand پیاده سازی کنید.

(How to program, Deitel Deitel)

1. برنامه‌ای بنویسید که جدول سؤال 27 را با اعداد تصادفی پر کند و جدول را نشان دهد. از دستور srand استفاده کنید و مقدار اولیه‌ی srand را در ابتدای برنامه دریافت کنید. توجه داشته باشید که درصورت وارد شدن اعداد مشابه، جداول مشابهی دریافت می‌شود. نیازی به ادامه‌ی حل آن سوال نیست و ساخت جدول کفایت می‌کند. (عداد یک یا دو رقمی باشند.)

(تألیفی)

1. برنامه‌ای بنویسید که N را از کاربر دریافت کند، به تعداد N عدد تصادفی تولید کرده و نمایش دهد و مجموع حاصله را از کاربر بخواهد. امتیازی برای کاربر بوجود آورید که درصورت پاسخ درست، N مقدار به آن اضافه شود و درصورت پاسخ غلط، 5 امتیاز از آن کسر گردد. به برنامه خاصیت تکرار بدهید و درصورت وارد شدن عدد 0، برنامه را پایان دهید. قسمت تکرار برنامه باید شامل دریاف N نیز باشد.

برای تولید اعداد تصادفی از srand استفاده کنید و برای مقدار اولیه‌ی آن، از ساعت کامپیوتر کمک بگیرید. در این صورت برنامه در هر زمانی که اجرا شود، با توجه به لحظه‌ی اجرا، اعداد متفاوتی را ارائه می‌کند. این یک روش عالی برای تولید اعداد تصادفی است. (اعداد فقط 2 یا 3 رقمی باشند.)

**گامی رو به جلو:**  برای به دست آوردن ساعت کامپیوتر به طریق زیر عمل کنید:

time ( 0 );

این عبارت زمان کامپایل شدن برنامه را برمی‌گرداند. این تابع در کتابخانه‌ی <ctime> قرار دارد.

(تألیفی)

1. برنامه‌ای بنویسید که مسئله‌ی هشت وزیر شطرنج را حل کند.

**راهنمایی:** مسئله‌ی هشت وزیر یعنی که هشت مهره را در صفحه‌ی 8 در 8 شطرنج طوری قرار دهید که هیچکدام از مهره ها با حرکت طولی، عرضی یا اوریب با هر طولی، به یکدیگر نرسند.

(تألیفی)

***فصل چهارم***

***StringS***

* **مسائل این قسمت مربوط به بخش Arrays و همینطور File I/O جزوۀ استاد کریمیان می‌باشند و به ترتیب آسان به سخت و همینطور ترتیب مباحث جزوه مرتب شده اند.**
* **5 کادر آموزشی**
* **5 سوال با کد پاسخ**
* **9 سوال بدون کد پاسخ**

***4-1***

*یک راه قدیمی و ابتدایی برای ذخیره‌ی رشته‌ها، ذخیره‌ی آنها، کاراکتر به کاراکتر، درون یک آرایه از کاراکترهاست. توجه کنید که یک رشته‌ی 5 کاراکتری به آرایه‌ای 6 عضوی نیاز دارد، زیرا باید یک عضو نهایی برای کاراکتر ‘\0’ رزرو شود. اگر مقادیر آرایه را در کد برنامه به آن می‌دهید، درصورت عدم رعایت قوانین مربوط به اندازه‌ی آرایه با خطای کامپایلر مواجه می‌شوید، اما اگر به عنوان مثال با کدی مانند دستور زیر، مقدار ورودی را در طول اجرای برنامه دریافت کنید و تعداد کاراکترها از اندازه‌ی آرایه بزرگتر باشد، با ایندکس بیرون از حدود آرایه مواجه خواهیم شد که دراین صورت در برخی موارد، خود برنامه اندازه‌ی آرایه را تا حد مورد نیاز افزایش می‌دهد. اما این کار امن نیست و ممکن است جزئی از اطلاعات از دست بروند. سعی کنید با انتخاب اندازه‌ای به اندازه‌ی کافی بزرگ، از رخ دادن چنین رویدادی جلوگیری کنید:*

*char C[4];*

*cin >> C;*

*بدین صورت می‌توان با نوشتن نام آرایه (بدون اندیس) به کل رشته دست پیدا کرد یا اینکه با استفاده از اندیس آن به تک تک کاراکترها دسترسی داشت. می‌توان برای کنترل ایندکس بیرون از حدود آرایه، برای دریافت کاراکترها از حلقه‌ای مانند زیر استفاده کرد:*

*char C[12];*

*for(int i=0; i<11; i++)*

*cin >> C[ i ];*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*راه کاربردی‌تر برای استفاده از رشته‌ها، استفاده از نوع string می‌باشد. به سه خط کد زیر که یک string به نام str را اعلان و مقداردهی می‌کند، سپس مقدار را از طرف ورودی در آن ذخیره می‌کند و درنهایت آن را چاپ می‌کند دقت کنید. توضیحات بیشتر درمورد نکات این نوع داده در آینده داده خواهد شد.*

*string str = “Behzad”;*

*cin >> str;*

*cout << srt;*

*یکی از مزیت‌های استفاده از این نوع داده بجای آرایه‌ی کاراکتری، نامحدود بودن رشته‌ی ورودیست.*

***نکته)*** *توجه کنید که با دریافت رشته‌ها از طریق cin نمی‌توان کاراکتر‌های فضای سفید مانند خط جدید یا فاصله را دریافت کرد(حتی در نوع داده‌ی string). البته اگر این کاراکترها را هنگام مقداردهی زمان کامپایل وارد کنیم مشکلی نخواهد بود اما هنگام دریافت زمان اجرا، به مشکل خواهیم خورد. روش‌های حل این مشکل را در کادرهای بعدی توضیح خواهیم داد.*

* برنامه‌ای بنویسید که برای 10 نفر، فضای کاربری تهی و آماده برای پرشدن ایجاد کند. این فضاها باید شامل اطلاعات زیر باشند.

1- نام 2- نام خانوادگی 3- شهر محل تولد 4- سن 5- تاریخ تولد 6- شغل

7- جنسیت 8- وضعیت تأهل 9- کد حساب 10- وضعیت حساب

موارد 9 و 4 را با نوع عدد صحیح، موارد 7 و 8 را با نوع enum، مورد 10 (از نظر اشغال یا آزاد بودن) از نوع bool و دیگر موارد را به صورت آرایه‌ای از کاراکترها دریافت کنید. فرض کنید هیچ کدام از این موارد بیشتر از 10 حرف نیستند و به دریافت فاصله نیازی نیست. هر حساب باید یک کد و یک رمز عبور داشته باشد. کدها برای حساب‌ها باید به صورت تصادفی داده شوند (نیازی به محدود کردن بازه اعداد تصادفی نیست و همینطور اعداد شبه تصادفی کفایت می‌کنند.). کاربر با وارد کردن درخواست ثبت نام، باید به مرحله‌ی ثبت نام رسیده و پس از پایان، کد حساب را دریافت و یک رمز عبور انتخاب کند. اگر حساب کد وارد شده پر بود، باید رمز عبور وارد کند و درصورت صحیح بودن، اطلاعات را مشاهده کند.

چون هنوز با ذخیره اطلاعات آشنا نیستید، برنامه را به نحوی بنویسید که پس از پایان کار برای هر کاربر (حتی کسی که رمز اشتباه وارد کرده) برنامه خاتمه نیابد و به مرحله دریافت کد برگردد.

(تألیفی)

1. برنامه‌ای بنویسید که پیوسته، یک رشته نامحدود از کاربر دریافت کرده و در یک آرایه‌ی کاراکتری ذخیره کند. نیازی به دریافت فاصله‌ها نیست و می‌توانید آن‌ها را نادیده بگیرید. توجه کنید که شما با «وکتور» آشنایی ندارید پس منظور سؤال یافتن راه‌حلی برای ذخیره رشته، **بدون محدودیت فضا** است. (برنامه را با خواندن مقدار ‘0’ پایان دهید.)

**راهنمایی:** به دو آرایه نیاز دارید.

**نکته:** کد پاسخ این سؤال به صورت مطلوب نوشته نشده است اما نگاهی گذرا و ساده است به روند کار وکتور برای تولید آرایه‌ی نامحدود. (روند کار وکتور پیچیده‌تر و کاربردی‌تر می‌باشد اما از همین تکنیک کلی که در پاسخ این سؤال از آن استفاده شده است استفاده می‌کند.)

(تألیفی)

* برنامه‌ی سؤال اضافه‌ی قبل از سؤال 32 را با استفاده از نوع string بجای آرایه‌ی کاراکتری بازنویسی کنید. دیگر نیازی به محدود کردن رشته‌ها نیست.

(تألیفی)

* برنامۀ سؤال 32 را برای دریافت تعداد نامحدودی کلمه تکرار کنید. کلمات را با cin دریافت کرده و در یک آرایه از نوع string ذخیره کنید. دقت کنید که باید همانند سؤال 31، یک آرایه‌ی بینهایت بسازید. دقت کنید که با دریافت با استفاده از cin، فاصله ها ذخیره نمی‌شوند و دقیقاً با دیدن هر فاصله، عبارت به عنوان یک کلمه مجزا در یک عضو از آرایه ذخیره می‌شود. پس اگر کاربر متنی را پیوسته وارد کند، برنامه خود به خود آن را به کلمات مجزا تقسیم کرده و ذخیره می‌کند. پس از دریافت عدد 0، به دریافت خاتمه دهید و عبارت ورودی را چاپ کنید. با چاپ فاصله بین کلمات، متن را بازسازی کنید.

(تألیفی)

***4-2***

*گفته شد یکی از ایرادهای این نحوه‌ی دریافت رشته (از طریق دستور cin)، این است که نمی‌توان کاراکترهای فضای سفید، مثل فاصله یا خط جدید را ذخیر کرد. برای رفع این مشکل می‌توان از تابع get استفاده کرد:*

*cin.get(C[0]);*

*این دستور، یک کاراکتر از ورودی دریافت و در عضو مشخصی از آرایه یا در متغیری از نوع کاراکتر ذخیره می‌کند. نکته‌ی مثبت این دستور توانایی دریافت کاراکترهای سفید است. برای دریافت تعداد زیادی کاراکتر باید از حلقه‌ها استفاده کرد (برای نمونه، با اینکه این روش سخت حل مسئله است، سؤال 33 به اینصورت حل شده است.) اما راه ساده‌تری نیز وجود دارد. این تابع توانایی دریافت آرگومانی برای تعداد کاراکترهای دریافتی دارد:*

*char C[20];*

*cin.get(C , 10);*

*همچنین می‌تواند آرگومان سومی از نوع کاراکتر به آن داد تا به محض رسیدن به آن، به دریافت خاتمه دهد:*

*cin.get(C , 10 , ‘.’ );*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*یک تابع کاربردی، تابع ignore برای صرف نظر از ورودی می‌باشد.*

*cin.ignore(20 , ‘c’)*

*این تابع دو آرگومان به عنوان ورودی دریافت می‌کند. اولین آرگومان یک مقدار عددی صحیح مثبت است که تعداد کاراکتر‌هایی را نشان می‌دهد که باید از آنها صرف‌نظر شود. این دستور برنامه را برای دریافت و صرف‌نظر از 20 کاراکتر متوقف می‌کند. توجه کنید که کاراکتر خط جدید و فاصله نیز محسوب هستند. آرگومان دوم یک کاراکتر است که برنامه با دریافت آن، از آن صرف‌نظر کرده و به صرف‌نظر کردن خاتمه می‌دهد. معمولاً کاراکتر ‘\n’ (سطر جدید) را به عنوان کاراکتر خاتمه قرار می‌دهند تا با وارد کردن هر عبارتی و فشردن کلید enter ، به صرف‌نظر کردن خاتمه داده شود.*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*تابع getch( ) تابعی به شدت پرکاربرد از کتابخانه‌ی coino.h است که صلاح دیده شده در این بخش توضیح داده شود. این تابع که آرگومانی دریافت نمی‌کند، برای دریافت* ***یک*** *کاراکتر از جریان ورودی ساخته شده است. نکته‌ی متمایز کننده‌ی این تابع این است که برای وارد کردن کاراکتر، نیازی به فشردن کلید enter نیست. درواقع بلافاصله پس از فشردن هر کلید، آن را دریافت می‌کند. این تابع برای ایجاد یک «ایست» در برنامه استفاده می‌شود تا بتوان اطلاعات برنامه را با دقت از روی کنسول خواند. اگر بخواهیم کاراکتر ورودی را در یک متغیر ذخیره کنیم، باید دستور را به شکل زیر بنویسیم:*

*C = getch();*

*و بدین صورت کاراکتر ورودی در متغیر C ذخیره می‌شود. چند مثال از کاربردهای این تابع را در مسائل زیر مشاهده می‌کنید.*

1. برنامه سؤال اضافه قبل از سؤال 32 را به نحوی بازنویسی کنید که داده‌های از نوع آرایه‌ی کاراکتری قابلیت دریافت فاصله را نیز دارا باشند. (از تابع get استفاده کنید.)

در همان سؤال، برای دریافت تاریخ تولد، فرض کنید کاربر به یکی از دو شکل زیر تاریخ را وارد می‌کند:

1- چهار رقم سال، علامت «/»، 2 رقم ماه (مثلاً 04)، علامت «/»، دو رقم روز

2- دو رقم ماه، علامت «/»، دو رقم روز

برنامه را به نحوی گسترش دهید که درصورت وارد شدن تاریخ، به هر طریق دیگری، پیغام خطا چاپ کرده و دوباره از کاربر درخواست وارد کردن تاریخ کند. (نیازی نیست ورود حرف بجای عدد را بررسی کنید.) درصورت وارد شدن تاریخ به صورت اول، از 4 رقم اول سال و همینطور علامت «/» مربوطه صرف‌نظر کنید.

(تألیفی)

* پاسخ برنامه‌ی سؤال 33 را به نحوی اصلاح کنید که بجای دریافت حروف به صورت یکی یکی با استفاده از حلقه‌ها، از تابع cin.get با دو آرگومان استفاده کند.

**راهنمایی:** برای اینکار، لازم است جای دو اندیس آرایه را جابجا کنید. یعنی اول شماره‌ی کاربر قرار گیرد و سپس تعداد کاراکترهای ورودی.

**نکته:** با انجام تکنیک بالا (راهنمایی) برای نمایش عبارات نیز دیگر نیازی به حلقه‌ها نیست. کافیست نام آرایه را به همراه اندیس اول و بدون ذکر اندیس دوم در دستور خروجی قرار دهید.

درواقع از هر نظر، منطقی‌تر بود که کد این سؤال به این صورتی که ذکر شد نوشته شود اما صرفاً برای نشان دادن چگونگی استفاده از حلقه‌ها در این مسائل، کد موردنظر به این صورت نوشته شد.

(تألیفی)

* برنامه‌ی سؤال 27 را به نحوی تکمیل کنید که برای نمایش نتیجه‌ها، منتظر فشار دادن یک کلید (هر کلیدی) بماند و با وارد شدن هربار یک کلید، یکی از پاسخ‌ها را چاپ کند.

(تألیفی)

1. برنامه‌ای بنویسید که با شروع از عدد 2، با هربار فشار دادن کلید N (n یا N)، عدد اول بعدی را چاپ کند و با فشار دادن کلید E (e یا E)، از برنامه خارج شود.

(تألیفی)

* برنامه سؤال 32 را به نحوی تکمیل کنید که منتظر وارد شدن کلید enter برای ذخیره حروف نماند و همینطور فاصله را نیز دریافت کند.

(تألیفی)

1. برنامه‌ای بنویسید که «\*» را در صفحه حرکت دهد. اینگونه عمل کنید که در ابتدا، یک ستاره در نقطه شروع برنامه چاپ شود، سپس با هربار وارد کردن کلید «D»، این ستاره با یک فاصله (یعنی یک خانه به راست) چاپ شود (از setw استفاده کنید.) و با ورود کلید «A»، یک خانه به سمت چپ برود. درصورت قرار گرفتن ستاره در چپ ترین نقطه، حرکت آن را به سمت چپ متوقف کنید. همین کار را برای کلیدهای «W» و «S» برای بالا و پایین (به وسیله endl) تکرار کنید.

**راهنمایی:** هربار باید با ورود کلید جدید، صفحه پاک شده و از اول رسم شود.

**گامی رو به جلو:** برای پاک کردن صفحه از تابع clean استفاده کنید.

برای استفاده از این تابع کافی است هرجایی از کد که می‌خواهید کنسول تماماً خالی شود، عبارت زیر را به عنوان دستور قرار دهید.

system("cls");

(تألیفی)

***4-3***

*گفته شد یک نوع ساده‌تر برای رشته‌ها نوع string است. یک تابع بسیار ساده و پرکاربرد برای دریافت این نوع وجود دارد. تابع getline تابعیست که دو آرگومان دریافت می‌کند و یک* ***سطر*** *داده از نوع string را در متغیری معلوم ذخیره می‌کند. آرگومان دوم نام متغیری است که می‌خواهیم یک سطر را در آن ذخیره کنیم اما متغیر اول، نحوه‌ی دریافت این یک سطر را مشخص می‌کند. فعلاً چون ما فقط با دستور cin اطلاعات را از صفحه کلید دریافت می‌کردیم، برای آرگومان اول این تابع، فقط عبارت cin را می‌شناسیم. پس کاربرد تابع به شکل زیر می‌باشد:*

*string str;*

*getline(cin,str);*

***نکته)*** *نکته‌ی شایان توجه این است که این دستور، داده‌ها را تا زمانی دریافت می‌کند که کاراکتر ‘\n’ (enter بر روی کیبورد) وارد نشده باشد و این کاراکتر آخرین کاراکتر دریافتی در این دستور است. اما دستور cin (حتی برای دریافت داده‌های عددی یا ...) بدون دریافت این کاراکتر خاتمه می‌یابد پس اگر قبل از دستور getline از دستور cin استفاده شده باشد، یک کاراکتر نوسطر در ورودی باقی می‌ماند که توسط اولین دستور getline دریافت می‌شود و دستور مذکور خاتمه می‌یابد! برای رفع این مشکل کافیست اولین دستور getline پس از cin را دو بار اجرا کنید یا اینکه حرفه‌ای تر عمل کرده و کاراکتر مذکور را با cin.ignore حذف کنید.*

* با استفاده از دستور getline() و استفاده از string ها منظور سؤال 32 را با اضافه کردن قابلیت دریافت فاصله برآورده کنید. هدف این سؤال نمایان کردن برتری string نسبت به آرایه‌ی کاراکتری و همچنین دستور دریافت getline نسبت به دیگر دستورهاست. همانطور که می‌بینید، کار با آن بسیار ساده تر است.

(تألیفی)

***4-4***

*یک دستور کاربردی که شاید در سؤال 33 می‌توانست بسیار مفید باشد، دستور typedef می‌باشد. با یک مثال این دستور را شرح خواهیم داد:*

*typedef char C\_Str[10];*

*این دستور یک نوع داده‌ی جدید به نام C\_Str خلق می‌کند. هر متغیر جدیدی که از این نوع تعریف شود، درواقع یک آرایه‌ی کاراکتری 10 عضوی می‌باشد:*

*C\_Str name;*

*می‌توان آرایه‌ای از این نوع تعریف کرد:*

*C\_Str name[15];*

*این عبارت 15 متغیر از نوع رشته‌های کاراکتری 10 عضوی اعلان می‌کند.*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*یک دستور دریافت دیگر برای دریافت آرایه‌ی کاراکتری نیز وجود دارد:*

*char A[10];*

*cin.getline(A , 5);*

*این دستور معادل دستور زیر است:*

*cin.get(A , 5 , ‘\n’);*

* برنامه‌ی سوال اضافه‌ی قبل از سوال 32 را به طوری بازنویسی کنید که با استفاده از cin.getline فاصله‌ها را هم دریافت کند و در آرایه‌های کاراکتری ذخیره کند. اول با استفاده از typedef یک نوع آرایه‌ی کاراکتری 10 عضوی تعریف کنید و سپس فضاهای رشته‌های مورد نیاز را با استفاده از آن (نه به صورت آرایه‌ای) تعریف کنید.

(تألیفی)

***4-5***

*در این کادر آموزشی قصد معرفی کتابخانه‌ای کاربردی را داریم که بر داده‌هایی از نوع string عمل‌های گوناگونی انجام می‌دهد. کتابخانه‌ی <string> حاوی توابعی بسیار کاربردی در کار با رشته‌ها است. قسمی از این توابع را صرفاً با آوردن مثالی از آن‌ها و کاری که انجام می‌دهند معرفی می‌کنیم. قبلاً با اعلان string ها آشنا شدید:*

*string s1;*

*s1 = “country!”;*

*string s2 = “city!”*

*معنای این خطوط برای شما آشناست.*

*string s3(12 , ‘\*’)*

*رشته‌ی s3 حاوی 12 کاراکتر ‘\*’ اعلان می‌شود.*

*string s4 = s3;*

*رشته‌ی s4 با محتویات s3 اعلان می‌شود. درواقع یک کپی ایجاد می‌شود.*

*string s5(s1 , 3 , 2)*

*رشته‌ی s5 اعلان و به این صورت مقداردهی می‌شود: کاراکترهای s1 به ترتیب با شروع از 0 شماره گذاری می‌شوند و از کاراکتر سوم، به تعداد 2 کاراکتر در s5 ذخیر می‌شود. درواقع رشته‌ی s5 برابر “nt” خواهد بود.*

*نحوه‌ی دریافت string ها قبلاً توضیح داده شد (تابع getline) پس نیازی به تکرار نیست.*

*char C = s1[2];*

*با شماره گذاری از 0، کاراکتر شماره 2 از رشته‌ی s1 (یعنی ‘u’) را در C جایگذاری می‌کند. درواقع با اندیس‌ها می‌توان به تک تک کاراکترهای یک رشته دسترسی پیدا کرد یا آنهارا جایگزین کرد:*

*s1[3] = ‘g’;*

*cout << s1.length( );*

*تعداد کاراکترهای درون رشته‌ی s1 چاپ می‌شود. درواقع تابع length تابعی برای استخراج طول رشته است.*

*s4 = = s3;*

*تساوی دو رشته را (کاراکتر به کاراکتر) بررسی می‌کند.*

*همچنین می‌توان از عملگر‌های مقایسه‌ای برای رشته‌ها استفاده کرد:*

*s2 > s1; s4 <= s3;*

*این مقایسه‌ها به این صورت انجام می‌شوند که مقدار عددی کاراکتر اول هر رشته با هم مقایسه می‌شوند (مقدار عددی، همان کد کاراکتر که می‌توانید با دستور تغییر نوع کاراکتر به int آن‌ها را بدست آورید.)، اگر برابر نبودند، مقایسه‌ی همین دومقدار بجای مقایسه‌ی کل انجام می‌شود اما اگر برابر بودند مقایسه به سراغ کاراکتر‌های دوم و به همین صورت الی‌آخر می‌رود. اگر یکی از رشته‌ها زودتر تمام شد، رشته‌ای که ادامه دارد بزرگتر شناخته می‌شود. تساوی دو رشته نیز کاملاً روشن است.*

*string s6 = s1 + s2 + “-done”;*

*رشته‌ها را به هم پیوند می‌زند. مقدار s6 برابر “country!city!-done” خواهد بود. همچنین عملگر += نیز قابل استفاده است.*

*دو دستور زیر به یک ‌معنا هستند:*

*string s7= s6.substr(3 , 4);*

*string s7(s6 , 3 , 4)*

*درواقع، تابع substr() تابعی برای تولید زیررشته است. مزیت آن، استفاده بدون ذخیره در متغیری دیگر است.*

*s6.erase(7 , 5);*

*در رشته‌ی s6 از کاراکتر شماره‌ی 7، تعداد 5 کاراکتر را حذف می‌کند. درواقع عبارت “!city” حذف می‌شود و s6 برابر “country!-done” باقی می‌ماند.*

*s6.replace(7, 2 , “ is for ”)*

*از کاراکتر شماره هفت، دو کاراکتر حذف و بجای آن رشته‌ی نوشته شده جای‌گذاری می‌شود. درواقع s6 برابر “country is for done” خواهد بود.*

*s6.find(“or”);*

*عبارت وارد شده را در درون رشته جستجو می‌کند. اگر یافت نشد، مقداری پرت برمی‌گرداند. اگر پیدا شد، شماره‌ی اولین کاراکتر مربوطه در‌رشته را (در اینجا 12) برمی‌گرداند. می‌توان به این تابع آرگومان دومی از نوع int نیز داد که این بررسی را از آن شماره کاراکتر به بعد انجام دهد.*

*توجه کنید توابع دیگری نیز در این کتابخانه وجود دارند که در اینجا مطرح نمی‌شوند.*

* برنامه‌ی سؤال 36 را بدون رعایت نکته‌ی ذکر شده در سؤال بازنویسی کنید. درواقع نیازی به استفاده از تابع find نیست و شما با دانستن رشته‌ی اصلی، می‌توانید شماره‌ی کاراکتر‌های مورد نظر را با دست بدست آورید و دربرنامه، از شکل اندیس‌دار رشته استفاده کنید. رعایت نکته‌ی مذکور، سطح سؤال را تا حدی بالا می‌برد.

(تألیفی)

1. برنامه‌ای بنویسید که رشته‌ی زیر را اعلان کرده و سپس اعمال گفته شده را به ترتیب بر آن انجام دهد.

“Ali\tZahra\tRashid\tZohre\tShahla\tNima\tReza\tSohrab\tNahid\n”

الف) این رشته را چاپ کند.

ب) نام «رضا» را به همراه فاصله‌ی بعد از آن حذف کند و رشته‌ی جدید را چاپ کند.

پ) کاراکتر پایان سطر را حذف کرده، نام «لاله» را به همراه فاصله‌ی قبل و کاراکتر نوسطر بعد، آخر آن اضافه کند.

ت) نام «شهلا» را با نام «شهره» تعویض کند و رشته‌ی حاصل را چاپ کند.

ث) این 9 نام جدید را در 9 رشته‌ی مجزا درون یک آرایه ذخیره کرده و آن‌ها را به صورت ستونی چاپ کند.

ج) این رشته‌های جدید را برحسب حروف الفبا درون آرایه مرتب کنید و دوباره به همین صورت چاپ کنید.

در هر مرحله بعد از چاپ، رشته‌ای حاوی 60 کاراکتر ‘\_’ چاپ کنید.

**نکته:** برای یافتن نام‌های مذکور درون رشته‌ از تابع مربوطه استفاده کنید. تعداد کاراکترها را به صورت دستی نشمارید. حتی شمردن تعداد کاراکتر‌های هر نام نیز مجاز نیست.

**راهنمایی:** برای حل مشکلات حاصل از نکته‌ی بالا، از شماره‌ی کاراکتر‌های ‘\t’ استفاده کنید.

(تألیفی)

***فصل پنجم***

***FunctionS***

* **مسائل این قسمت مربوط به بخش Functions جزوۀ استاد کریمیان می‌باشند و به ترتیب آسان به سخت و همینطور ترتیب مباحث جزوه مرتب شده اند.**
* **4 کادر آموزشی**
* **6 سوال با کد پاسخ**
* **5 سوال بدون کد پاسخ**

***5-1***

*در C++ برخی اعمال ریاضی و... گاهی لازم است که چندین بار تکرار شوند.  
ار آنجا که نوشتن هر بار این اعمال کار غیر منطقی و باعث افزایش زمان اجرای برنامه و شلوغی بیش از اندازه محیط کد نویسی میشود، می توان عملیات مورد نظر را در قالب یک تابع تعریف کرده و هر بار در صورت نیاز تابع را فرا خوانی کنیم.*

*نحوه تعریف کردن تابع:*

*function type function name (variables)*

*{*

*Operations…;*

*Return value;*

*}*

*function type: نوع بازگشتی تابع را مشخص میکند.  
function name نام تابع مورد نظر ماست به صورت دلخواه تعریف میشود  
operations عملیات، فرمان ها، شرط ها و ... هستند مه برای اجرا شدن مدنظر داریم.  
return value یا مقدار بازگشتی تابع که تابع، پس از انجام عملیات به برنامه باز میگرداند.*

*انواع تابع(انواع بازگشتی):*

*int  
single  
 خروجی عددی   
float  
double*

*char و string خروجی کارکتر و رشته*

*bool خروجی مقدار منطقی (true or false)*

*void مقدار بازگشتی ندارد*

*مثال: در زیر کد یک تابع به نام max() را مشاهده می کنید. این تابع دو پارامتر به ترتیب به نام های num1 و num2 (به عنوان ورودی) دریافت می کند و مقداری که از دیگری بزرگتر است را به عنوان خروجی برمی گرداند.*

*// function returning the max between two numbers*

*int max (int num1, int num2) {*

*// local variable declaration*

*int result;*

*if (num1 > num2)*

*result = num1;*

*else*

*result = num2;*

*return result;*

*}*

*مثال: تابع زیر عدد x را از کاربر گرفته به توان 3 میرساند و به برنامه بر میگرداند.*

*#include<iostream>  
int mokaab (int a) {  
 a=a\*a\*a;  
return a;*

*}*

*int main() {  
 int x;  
 cin>>x;  
 cout<<mokaab(x);  
 return 0;*

*}*

*توضیح: تابع مکعب (توان 3) را به نحوی تعریف میکنیم که با گرفتن عدد،آن را به توان 3 رسانده و مقدار نهایی را به برنامه باز میگرداند.*

*در برنامه با دریافت عدد از کاربر، عدد به تابع مکعب فرستاده شده، به توان میرسد، نتیجه بازگردانده شده به چاپ خواهد رسید.*

1. *برنامه ای بنویسید که با دریافت یک رشته و یک کاراکتر، تعداد تکرار کاراکتر در رشته را برگرداند. برای عمل شمارش از تابع استفاده کنید.*

*(تألیفی)*

1. *تابعی بنویسید که مقدار فاکتوریل یک عدد را حساب کند. این تابع را یک بار در main صدا بزنید.*

*(تألیفی)*

* *سوال های 10و 11و 13و 14و 15 همین جزوه را با استفاده از توابع حل کنید. برای هر سوال، تابعی که نوشتید را در main در یک حلقه ی do while قرار دهید تا برنامه روند تکرار پذیر پیدا کند. (فقط عمل محاسبات را به درون توابع ببرید.)*

*(تألیفی)*

1. *برنامه ای بنویسید که با دریافت دو عدد از کاربر، تمام اعداد اول بین آن دو را چاپ کند.*

*(تألیفی)*

* *برنامه ای برای محاسبه ی دترمینان ماتریس 2 در 2، 3 در 3 و 4 در 4 بنویسید. ابتدا یک تابع برای محاسبه ی دترمینان ماتریس 2 در 2 بنویسید و برای محاسبه ی دترمینان ماتریس 3 در 3 تابعی بنویسید که از تابع قبلی استفاده کند. همچنین برای محاسبه ی دترمینان ماتریس 4 در 4 نیز تابعی بنویسید که از تابع دترمینان 3 در 3 استفاده کند.*

*(تألیفی)*

* *با استفاده از سوال قبل، 4 معادله و 4 مجهول زیر را حل کنید و به صورت ماتریس نمایش دهید.*

*10x + 3y + 5z + 3w = 4*

*2x + 6z + w = 3*

*3x + 5y + 2w = 6*

*3y + 2z + w = 1*

*جواب:y = -74/11, z = -27/11, w = 287/11 x = -46/11,*

*(تألیفی)*

***5-2***

*انواع شیوه ارسال اطلاعات به توابع:*

|  |  |
| --- | --- |
| *در روش حاضر کپی از مقدار حقیقی یک آرگومان در پارامتر رسمی یک تابع جایگذاری شده و به پارامتر فرستاده می شود. در این روش، تغییراتی که بر روی پارامتر موجود در بدنه ی تابع اعمال می شود، در واقع هیچ تاثیری بر روی مقدار اصلی و اولیه ی آرگومان نخواهد داشت. زمانی بکار می آید که بخواهید تابع یا هیچ خروجی نداشته باشد و یا حداکثر 1 مقدار را برگرداند.* | *By value*  *مقدار* |
| *در این روش فراخوانی، آدرس آرگومان در پارامتر رسمی تابع جایگذاری شده یا به عبارت دیگر آدرس آرگومان ها به پارامتر ارسال می شود. داخل بدنه ی تابع، اشاره گر به منظور دسترسی به مقدار حقیقی آرگومان مورد استفاده قرار می گیرد. در روش حاضر هر تغییری که بر روی پارامتر اعمال می شود نهایتا بر روی اصل آرگومان نیز منعکس می گردد و مقدار نهایی آن را تغییر می دهد. این روش بیشتر زمانی کاربرد دارد که بخواهید بیش از یک مقدار خروجی تابع باشد.* | *By reference*  *ارجاع* |
| *برای اطلاع بیشتر:*  *در این روش فراخوانی، کپی آدرس یا نشانی آرگومان در حافظه در جایگاه پارامتر رسمی جایگذاری می شود (به پارامتر انتقال می یابد). داخل بدنه ی تابع، این آدرس جهت دسترسی به مقدار حقیقی آرگومانی که در تابع بکار رفته، مورد استفاده قرار می گیرد. با توجه به آنچه گفته شد، تغییراتی که بر روی پارامتر اعمال می شود، متعاقبا بر روی آرگومان نیز منعکس می گردد و نتیجتا مقدار اصلی آرگومان در حافظه را تغییر می دهد.* | *By pointer*  *پوینتر* |

***5-3***

*ارسال آرایه به عنوان پارامتر برای تابع:*

*از آرایه ها هم می توان در پارامترهای توابع استفاده نمود. برای اینکار می توان آرایه را با طول ارسال نمود و یا آرایه را بدون طول ارسال کرد و با یک پارامتر دیگری طول را تعیین و ارسال نمود.*

*مثال:*

*int minFunc(int arr[], int length) //------> تعریف تابع با تمامی متعلقات  
{*

*for (int count = 0; count < length; count++) {*

*cout << "Enter number [" << count+1 << "] :";*

*cin >> arr[cout];*

*}*

*int minNum = arr [0];*

*for (count = 1; count < length; count++) {*

*if (arr[count] < minNum)*

*minNum = arr[cout];*

*}*

*return minNum;*

*}*

*int main(){  
 const int k=4;*

*int array[k];*

*cout << "\nMinimum of array elements is " << minFunc (array, k); //------> فراخوانی تابع*

*return 0;*

*}*

* سوال های 21 و 23 و سوال اضافه ی بعد از سوال 25 همین جزوه را با استفاره ازتوابع و ارسال آرایه بازنویسی کنید.

(تألیفی)

* برنامه هایی که برای سوال های 27 و 29 نوشته شده را به صورت دو تابع جداگانه در بیاورید (می‌توانید و بهتر است که در درون این توابع نیز از توابع دیگری استفاده کنید) و آنها را برای ساختن یک برنامه ی کامل درکنار هم قرار دهید. سعی کنید خروجی سوال 29 را به صورت مستقیم به تابع سوال 27 بفرستید (تابع را به عنوان آرگومان صدا بزنید تا خروجی تابع ارسال شود).

(تألیفی)

***5-4***

*توابع بازگشتی توابعی هستند که در درون تعریف خود تابع هم فراخوانی میشوند.*

*البته این نکته هم قابل ذکر ست که حتما شرطی برای تابع بازگشتی باید وجود داشته باشد تا برنامه به درستی اجرا شود و اگر این مورد رعایت نشود تابع بینهایت بار فراخوانی شده و هیچ وقت برنامه به پایان نمی رسد .*

1. سوال 38 را با استفاده از تابع بازگشتی بازنویسی کنید.

(تألیفی)

1. تابعی بنویسید که با دریافت n، جمله ی nام سری فیبوناچی را چاپ کند و به کمک آن 20 جمله اول سری فیبوناچی را بنویسید.

(تألیفی)

1. مسئله ی 8 رخ را با استفاده از جایگشت های مختلف مناسب 8 رخ در صفحه ی شطرنج (که با استفاده از توابع بازگشتی حاصل می شود) حل کنید. برای نمایشنتیجه یک جدول چاپ کنید.

**راهنمایی:** هر رخ می تواند تمامی مهره های هم سطر و هم ستون خودش را بزند. مسئله ی 8 رخ آن هست که تمام حالات ممکن را که در آن ها 8 رخ طوری در صفحه ی 8 در 8 شطرنج قرار می گیرند که نمی توانند همدیگر را بزنند بدست آوریم.

***فصل ششم***

***FileS***

* **مسائل این قسمت مربوط به بخش Files و Structures جزوۀ استاد کریمیان می‌باشند و به ترتیب آسان به سخت و همینطور ترتیب مباحث جزوه مرتب شده اند.**
* **5 کادر آموزشی**
* **2 سوال با کد پاسخ**
* **3 سوال بدون کد پاسخ**

***6-1***

*به صورت کاربردی structure را میتوان تعریف نوع جدیدی از متغیر در نظر گرفت. همانطور که پیشتر بیان شد متغیر ها انواع int, char و ... را دارند.*

*توسط structure میتوان نوع جدیدی از ساختار تعریف کرد که خواصی را که ما میخواهیم دارا باشد.*

*برای مثال:*

*struct Person*

*{*

*char name [50];*

*int age;*

*float salary;*

*};*

*که خواصی مثل ورودی نام به صورت کارکتر و سن به صورت عدد صحیح و حقوق به صورت اعشاری را دارا میباشد.*

*بهتر است مفهوم ساختار را با یک مثال شرح دهیم:*

*#include <iostream>*

*using namespace std;*

*struct Person{  
 char name[50];  
 int age;  
 float salary;  
};*

*int main(){  
 Person p1;   
 cout << "Enter Full name: ";  
 cin.get(p1.name, 50);  
 cout << "Enter age: ";  
 cin >> p1.age;  
 cout << "Enter salary: ";  
 cin >> p1.salary;*

*cout << "\nDisplaying Information." << endl;  
 cout << "Name: " << p1.name << endl;  
 cout <<"Age: " << p1.age << endl;  
 cout << "Salary: " << p1.salary;*

*return 0;*

*}*

* برنامه ی سوال اضافه ی قبل از سوال 31 همین جزوه را با استفاده از ساختارها بازنویسی کنید.

(تألیفی)

***6-2***

*بسیار برای ما پیش آمده که در برنامه نویسی احتیاج به ذخیره اطلاعاتی داشته باشیم که پس از اتمام برنامه باز هم در دسترس باشند و در اجرا های بعدی برنامه هم بتوانیم از آنها استفاده کنیم. در C++ برای رفع این نیاز میتوان اطلاعات مورد نظر را در قالب یک فایل ( متن، باینری و ...) ذخیره کرد و سپس در صورت نیاز از آن استفاده کرد.*

*اما کار با فایل در C++ چگونه ست؟*

*برای شروع کار، ما به کتابخانه fstream احتیاج داریم که شامل سه کلاس کاربردی fstream، ofstream و ifstream میباشد.*

***نحوه تعریف:***

*#include<fstream>*

*#include<iostream>*

*using namespace std;*

*int main(){*

*fstream name (“آدرس فایل”); //فایل مورد نظر به برنامه معرفی می شود*

*//یا*

*ifstream name (“آدرس فایل”);*

*//یا*

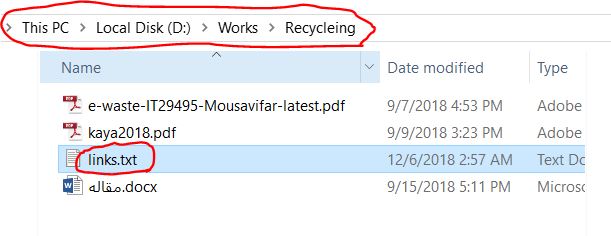
*ofstream name (“آدرس فایل”);*

*return 0;*

*}*

*تفاوت fstream و ifstream و ofstream در چیست؟*

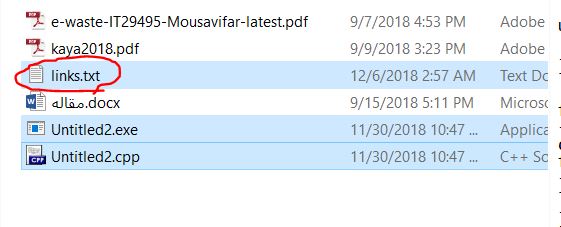
*ifstream زمانی مورد استفاده قرار میگیرد که هدف صرفا خواندن اطلاعات از فایل باشد (in-file stream)  
ofstream زمانی کاربرد دارد که هدف نوشتن اطلاعات درون فایل باشد (out-file stream)  
fstream حالت کلی بوده و برای هر دو حالت کاربرد دارد ولی بهتر است استفاده نشود زیرا باعث بروز برخی باگ ها در برنامه میشود.*

******

***نحوه نوشتن آدرس فایل:***

***1.آدرس دقیق***

***مثال: ”******D:\Works\Recycleing\links.txt”***

******

***2.آدرس فایلی در همان پوشه برنامه یا پوشه قبلی***

***مثال: "links.txt"***

***اگر در پوشه قبلی بود: “..\links.txt”***

***6-3***

***نحوه نوشتن اطلاعات در فایل:***

*کار با فایل شباهت زیادی به فرمان های cin و cout دارد؛برای نوشتن در فایل از روش زیر استفاده میکنیم:*

*ofstream file (“file address”);*

*file << “Matn mored nazar”;*

*اشکال کار چیست؟*

*اگر با همین دستور بخواهید بر روی یک فایل تغییر ایجاد کنید  
با هر بار اجرا شدن، کل فایل پاک شده و اطلاعات جدید که وارد میکنید در آن جای میگیرد.*

*اگر بخواهیم اطلاعات قبلی حذف نشود و اطلاعات جدید به آن اضافه شود چه کنیم؟*

*کافیست کد بالا را به کد زیر تغییر دهید:*

*ofstream file (“file address”, ios::ate); // at the end*

*file << “Matn mored nazar”;*

*ios::ate چیست؟*

*کار این دستور ، بردن محل نوشتن اطلاعات به آخر فایل مورد نظر میباشد؛ یعنی اطلاعات به آخر فایل اضافه میشود.*

*شاید نیاز داشته باشیم که اطلاعات را در بخش خاصی از فایل اضافه کنیم؛مثلا خط دوم فایل یا کارکتر 76 ام فایل...  
برای اینکار میتوانیم از دستور seekp استفاده کنیم.*

***6-4***

***نحوه استفاده:***

*ofstream file(“test.txt”);*

*file.seekp(مکان , موقعیت ) یا file.seekp(pos,direction)*

*که در آن منظور از مکان تعداد کارکتر(فاصله) و منظور از موقعیت یکی از دو حالت زیر است:*

*1.ios::beg : از محل شروع فایل*

*2.ios::end : از محل پایان فایل*

*مثال:*

*file.seekp(10,ios::end);//10 khane ghabl az entehaye file*

***6-5***

***خواندن فایل:***

*اگر فایل متنی باشد:*

*1.خواندن کارکتر به کارکتر*

*#include <iostream>*

*#include <fstream>*

*using namespace std;*

*int main()*

*{ ifstream file ("a.txt");*

*char temp;*

*while (file>>temp)*

*{*

*cout<<temp<<" ";*

*}*

*}*

*2.خواندن کلمه به کلمه*

*#include <iostream>*

*#include <fstream>*

*using namespace std;*

*int main() {*

*ifstream file ("a.txt");*

*string temp;*

*while (file>>temp)*

*{*

*cout<<temp<<" ";*

*}*

*}*

*3.خواندن خط به خط*

*#include <iostream>*

*#include <fstream>*

*using namespace std;*

*int main () {*

*ifstream file ("a.txt");*

*string temp;*

*while (getline (file, temp))*

*{*

*cout<<temp<<" ";*

*}*

*}*

***\*\*خواندن تمام فایل به صورت یک جا (فراتر از کتاب)***

*در صورتی که می خواهید کل فایل رو بصورت یکجا بخونین میتونین از stringstream هم استفاده کنین که به مراتب سریع تر از خوندن خط به خط فایل بوسیله getline هست :*

*#include <iostream>*

*#include <fstream>*

*#include <sstream>*

*using namespace std;*

*int main()*

*{*

*ifstream file("file.txt");*

*string wholeFile;*

*std::stringstream buffer;*

*buffer << file.rdbuf();*

*wholeFile=buffer.str();*

*cout<<wholeFile;*

*return 0 ;*

*}*

1. برنامه ای بنویسید که با دریافت آدرس فایل متنی، فایل را خوانده کارکتر های کوچک را به کارکتر های بزرگ تبدیل کند.(capital letter)

*(تألیفی)*

1. برنامه ای بنویسید که با دریافت آدرس فایل، بررسی کند اگر فایل خالی ست مقدار 0 و اگر شامل متن ست مقدار 1 را بازگرداند.

*(تألیفی)*

* برنامه ای بنویسید که با دریافت فایل متنی، متن را از آخر به اول درون فایل بازنویسی کند.

*(تألیفی)*

* *می توانید برای تمرین، هر یک از تمرینات جزوه را با استفاده از فایل ها حافظه دار کنید. در این فرایند فقط باید توجه داشته باشید که اطلاعات را با نظامی مشخص درون فایل بنویسید که هنگام خواندن از آن بتوانید کار را به حلقه ها بسپاریدو*

*(تألیفی)*

***بانک کد***

***Codes***

//program 1:

#include <iostream>

using namespace std;

int main(){

int integer;

cout << "Enter integer:\t";

cin >> integer;

int c1, c2, c3, c4, c5;

c1 = integer%10;

integer /= 10;

c2 = integer%10;

integer /= 10;

c3 = integer%10;

integer /= 10;

c4 = integer%10;

c5 = integer/10;

cout <<c5<<" "<<c4<<" "<<c3<<" "<<c2<<" "<<c1;

return 0;

}

//programmed by B.Shayegh.

//program 2:

#include <iostream>

#include <cmath>

using namespace std;

int main(){

double A,B,C,a,b,c;

cout << "Enter lengths:\n";

cin >> A >> B >> C;

a = acos( (pow(A,2) - pow(B,2) - pow(C,2))/(-2\*B\*C) );

b = acos( (pow(B,2) - pow(A,2) - pow(C,2))/(-2\*A\*C) );

c = acos( (pow(C,2) - pow(B,2) - pow(A,2))/(-2\*B\*A) );

cout <<endl<<"a = "<<a<<endl<<"b = "<<b<<endl<<"c = "<<c;

return 0;

}

//programmed by B.Shayegh.

//program 3:

#include <iostream>

using namespace std;

int main(){

int a,b;

cout << "Enter a and b:\n";

cin >> a >> b;

int max,min;

if(a>=b) {

max = a;

min = b;

}

if(b>a) {

max = b;

min = a;

}

if(max%min == 0)

cout << max << " is divisible to " << min;

if(max%min != 0)

cout << max << " is not divisible to " << min << ". remain is " << max%min;

return 0;

}

//programmed by B.Shayegh.

//program 4:

#include <iostream>

using namespace std;

int main(){

int number, n1, n2, n3, n4, new\_number;

cin >> number;

n1 = number%10; number /= 10;

n2 = number%10; number /= 10;

n3 = number%10; number /= 10;

n4 = number%10;

n1 = 9-n1 ; n1 += 7; n1 %= 10;

n2 = 9-n2 ; n2 += 7; n2 %= 10;

n3 = 9-n3 ; n3 += 7; n3 %= 10;

n4 = 9-n4 ; n4 += 7; n4 %= 10;

new\_number = n1\*1000 + n2\*100 + n3\*10 + n4;

cout << new\_number;

return 0;

}

//programmed by B.Shayegh.

//program 5:

#include <iostream>

using namespace std;

int main(){

int integer;

cout << "Enter integer:\t";

cin >> integer;

int c1, c2, c3, c4, c5;

c1 = integer%10;

integer /= 10;

c2 = integer%10;

integer /= 10;

c3 = integer%10;

integer /= 10;

c4 = integer%10;

c5 = integer/10;

if(c1==c5 && c2==c4)

cout << "Yes it is.";

if(c1!=c5 || c2!=c4)

cout << "No it isn't'";

return 0;

}

//programmed by B.Shayegh.

//program 6:

#include <iostream>

using namespace std;

int main(){

int number;

do{

cout << "Enter a new number.\n";

cin >> number;

if (number%2 == 0)

cout << "This number is even.\n\n";

if (number%2 != 0)

cout << "This number is odd.\n\n";

} while(number != 0);

return 0;

}

//programmed by B.Shayegh.

//program 7:

#include <iostream>

using namespace std;

int main(){

int number, min, max;

min = max = 0;

do{

cout << "Enter a new number. Enter 0 to Exit.\n";

cin >> number;

if (number > max) max = number;

if (number < min) min = number;

} while(number != 0);

cout << endl << "Max is :\t" << max << endl << "Min is :\t" << min;

return 0;

}

//programmed by B.Shayegh.

//program 8:

#include <iostream>

using namespace std;

int main(){

char answer;

int point=0;

bool done=false;

cout << "Hi. now you are in A. select your way:\nB\tD\n";

cin >> answer;

if(answer=='B'){point+=10;

cout << "well, now you are in B. select your way:\nC\tE\n";

cin >> answer;

if(answer=='C'){point+=35;

cout << "well, now you are in C. select your way:\nF\n";

cin >> answer;

if(answer=='F'){point+=10;

cout << "well, now you are in F. select your way:\nJ\tQ\n";

cin >> answer;

if(answer=='J'){point+=35;

cout << "well, now you are in J. select your way:\nK\n";

cin >> answer;

if(answer=='K'){point+=35;

cout << "well, now you are in K. select your way:\nP\n";

cin >> answer;

if(answer=='P'){point+=35;

cout << "well, now you are in P. select your way:\nG\n";

cin >> answer;

if(answer=='G'){point+=35;

cout << "well, now you are in G. select your way:\nM\n";

cin >> answer;

if(answer=='M'){point+=35;

cout << "well, now you are in M. its end of line.\n";

done = true;

}

}

}

}

}

else if(answer=='T'){point+=35;

cout << "well, now you are in T. select your way:\nZ\n";

cin >> answer;

if(answer=='Z'){point+=35;

cout << "well, now you are in M. its end of line.\n";

done = true;

}

}

else if(answer=='Q'){point+=35;

cout << "well, now you are in Q. its end of line.\n";

done = true;

}

}

}

else if(answer=='E'){point+=10;

cout << "well, now you are in E. select your way:\nH\n";

cin >> answer;

if(answer=='H'){point+=10;

cout << "well, now you are in H. select your way:\nI\tR\n";

cin >> answer;

if(answer=='I'){point+=35;

cout << "well, now you are in I. select your way:\nO\tX\n";

cin >> answer;

if(answer=='O'){point+=35;

cout << "well, now you are in O. select your way:\nN\n";

cin >> answer;

if(answer=='N'){point+=20;

cout << "well, now you are in N. its end of line.\n";

done = true;

}

}

if(answer=='X'){point+=35;

cout << "well, now you are in X. its end of line.\n";

done = true;

}

}

if(answer=='R'){point+=35;

cout << "well, now you are in R. select your way:\nL\tS\n";

cin >> answer;

if(answer=='L'){point+=20;

cout << "well, now you are in L. its end of line.\n";

done = true;

}

if(answer=='S'){point+=10;

cout << "well, now you are in S. its end of line.\n";

done = true;

}

}

}

}

}

else if(answer=='D'){point+=35;

cout << "well, now you are in D. its end of line.\n";

done = true;

}

if(done) cout << "Your point is:\t" << point;

else cout << "Game Over.";

return 0;

}

//programmed by B.Shayegh.

//program 9:

#include <iostream>

using namespace std;

int main(){

int m1, m2, m3, m4, m5;

int n1, n2, n3, n4, n5;

cout << "Enter first mark:\t";

cin >> m1;

if(m1<0 || m1>20){

cout << "Wrong mark. try again. mark should be between 0 and 20.\n";

cin >> m1;

if(m1<0 || m1>20){

cout << "Wrong mark!";

return 0;

}

}

cout << "Now, Enter weight of this mark:\t";

cin >> n1;

if(n1<0 || n1>4){

cout << "Wrong. try again. it should be between 0 and 4.\n";

cin >> n1;

if(n1<0 || n1>4){

cout << "Wrong!";

return 0;

}

}

cout << "Enter mark 2:\t";

cin >> m2;

if(m2<0 || m2>20){

cout << "Wrong mark. try again. mark should be between 0 and 20.\n";

cin >> m2;

if(m2<0 || m2>20){

cout << "Wrong mark!";

return 0;

}

}

cout << "Now, Enter weight of this mark:\t";

cin >> n2;

if(n2<0 || n2>4){

cout << "Wrong. try again. it should be between 0 and 4.\n";

cin >> n2;

if(n2<0 || n2>4){

cout << "Wrong!";

return 0;

}

}

cout << "Enter first mark:\t";

cin >> m2;

if(m2<0 || m2>20){

cout << "Wrong mark. try again. mark should be between 0 and 20.\n";

cin >> m2;

if(m2<0 || m2>20){

cout << "Wrong mark!";

return 0;

}

}

cout << "Now, Enter weight of this mark:\t";

cin >> n2;

if(n2<0 || n2>4){

cout << "Wrong. try again. it should be between 0 and 4.\n";

cin >> n2;

if(n2<0 || n2>4){

cout << "Wrong!";

return 0;

}

}

cout << "Enter mark 3:\t";

cin >> m3;

if(m3<0 || m3>20){

cout << "Wrong mark. try again. mark should be between 0 and 20.\n";

cin >> m3;

if(m3<0 || m3>20){

cout << "Wrong mark!";

return 0;

}

}

cout << "Now, Enter weight of this mark:\t";

cin >> n3;

if(n3<0 || n3>4){

cout << "Wrong. try again. it should be between 0 and 4.\n";

cin >> n3;

if(n3<0 || n3>4){

cout << "Wrong!";

return 0;

}

}

cout << "Enter mark 4:\t";

cin >> m4;

if(m4<0 || m4>20){

cout << "Wrong mark. try again. mark should be between 0 and 20.\n";

cin >> m4;

if(m4<0 || m4>20){

cout << "Wrong mark!";

return 0;

}

}

cout << "Now, Enter weight of this mark:\t";

cin >> n4;

if(n4<0 || n4>4){

cout << "Wrong. try again. it should be between 0 and 4.\n";

cin >> n4;

if(n4<0 || n4>4){

cout << "Wrong!";

return 0;

}

}

cout << "Enter mark 5:\t";

cin >> m5;

if(m5<0 || m5>20){

cout << "Wrong mark. try again. mark should be between 0 and 20.\n";

cin >> m5;

if(m5<0 || m5>20){

cout << "Wrong mark!";

return 0;

}

}

cout << "Now, Enter weight of this mark:\t";

cin >> n5;

if(n5<0 || n5>4){

cout << "Wrong. try again. it should be between 0 and 4.\n";

cin >> n5;

if(n5<0 || n5>4){

cout << "Wrong!";

return 0;

}

}

double w;

w=((m1\*n1)+(m2\*n2)+(m3\*n3)+(m4\*n4)+(m5\*n5)) / (n1+n2+n3+n4+n5);

if(w<4) cout<<endl<<"average is: E";

else if(w<8) cout<<endl<<"average is: D";

else if(w<12) cout<<endl<<"average is: C";

else if(w<16) cout<<endl<<"average is: B";

else if(w<20) cout<<endl<<"average is: A";

else cout<<endl<<"average is: A+";

return 0;

}

//programmed by B.Shayegh.

//program 10:

#include <iostream>

using namespace std;

int main(){

double x, answer=1;

int y;

cout << "Enter x and then y for x^y:\n";

cin >> x >> y;

int i=0;

while(++i<=y)

answer \*= x;

cout << endl << "answer = " << answer;

return 0;

}

//programmed by B.Shayegh.

//program 11:

#include <iostream>

using namespace std;

int main(){

int number, i;

bool prime;

do{

cout << "\nEnter a new number. Enter 0 to exit.\n";

cin >> number;

i=number;

prime=true;

while (--i > 1)

if (number%i == 0) prime=false;

if (number==0 || number==1)

cout << "Nothing to say.\n";

else if (prime)

cout << "This number is a prime number.\n";

else

cout << "This number is not a prime number.\n";

} while(number != 0);

return 0;

}

//programmed by B.Shayegh.

//program 12:

#include <iostream>

using namespace std;

int main(){

int x, answer=1;

cout << "Enter x for x!:\n";

cin >> x;

x++;

while(--x>0)

answer \*= x;

cout << endl << "answer = " << answer;

return 0;

}

//programmed by B.Shayegh.

//program 13:

#include <iostream>

using namespace std;

int main(){

int n;

double answer=0;

cout << "Enter n:\n";

cin >> n;

int i=0;

while(++i <= n){

double a = 4/(2\*double(i) - 1);

if(i%2 != 0) answer += a;

else answer -= a;

}

cout << endl << "answer = " << answer;

return 0;

}

//programmed by B.Shayegh.

//program 14:

#include <iostream>

using namespace std;

int main(){

int N, sum;

bool prime;

do{

cout << "\nEnter a new number. Enter 0 to exit.\n";

cin >> N;

sum=0;

for (int counter=2; counter<=N; counter++){

prime=true;

for (int i=counter-1; i>1; i--)

if (counter%i == 0) prime=false;

if (prime) sum += counter;}

cout << "Sum = " << sum << endl;

} while(N != 0);

return 0;

}

//programmed by B.Shayegh.

//program 15:

#include <iostream>

#include <cmath>

using namespace std;

int main(){

int n=0, oo, factor\_n;

double x, ex1=1, ex2, power\_x\_n;

cout << "Enter x:\t";

cin >> x;

cout << "now Enter Number of decimal places:\t";

cin >> oo;

do{ n++;

ex2 = ex1;

factor\_n=1;

for(int i=n; i>0; i--) factor\_n \*=i;

power\_x\_n=1;

for(int i=1; i<=n; i++) power\_x\_n \*= x;

ex1 += power\_x\_n/factor\_n;

} while(ex2-ex1 > pow(10,-1\*oo) || ex1-ex2 > pow(10,-1\*oo));

ex1 = int(ex1\*pow(10,oo))/pow(10,oo);

cout << ex1;

return 0;}

//programmed by B.Shayegh.

//program 16:

#include <iostream>

using namespace std;

int main(){

double x1,x2,x3 , c11,c12,c13,c21,c22,c23,c31,c32,c33 , b1,b2,b3 , a , replace;

cout << "Welcome. you should have a matris like this for your equations:" << endl;

cout <<"c11\tc12\tc13\t\tb1\nc21\tc22\tc23\t\tb2\nc31\tc32\tc33\t\tb3\n";

cout << "Enter the coefficients (Cs):" << endl;

cout << "for first equation, enter coefficients in ORDER (c11,c12,c13)" << endl << " (Enter 0 if there is not one of vars.):" << endl;

cin >> c11 >> c12 >> c13;

cout << "for second equation, enter coefficients in ORDER (c21,c22,c23)" << endl << " (Enter 0 if there is not one of vars.):" << endl;

cin >> c21 >> c22 >> c23;

cout << "for third equation, enter coefficients in ORDER (c31,c32,c33)" << endl << " (Enter 0 if there is not one of vars.):" << endl;

cin >> c31 >> c32 >> c33;

cout << "Now enter integer (Bs) of equations in ORDER (b1,b2,b3):" << endl;

cin >> b1 >> b2 >> b3;

cout << "Ok. Your matris is:" << endl;

cout <<c11<<"\t"<<c12<<"\t"<<c13<<"\t\t"<<b1<<"\n"<<c21<<"\t"<<c22<<"\t"<<c23<<"\t\t"<<b2<<"\n"<<c31<<"\t"<<c32<<"\t"<<c33<<"\t\t"<<b3<<"\n";

if (c11==0 && c21!=0) {

replace=c11; c11=c21; c21=replace;

replace=c12; c12=c22; c22=replace;

replace=c13; c13=c23; c23=replace;

}

else if (c11==0 && c31!=0) {

replace=c11; c11=c31; c31=replace;

replace=c12; c12=c32; c32=replace;

replace=c13; c13=c33; c33=replace;

}

else if (c11==0) {

cout << "WRONG!!"; return 0;

}

a=c21/c11;

c21 -= c11\*a; c22 -= c12\*a; c23 -= c13\*a; b2 -= b1\*a;

a=c31/c11;

c31 -= c11\*a; c32 -= c12\*a; c33 -= c13\*a; b3 -= b1\*a;

if (c22==0 && c32!=0) {

replace=c21; c21=c31; c31=replace;

replace=c22; c22=c32; c32=replace;

replace=c23; c23=c33; c33=replace;

}

else if (c22==0) {

cout << "WRONG!!";

return 0;

}

a=c32/c22;

c31 -= c21\*a; c32 -= c22\*a; c33 -= c23\*a; b3 -= b2\*a;

if (c33==0) {

cout << "WRONG!!";

return 0;

}

a=c23/c33;

c23 -= c33\*a; c22 -= c32\*a; c21 -= c31\*a; b2 -= b3\*a;

a=c13/c33;

c13 -= c33\*a; c12 -= c32\*a; c11 -= c31\*a; b1 -= b3\*a;

a=c12/c22;

c13 -= c23\*a; c12 -= c22\*a; c11 -= c21\*a; b1 -= b2\*a;

x1 = b1/c11; x2 = b2/c22; x3 = b3/c33;

cout << "Now the matris is:" << endl;

cout <<c11<<"\t"<<c12<<"\t"<<c13<<"\t\t"<<b1<<"\n"<<c21<<"\t"<<c22<<"\t"<<c23<<"\t\t"<<b2<<"\n"<<c31<<"\t"<<c32<<"\t"<<c33<<"\t\t"<<b3<<"\n";

cout << "So:\tx1=" << x1 << "\tx2= " << x2 << "\tx3= " << x3 << endl;

return 0;

}

//programmed by B.Shayegh.

//program 17:

#include <iostream>

using namespace std;

int main(){

char choic;

int point = -12;

cout << "Rules of the game." << endl << "OK? So start..." << endl;

A: point+=12;

cout << "You are in A now. where you want to go?" << endl;

cout << "B\tC" << endl;

cin >> choic;

switch (choic){

case 'B': goto B;

case 'C': goto C;

default: cout << "Wrong choic. Point = " << point; return 0;

}

B: point+=14;

cout << "You are in B now. where you want to go?" << endl;

cout << "A\tF\tk\to\ti" << endl;

cin >> choic;

switch (choic){

case 'A': goto A;

case 'F': goto F;

case 'k': goto k;

case 'o': goto o;

case 'i': goto i;

default: cout << "Wrong choic. Point = " << point; return 0;

}

C: point+=9;

cout << "You are in C now. where you want to go?" << endl;

cout << "D\tg" << endl;

cin >> choic;

switch (choic){

case 'D': goto D;

case 'g': goto g;

default: cout << "Wrong choic. Point = " << point; return 0;

}

D: point+=15;

cout << "You are in D now. where you want to go?" << endl;

cout << "E\tF\tB\tk\to\ti" << endl;

cin >> choic;

switch (choic){

case 'E': goto E;

case 'F': goto F;

case 'B': goto B;

case 'k': goto k;

case 'o': goto o;

case 'i': goto i;

default: cout << "Wrong choic. Point = " << point; return 0;

}

E: point+=10;

cout << "You are in E now. where you want to go?" << endl;

cout << "D\tj" << endl;

cin >> choic;

switch (choic){

case 'D': goto D;

case 'j': goto j;

default: cout << "Wrong choic. Point = " << point; return 0;

}

F: point+=15;

cout << "You are in F now. where you want to go?" << endl;

cout << "B\tk\ti\to\tf" << endl;

cin >> choic;

switch (choic){

case 'B': goto B;

case 'k': goto k;

case 'i': goto i;

case 'o': goto o;

case 'f': goto f;

default: cout << "Wrong choic. Point = " << point; return 0;

}

f: point+=1;

cout << "You are in f now. where you want to go?" << endl;

cout << "e" << endl;

cin >> choic;

switch (choic){

case 'e': goto e;

default: cout << "Wrong choic. Point = " << point; return 0;

}

a: point+=3;

cout << "You are in a now. where you want to go?" << endl;

cout << "b" << endl;

cin >> choic;

switch (choic){

case 'b': goto b;

default: cout << "Wrong choic. Point = " << point; return 0;

}

e: point+=1;

cout << "You are in e now. where you want to go?" << endl;

cout << "f\td\tb" << endl;

cin >> choic;

switch (choic){

case 'f': goto f;

case 'd': goto d;

case 'b': goto b;

default: cout << "Wrong choic. Point = " << point; return 0;

}

g: point+=2;

cout << "You are in g now. where you want to go?" << endl;

cout << "C\th" << endl;

cin >> choic;

switch (choic){

case 'C': goto C;

case 'h': goto h;

default: cout << "Wrong choic. Point = " << point; return 0;

}

b: point+=2;

cout << "You are in b now. where you want to go?" << endl;

cout << "a\tc" << endl;

cin >> choic;

switch (choic){

case 'a': goto a;

case 'c': goto c;

default: cout << "Wrong choic. Point = " << point; return 0;

}

k: point+=4;

cout << "You are in k now. where you want to go?" << endl;

cout << "B\tF\ti\to\tm\tp" << endl;

cin >> choic;

switch (choic){

case 'B': goto B;

case 'F': goto F;

case 'i': goto i;

case 'o': goto o;

case 'm': goto m;

case 'p': goto p;

default: cout << "Wrong choic. Point = " << point; return 0;

}

p: point+=1;

cout << "You are in p now. where you want to go?" << endl;

cout << "k\tm\tl\tn" << endl;

cin >> choic;

switch (choic){

case 'k': goto k;

case 'l': goto l;

case 'n': goto n;

case 'm': goto m;

default: cout << "Wrong choic. Point = " << point; return 0;

}

h: point+=1;

cout << "You are in h now and its the end of the line" << endl;

cout << "Point = " << point; return 0;

i: point+=2;

cout << "You are in i now and its the end of the line" << endl;

cout << "Point = " << point; return 0;

j: point+=3;

cout << "You are in j now and its the end of the line" << endl;

cout << "Point = " << point; return 0;

o: point+=2;

cout << "You are in o now and its the end of the line" << endl;

cout << "Point = " << point; return 0;

d: point+=4;

cout << "You are in d now and its the end of the line" << endl;

cout << "Point = " << point; return 0;

c: point+=3;

cout << "You are in c now and its the end of the line" << endl;

cout << "Point = " << point; return 0;

m: point+=1;

cout << "You are in m now and its the end of the line" << endl;

cout << "Point = " << point; return 0;

l: point+=3;

cout << "You are in l now and its the end of the line" << endl;

cout << "Point = " << point; return 0;

n: point+=3;

cout << "You are in n now and its the end of the line" << endl;

cout << "Point = " << point; return 0;

return 0;

}

//programmed by B.Shayegh.

//program 18:

#include <iostream>

using namespace std;

int main(){

int N;

double number, Max\_1, Max\_2, Max\_3;

cout << "Enter \"N\":\t";

cin >> N;

cout << "Ok, Now enter " << N << " number:\n";

cin >> number;

Max\_1 = Max\_2 = Max\_3 = number;

for(int i=1; i<N; i++){

cin >> number;

if (number > Max\_1){

Max\_3 = Max\_2;

Max\_2 = Max\_1;

Max\_1 = number;

}

else if (number > Max\_2){

Max\_3 = Max\_2;

Max\_2 = number;

}

else if (number > Max\_3)

Max\_3 = number;

}

cout << endl << "OK. third Max number is:\t" << Max\_3;

return 0;

}

//programmed by B.Shayegh.

//program 19:

#include <iostream>

#include <cmath>

using namespace std;

int main(){

int N;

double d, m, v=0;

cout << "Enter N:\t";

cin >> N;

cout << "Ok, Start:\n";

for (int i=1; i<=N; i++){

cin >> d;

m += d;

}

m /= N;

cout << "sorry. you should do this once again.\n";

for (int i=1; i<=N; i++){

cin >> d;

v += pow( (d-m) , 2 );

}

v /= N;

cout << endl << "Ok, variance is:\t" << v;

return 0;

}

//programmed by B.Shayegh.

//program 20:

#include <iostream>

using namespace std;

int main(){

int a;

cout << "Enter \"a\" (between 1 and 20):\t";

cin >> a;

if (a<1 || a>20){

cout << "WRONG!!"; return 0;

}

for (int i=0; i<a; i++)

cout << "\*";

for (int i=0; i<a-2; i++){

cout << endl << "\*";

for (int j=0; j<a-2; j++) cout << " ";

cout << "\*";

}

cout << endl;

for (int i=0; i<a; i++) cout << "\*";

return 0;

}

//programmed by B.Shayegh.

//program 21:

#include <iostream>

#include <cmath>

using namespace std;

int main(){

int N;

cout << "Enter N:\t";

cin >> N;

double d[N], v=0, m=0;

cout << "Ok, Start:\n";

for (int i=0; i<N; i++)

cin >> d[i];

for (int i=0; i<N; i++)

m += d[i];

m /= N;

for (int i=0; i<N; i++)

v += pow(d[i]-m , 2);

v /= N;

cout << endl << "Ok, variance is:\t" << v;

return 0;

}

//programmed by B.Shayegh.

//program 22:

#include <iostream>

using namespace std;

int main(){

double mark;

int S[21]={0}, k=0;

cout << "Enter \"1\" when ypu have done." << endl;

do{

cout << "Enter mark" << ++k << "\t";

cin >> mark;

if (mark == -1)

cout << "Exit mark. okay, thanks." << endl;

else if (mark>20 || mark<0)

cout << "Wrong mark" << k-- << ". try again" << endl;

else S[ int(mark) ] ++;

} while (mark != -1);

int mode\_mark=0, number\_of\_mode\_mark=0;

for (int i=0; i<=20; i++)

if (S[i] >= number\_of\_mode\_mark){

number\_of\_mode\_mark = S[i];

mode\_mark = i;

}

if (mode\_mark==20) cout << endl << "Marks " << mode\_mark << " are " << number\_of\_mode\_mark << " number and must in number between other marks." << endl;

else cout << endl << "Marks between " << mode\_mark << " and " << mode\_mark+1 << " are " << number\_of\_mode\_mark << " number and must in number between other marks." << endl;

for (int i=0; i<20; i++){

cout << i << "\_" << i+1 << "\t" << S[i] << "\t";

for (int j=0; j<S[i]; j++) cout<<"\*";

cout << endl;

}

cout << 20 << "\t" << S[20] << "\t";

for (int j=0; j<S[20]; j++) cout<<"\*";

return 0;

}

//programmed by B.Shayegh.

//program 23:

#include <iostream>

using namespace std;

int main(){

int N;

cout << "Please Enter N (number of marks)" << endl;

cin >> N;

int mark[N];

cout << "Now Enter marks one after one:" << endl;

for (int i=0; i<N; i++) cin >> mark[i];

int number\_of\_mark[N];

for (int i=0; i<N; i++) number\_of\_mark[i]=1;

for (int i=0; i<N-1; i++)

for (int j=i+1; j<N; j++)

if (mark[i] == mark[j]){

number\_of\_mark[i]++;

number\_of\_mark[j]++;

}

int max\_of\_number\_of\_marks=0, mode\_of\_marks;

for (int i=0; i<N; i++)

if (number\_of\_mark[i] > max\_of\_number\_of\_marks){

max\_of\_number\_of\_marks = number\_of\_mark[i];

mode\_of\_marks = mark[i];

}

cout << endl << "Ok, mode of marks is " << mode\_of\_marks << " and number of mode mark is " << max\_of\_number\_of\_marks;

return 0;

}

//programmed by B.Shayegh.

//program 24:

#include <iostream>

using namespace std;

int main(){

int A[10], C;

cout << "Enter parametrs:" << endl;

for (int i=0; i<10; i++)

cin >> A[i];

for (int i=0; i<9; i++)

for (int j=0; j<9; j++)

if (A[j] > A[j+1]){

C = A[j+1];

A[j+1] = A[j];

A[j] = C;

}

for (int i=0; i<10; i++) cout << A[i] << "\t";

return 0;

}

//programmed by B.Shayegh.

//program 25:

#include <iostream>

using namespace std;

int main(){

bool prime[1000];

for(int i=1; i<1000; i++)

prime[i]=true;

for(int i=2; i<1000; i++)

if(prime[i])

for(int j=2; i\*j<100; j++)

prime[i\*j] = false;

for(int i=1; i<100; i++)

if(prime[i]) cout << i << endl;

return 0;

}

//programmed by B.Shayegh.

//program 26:

#include <iostream>

using namespace std;

int main(){

int num[20], answer[20]={0}, final\_answer[20]={0};

cout << "Enter numbers:\n";

for(int i=0; i<20; i++) cin >> num[i];

int j=0, k=0;

for(int i=0; i<20;){

answer[j++] = num[i++];

if(num[i] >= num[i-1]) continue;

if(j-1 > k) {k=j-1;

for(int s=0; s<=k; s++) final\_answer[s] = answer[s];}

j=0;

}

for(int i=0; i<=k; i++) cout << endl << final\_answer[i];

return 0;

}

//programmed by B.Shayegh.

//program 27:

#include <iostream>

using namespace std;

int main(){

int A[10][10] , I[100]={0} , J[100]={0}, length=0;

get:

for (int j=0; j<10; j++)

for (int i=0; i<10; i++){

cout<< "Enter A" <<j<<","<<i<<":\t";

cin >> A[j][i];

if (A[j][i] <= 0) i--;

}

cout << endl << "Is this what you said?" << endl;

for (int j=0; j<10; j++){

for (int i=0; i<10; i++) cout << A[j][i] << "\t";

cout << endl;

}

cout << "answer with \"Y\" (Yes) or \"N\" (No):\t";

char answer;

while (cin >> answer){

if (answer == 'Y') break;

if (answer == 'N') goto get;

else cout << "What?\t";

}

int pointer;

for (int j=0; j<10; j++)

for (int i=0; i<10; i++){

int II[100]={0}, JJ[100]={0};

int B[10][10]={0};

for (int jjj=0; jjj<10; jjj++)

for (int iii=0; iii<10; iii++)

B[jjj][iii] = A[jjj][iii];

int ii=i, jj=j;

char last[100];

for(int k=0; true; k++){

II[k]=ii; JJ[k]=jj;

pointer = B[jj][ii];

if (B[jj+1][ii] == pointer+1 && jj!=9) {

B[jj][ii] = -1;

jj++; last[k] = 'D';

continue;

}

else if (B[jj-1][ii] == pointer+1 && jj!=0) {

B[jj][ii] = -1;

jj--; last[k] = 'U';

continue;

}

else if (B[jj][ii+1] == pointer+1 && ii!=9) {

B[jj][ii] = -1;

ii++; last[k] = 'R';

continue;

}

else if (B[jj][ii-1] == pointer+1 && ii!=0) {

B[jj][ii] = -1;

ii--; last[k] = 'L';

continue;

}

else if (k==0) goto once\_more;

else {

if(k > length) {length = k;

for (int w=0; w<=k; w++){

I[w] = II[w];

J[w] = JJ[w];

}

}

B[jj][ii] = -1;

switch (last[k-1]){

case 'D': jj--; break;

case 'U': jj++; break;

case 'R': ii--; break;

case 'L': ii++; break;

}

B[jj][ii] = --pointer;

II[k] = JJ[k] = 0;

k-=2; continue;

}

}

once\_more: continue;

}

cout << endl << "The longest line (with "<< length+1 <<" parametrs) is:" << endl;

for(int w=0; w <= length; w++)

cout << "A" << J[w] << "," << I[w] << " =\t" << A[J[w]][I[w]] << endl;

return 0;

}

//programmed by B.Shayegh.

//program 28:

#include <iostream>

#include <stdlib.h>

#include <iomanip>

using namespace std;

int main(){

int R1, R2, R\_sum, sum\_num[13]={0};

for (int i=0; i<36000; i++){

R1 = rand()%6 +1;

R2 = rand()%6 +1;

R\_sum = R1+R2;

sum\_num[R\_sum]++ ;

}

cout << " Total:";

for (int i=2; i<=12; i++) cout << setw(8) << i;

cout << endl << "Repeat:";

for (int i=2; i<=12; i++) cout << setw(8) << sum\_num[i];

return 0;

}

//programmed by B.Shayegh.

//program 29:

#include <iostream>

#include <stdlib.h>

using namespace std;

int main(){

int A[10][10];

unsigned seed;

cout << "Enter \"seed\":\t";

cin >> seed;

srand(seed);

for (int j=0; j<10; j++)

for (int i=0; i<10; i++)

A[i][j] = rand()%100;

for (int j=0; j<10; j++){

for (int i=0; i<10; i++)

cout << A[i][j] << "\t";

cout << endl;

}

return 0;

}

//programmed by B.Shayegh.

//program 30:

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

int main(){

int N, user\_answer, real\_answer, point=0;

srand( time(0) );

do{

cout << "Enter N (enter 0 to exit, dont enter 1.):\t";

cin >> N;

if(N==1) continue;

int num[N];

real\_answer=0;

for(int i=0; i<N; i++){

num[i] = rand()%990 +10;

real\_answer += num[i];

}

cout << "Ok. Now answer question:" << endl;

for (int i=0; i<N-1; i++)

cout << num[i] << " + ";

cout << num[N-1] << " = ?" << endl;

cin >> user\_answer;

if (user\_answer == real\_answer){

point += N;

cout << "True. good, your point is:\t" << point << endl;

}

else {

point -= 5;

cout << "Oh no... sorry. your point is:\t" << point << endl;

}

} while (N != 0);

return 0;

}

//programmed by B.Shayegh

//program 31:

#include <iostream>

using namespace std;

int main(){

int line[8];

bool answer;

for (line[0]=0; line[0]<8; line[0]++)

for (line[1]=0; line[1]<8; line[1]++)

for (line[2]=0; line[2]<8; line[2]++)

for (line[3]=0; line[3]<8; line[3]++)

for (line[4]=0; line[4]<8; line[4]++)

for (line[5]=0; line[5]<8; line[5]++)

for (line[6]=0; line[6]<8; line[6]++)

for (line[7]=0; line[7]<8; line[7]++)

{

answer = true;

for(int i=0; i<7; i++)

for(int j=i+1; j<8; j++){

if(line[i]==line[j]) answer = false;

if(line[i]+j-i==line[j] || line[i]-j+i==line[j]) answer = false;

}

if(answer) goto write;}

cout << "No answer!"; return 0;

write:

for(int i=0; i<8; i++){

for(int j=0; j<line[i]; j++)

cout << "\t|";

cout << " @";

for(int j=line[i]; j<8; j++)

cout << "\t|";

cout << endl << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

}

return 0;

}

//programmed by B.Shayegh..

//program 32:

#include <iostream>

using namespace std;

int main(){

char str;

int counter=0;

while (cin >> str && str != '0'){

counter++;

char string1[counter];

for(int i=0; i<counter-1; i++)

string1[i] = string2[i];

string1[counter-1] = str;

char string2[counter];

for(int i=0; i<counter; i++)

string2[i] = string1[i];

}

for (int i=0; i<counter; i++)

cout << string1[i];

return 0;

}

//programmed by B.Shayegh.

//program 33:

#include <iostream>

#include <stdlib.h>

using namespace std;

int main(){

char first\_name[11][10]={0}, last\_name[11][10]={0}, born\_place[11][10]={0}, birth\_day[11][10]={0}, birth\_day2[11][10]={0}, job[11][10]={0};

int age[10], code[11]={0}, pass[11];

bool account\_using[10]={0}, answer, stop=false;

enum Sexuality {male=1, female=0};

Sexuality sexuality[10];

enum Marital\_Status {married=1, non\_married=0};

Marital\_Status marital\_status[10];

cout << "Hi, welcome to information bank.\n";

do{

cout << "What you want to do?\n";

cout << "0) Sign up a new account.\n";

cout << "1) Sign in.\n";

cin >> answer;

if(answer){

cout << "Enter your code:\t";

cin >> code[10];

int j, i=0;

for(; i<10; i++){

if(code[10] == code[i]) break;

else if(i!= 9) continue;

cout << "Wrong code!!!\n";

goto use\_again;

}

cout << "Enter password:\t";

cin >> pass[10];

if(pass[10] != pass[i]){

cout << "Wrong code!!!\n";

goto use\_again;

}

cout << endl << "Ok, You can see your information:\n";

cout << "Name:\t";

j=0;

do cout << first\_name[j][i]; while(first\_name[++j][i] != '\n');

cout << "\t";

j=0;

do cout << last\_name[j][i]; while(last\_name[j++][i] != '\n');

cout << "Born Place:\t";

j=0;

do cout << born\_place[j][i]; while(born\_place[j++][i] != '\n');

cout << "Birth day:\t";

j=0;

do cout << birth\_day[j][i]; while(birth\_day[j++][i] != '\n');

cout << "Job:\t";

j=0;

do cout << job[j][i]; while(job[j++][i] != '\n');

cout << "Age:\t";

cout << age[i] << endl;

cout << "Sexuality:\t";

cout << sexuality[i] << endl;

cout << "Marital Status:\t";

cout << marital\_status[i] << endl;

}

else{

int j, i=0;

for(; i<10; i++){

if(!account\_using[i]) break;

if(i==9){

cout << "There is no free account for sign up.\n";

goto use\_again;

}

}

code[i] = rand();

account\_using[i] = true;

cout << "account " << code[i] << endl;

cout << "First name?\n";

j=0;

do cin.get(first\_name[j][i]); while(first\_name[j++][i] != '\n');

j=0;

do cin.get(first\_name[j][i]); while(first\_name[j++][i] != '\n');

cout << "Last name?\n";

j=0;

do cin.get(last\_name[j][i]); while(last\_name[j++][i] != '\n');

cout << "Where did you born?\n";

j=0;

do cin.get(born\_place[j][i]); while(born\_place[j++][i] != '\n');

cout << "Birth day? Enter like this: 1378/05/16 or 05/16.\n";

j=0;

try\_a\_b\_d: do cin.get(birth\_day2[j][i]); while(birth\_day2[j++][i] != '\n');

if(birth\_day2[4][i] == '/'){

birth\_day[0][i] = birth\_day2[5][i];

birth\_day[1][i] = birth\_day2[6][i];

birth\_day[2][i] = birth\_day2[7][i];

birth\_day[3][i] = birth\_day2[8][i];

birth\_day[4][i] = birth\_day2[9][i];

birth\_day[5][i] = '\n';

}

else if(birth\_day2[5][i] != '\n'){

cout << "Wrong! try again:\n";

goto try\_a\_b\_d;

}

else{

birth\_day[0][i] = birth\_day2[0][i];

birth\_day[1][i] = birth\_day2[1][i];

birth\_day[2][i] = birth\_day2[2][i];

birth\_day[3][i] = birth\_day2[3][i];

birth\_day[4][i] = birth\_day2[4][i];

birth\_day[5][i] = '\n';

}

if(birth\_day[2][i] != '/'){

cout << "Wrong! try again:\n";

goto try\_a\_b\_d;

}

cout << "Job?\n";

j=0;

do cin.get(job[j][i]); while(first\_name[j++][i] != '\n');

cout << "Age?\n";

cin >> age[i];

cout << "Sexuality?\n1)male\t0)female\n";

cin >> answer;

sexuality[i] = Sexuality(answer);

cout << "Marital\_Status?\n1)married\t0)non\_married\n";

cin >> answer;

marital\_status[i] = Marital\_Status(answer);

cout << "Enter password:\n";

cin >> pass[i];

cout << "Its done...\n\n";

}

use\_again:

cout << "Do you want to use the bank again?\n0)Yes\t1)No\n";

cin >> stop;

}while(!stop);

return 0;}

//programmed by B.Shayegh.

//program 34:

#include <iostream>

#include <conio.h>

using namespace std;

int main(){

char ch;

int i=1;

bool prime;

do{ i++;

for(;;i++){

prime = true;

for(int j=2; j<i; j++)

if (i%j == 0) prime = false;

if(prime) break;

}

cout << i << endl;

do ch = getch(); while(ch!='e'&&ch!='E'&&ch!='n'&&ch!='N');

}while(ch != 'e' && ch != 'E');

return 0;

}

//programmed by B.Shayegh.

//program 35:

#include <iostream>

#include<iomanip>

#include <conio.h>

using namespace std;

int main(){

int right=0, down=0;

char move;

bool Move;

cout << "\*";

while (move = getch()){

Move = true;

switch (move){

case 'a':

case 'A':

if(right>0) right--;

break;

case 'd':

case 'D':

right++;

break;

case 'w':

case 'W':

if(down>0) down--;

break;

case 's':

case 'S':

down++;

break;

default: Move = false;

}

if(Move){

system("cls");

for(int i=0; i<down; i++) cout << endl;

cout << setw(right+1) << "\*";

}

}

return 0;

}

//programmed by B.Shayegh.

//program 36:

#include <iostream>

#include <string>

using namespace std;

int main(){

string S= "Ali\tZahra\tRashid\tZohre\tShahla\tNima\tReza\tSohrab\tNahid\n";

string \_(60,'\_');

string str[9] , str0;

cout << S << \_ << endl;

S.erase( S.find("Reza") , S.find('\t' , S.find("Reza") )-S.find("Reza")+1 );

cout << S << \_ << endl;

S.erase( S.length()-1 , 1);

S += "\tLale\n";

cout << S << \_ << endl;

S.replace( S.find("Shahla") , S.find('\t' , S.find("Shahla"))-S.find("Shahla") , "Shohreh");

cout << S << \_ << endl;

for(int i=0; i<8; i++){

str[i] = S.substr(0 , S.find('\t'));

S.erase(0 , S.find('\t')+1 );

cout << str[i] << endl;

}

str[8] = S.substr(0 , S.find('\n'));

cout << str[8] << endl << \_ << endl;

for(int i=0; i<8; i++)

for(int j=0; j<8-i; j++)

if(str[j] > str[j+1]){

str0 = str[j];

str[j] = str[j+1];

str[j+1]= str0;

}

for(int i=0; i<9; i++)

cout << str[i] << endl;

cout << \_;

return 0;

}

//programmed by B.Shayegh.

//program 37:

#include<iostream>  
#include<string>  
using namespace std;

int counter (string txt) {  
 char x; int i=0,count=0;  
 cout<<" which character you are looking for?";  
 cin>>x;  
 while(i < txt.size()) {  
 if (txt[i]==x) count++;  
 i++;

}  
 return count;

}

int main() {  
 string text="";  
 cout<<"input your text here: ";  
 getline(cin, text);  
 cout<<counter(text);  
 return 0;

}

//programmed by M.Verkiani.

//program 38:

#include<iostream>  
using namespace std;

int factorial(int n)  
{  
 if(n > 1)  
 return n \* factorial(n - 1);  
 else  
 return 1;

}

int main() {

int n;  
 cout<<"input n here: ";  
 cin >> n;  
 cout << factorial (n);  
 return 0;

}

//programmed by M.Verkiani.

//program 39:

#include <iostream>  
using namespace std;

int checkPrimeNumber(int n){  
 bool flag = true;  
 for(int j = 2; j <= n/2; ++j){  
 if (n%j == 0){  
 flag = false;  
 break;

}  
 }  
 return flag;

{

int main(){  
 int n1, n2;  
 bool flag;  
 cout << "Enter two positive integers: ";  
 cin >> n1 >> n2;  
 cout << "Prime numbers between " << n1 << " and " << n2 << " are: ";  
 for(int i = n1+1; i < n2; ++i)  
 { // If i is a prime number, flag will be equal to 1  
 flag = checkPrimeNumber(i);  
 if(flag)  
 cout << i << " ";

}

return 0;}

//programmed by M.Verkiani.

//program 40:

#include<iostream>  
using namespace std;

int factorial (int n) {

double sum=1;

for (int i=1; i<=n; i++) {

sum\*=i;

}

return sum;

}

int main() {

int n;  
 cout<<"input n here: ";  
 cin >> n;  
 cout << factorial (n);  
 return 0;

}

//programmed by M.Verkiani.

//program 41:

#include<iostream>  
using namespace std;

int fib(int n)

{

if (n==1||n==2)

return 1;

else

return fib(n-1) + fib(n-2);

}

int main() {

for (unsigned i = 0; i < 20; i++) \_

cout << “fib(“ << I << “) = “ << fib(i) << endl;

}

//programmed by M.Verkiani.

//program 42:

#include <iostream>

using namespace std;

void print\_board(int a[]){

for(int i=0;i<8;i++){

for(int j=0;j<a[i];j++)

cout<<" ";

cout<<'R'<<endl;

}

cout<<"------------------------------------"<<endl;

}

void f(int a[],int size){

if(size==8){

print\_board(a);

return;

}

for(int i=0;i<8;i++){

bool tekrari=false;

for(int k=0;k<size;k++)

if(i==a[k])

tekrari=true;

if(tekrari)

continue;

a[size]=i;

f(a,size+1);

}

}

int mai(){

int \*a=new int[8];

f(a,0);

return 0;

}

//programmed by B.Shayegh.

//program 43:

#include <iostream>

#include <fstream>

#include <sstream>

using namespace std;

int main () {

string address;

getline (cin, address);

ifstream file;

file.open (address);

char temp;

while (file>>temp) {

if(int(temp)>=65 && int(temp)<=91) {

temp=temp-32;

cout<<temp;

}

else cout<<temp;

}

return 0;

}

//programmed by M.Verkiani.

//program 44:

#include <iostream>

#include <fstream>

#include <sstream>

using namespace std;

int main(){

string address;

getline(cin,address);

ifstream file;

file.open(address);

file.seekg(0, ios::end);

if (file){

if (file.tellg() == 0) {

cout<<'0';

}

else cout<<'1';

}

else cout<<"file not found...";

return 0;

}

//programmed by M.Verkiani.