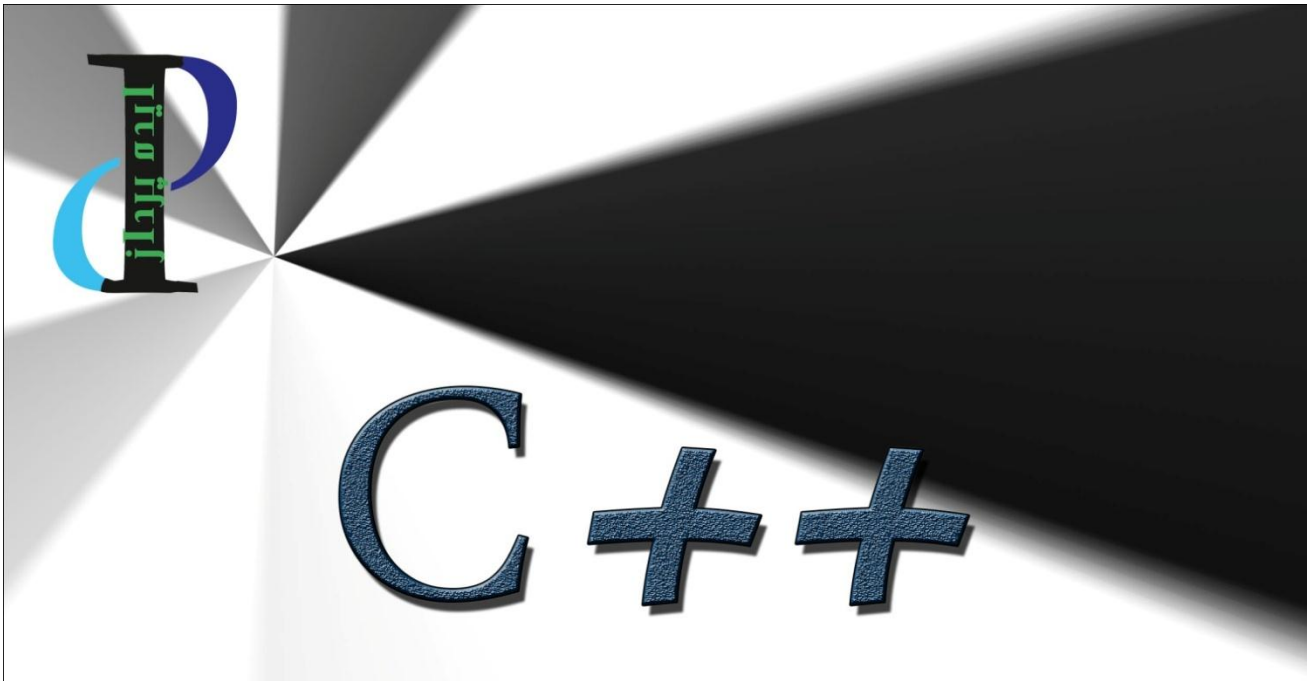


آموزش برنامه نویسی بدون ترس

ایده پرداز

- ☑ آموزش در کوتاهترین زمان
- ☑ آموزش کاربردی برنامه نویسی

مولف:زهرا بیات

فهرست مطالب

عنوان.....صفحه

تقدیم به: ۵

با تشکر: ۵

مقدمه: ۶

فصل اول اساس برنامه نویسی..... ۸

اساس برنامه نویسی ۹

انواع خطا در برنامه نویسی ۱۲

خطای متنی: ۱۲

خطای منطقی ۱۵

خطا در عبارت متنی..... ۱۵

توضیحات چیست؟..... ۱۶

تعریف متغیر: ۱۷

چار چوب C++ ۱۷

فصل دوم حلقه ها..... ۲۴

حلقه ها ۲۵

حافظه..... ۲۶

متغیرجمع ۲۷

متغیر ضرب ۲۷

حلقه های تو در تو ۲۹
حلقه تودر توی وابسته ۳۶
فصل سوم آرایه ها..... ۴۱
آرایه..... ۴۲
تعریف آرایه در C++ ۴۳
جمع آرایه متناظر ۵۳
فلگ (FLAG)چیست؟ ۵۶
چند نکته در مورد فلگ: ۵۷
آرایه ۲ بعدی و ۳بعدی:..... ۵۸
ماتریس خلوت: ۶۱
فصل چهارم کلاس ۶۴
حلقه شرطی(while): ۶۴
کلاس چیست؟ ۶۶
نحوه نوشتن دستورات داخل توابع:..... ۶۸
نحوه استفاده در بدنه: ۶۸
اشاره گرچیست؟..... ۷۵
تعریف اشاره گر:..... ۷۵
تابع سازنده چیست؟ ۷۶
فصل پنجم نوشتن و خواندن در فایل ۷۹
نوشتن و خواندن در فایل: ۸۰

نکاتی در مورد فایل ۸۰
در دستور fopen: ۸۲
خواندن از فایل : ۸۳

تقدیم به:

تقدیم به استاد مرحومم،جناب آقای مسعود شکری پور.

کسی که فکر و ذهنم را با ایده های جدیدش پرورش داد.

و راه رسیدن به تعالی را به من آموخت.امیدوارم این کتاب

روحش را شاد کند.

با تشکر:

با تشکر از همسر مهربانم که مرا برای نوشتن این کتاب

یاری داد . و بعد از خدای بزرگ و مهربان دومین حامی من بود.

هر چند که تشکر از او کمترین کار است.

مقدمه:

مهمترین مشکل دانش آموزان و دانشجویان تو رشته کامپیوتر درس برنامه نویسیه و همیشه به این درس مثل یه غول بی شاخ و دوم نگاه می‌کنن و فکر می‌کنن هیچ وقت این مسئله حل نمیشه ،شاید حتما باید مخ کامپیوتر باشن تا بتونن این درس و پاس کنن ، خیلی موقع‌ها قبل از امتحان دست به دامن این و اون میشن که قبل امتحان یه نفر پیدا بشه و معجزه کنه یه دفعه همه چیز مثل برق بره تو مخشون .بعد از امتحانم دست به دامن استاد میشن که یه نمره ای به ما بده تا این درسو پاس کنیم ، اما نمی دونن عاقبت این کار جز پشیمونی تو آینده چیزیو به ارمغان نمیاره . منم یکی مثل شما بودم که تو دبیرستان با حفظ کردن برنامه رفتم سر جلسه وآرزو می‌کردم غیر سوالای کتاب چیزی طرح نشده باشه ،اما وقتی وارد دانشگاه شدم تو اولین روز با کسی آشنا شدم که دید منو نسبت به برنامه نویسی عوض کرد و تموم فکر و ذکر منو با برنامه نویسی ساخت ، از اون روز به بعد من عاشق برنامه نویسی شدم و تو طول تحصیلی انقد پیشرفت کردم که استاد منو با خودش به شرکت برنامه نویسیش برد و من شدم یه برنامه نویس ...

خدا استاد مرحوممو بیامرزه ، من فقط برای شادی روح اون این کتاب رو نوشتم و تقدیمش می‌کنم به خودش. از شمایی هم که وقت گذاشتید و این کتاب رو می خونید، می‌خوام برای شادی روحش یه فاتحه بفرستید.

تو این کتاب سعی شده همه چیز راحت و روان مورد بحث قرار بگیره.. و شما مثل آب خوردن ، مرحله به مرحله برنامه نویسی C++رو یاد بگیرید،البته یه چیزی بگم: شما فقط C++ یاد نمی‌گیرید بلکه تفکر برنامه نویسیتون عوض میشه...به قول استاد شما آب خوردنو یاد می‌گیرید ، دیگه مهم نیست با چی آب بخوردید.... اگه آماده اید بریم دنبال برنامه نویسی و قدم های اول رو برداریم.

یه خواهشی ازتون دارم قبل از دیدن حل مسئله خودتونم یکم فکر کنید.

فصل اول

اساس برنامه نویسی

هدف های رفتاری:

مقدمات برنامه نویسی را می گیرید.

تفکر برنامه نویسی شما عوض می شود .

تحلیل برنامه را می آموزید

خطا یابی برنامه را فرا می گیرید.

تعریف متغیر و چارچوب C++ را هم فرا می‌گیرید.

اساس برنامه نویسی

برای نوشتن هر برنامه ای باید دو کار انجام دهیم:

۱. شکستن برنامه و نوشتن کد

۲. خطا یابی

هنگام نوشتن برنامه باید به موارد های زیر دقت کنیم:

۱. اول باید **مطلب را درک کنیم** و بفهمیم.

۲. مسئله رو **تحلیل** و به **قطعات کوچک بشکنیم.**

۳. مرحله آخر حل مسئله و **نوشتن کدهای برنامه نویسی**.

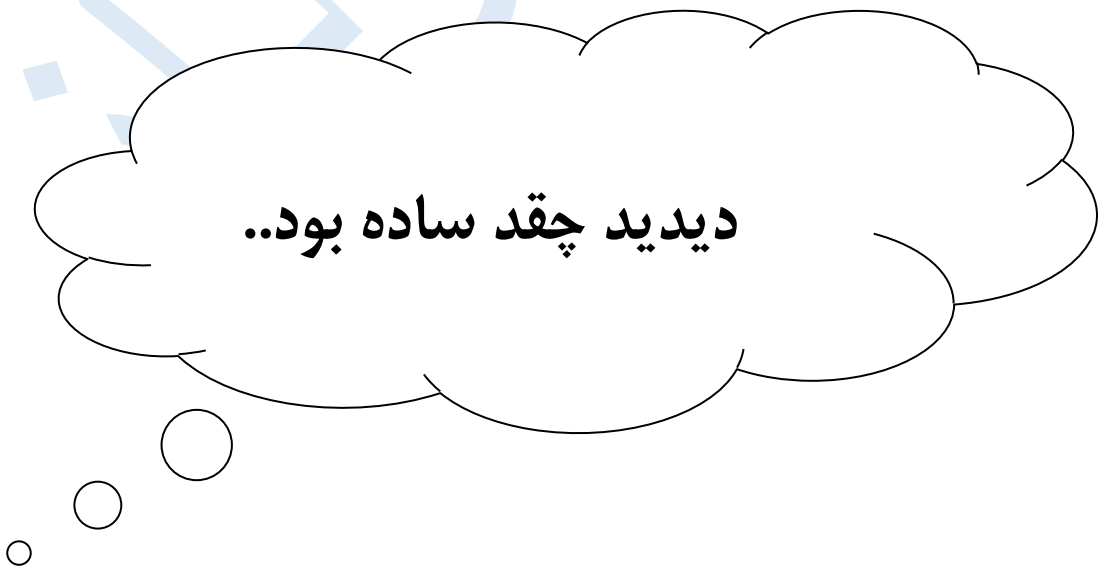
یه مثال ساده:جمع دو عدد

برای جمع دو عدد ابتدا باید **دو ظرف** حاضر کنیم، تا اعـداد داخـل آن هـا قـرار گیرند.

دوم باید از کاربر خواهش کنیم **دو تا عدد** وارد کند.

سوم :اینجا دیگه کار خودتونه یعنی باید دوتا عددو **جمع** کنید.

چهارم: با یه پیغام محترمانه **نتیجه** رو به کاربر نشان دهید.



پس با ما همره باشید...

به دستوراتی که برای کاربر مورد استفاده قرار می گیرد IO می گویند.

هر برنامه ای که می‌نویسیم از یک قانون استفاده می کنـد ،فقـط از نظـر ظـاهر متفاوت است. به مثال های زیر توجه کنید تا اصل موضوع را متوجه شوید:

برنامه ای بنویسید که **Hello** را چاپ کند.

Visual basic

```
Print “Hello”
```

Pascal

```
Begin
  Write(“Hello”);
End
```

Web

```
<?
   response write “Hello”
%>
```

C

```
Printf(“Hello”)
```

C++

```
Cout<< “Hello”;
```

مطمئنم تا اینجا همه چیزرا یاد گرفتید،پس سراغ خطا یابای و روش های حل می رویم:

انواع خطا در برنامه نویسی

خطای متنی:

خطای متنی خطایست که، هنگام رخ دادن مانع اجرای برنامه می‌شود و رفع آن خیلی ساده است . فقط کافیست پیغام را بخوانی.

هر خطایی که در متن برنامه رخ می‌دهد یکی از حالات زیر را دارد:

```
x = 5;
```

خطا می‌گیرد چون باید قبل از مقدار دهی x تعریف شود:

```
int x;
x = 5;
```

خطا در نوع متغیر

```
int x;
x = 5.1;
```

مقدارx از نوع صحیح تعریف شده در حالی که مقدار اعشاری بـه آن نسـبت داده شده پس در اینجا دو نکته مهم وجود دارد:

تعریف متغیر

جنسیت دو طرف که باید یکسان باشد.

خطا در دستور

خطا در **شکل دستور** که مانع اجرا می‌شود مثل غلط املایی

خطا در **ساختار دستور**

```
Prnt  “h”
```

خطای املایی (i گذاشته نشده)

```
Cout<< ”xx”
```

خطای ساختاری در C++ .باید در آخر دستور ; گذاشته شود.

برای حل این مشکلات بهتر است روی کلمه اشتباه قرار بگیریـد و کلیـد F1 را بزنید. که در پاسکال باید کلید Ctrl را هم با آن گرفت.

```
Print[“hello”]
```

خطای ساختاری در VB

```
For  x:=1 to 5.5 do
End
```

خطای ساختاری در پاسکال. **اجازه استفاده از اعشار را نداریم.**

```
Print  x=5+6
```

خطا در ساختار

خطای منطقی

این دسته خطاها در اثر اشتباه برنامه نویس در طراحی الگوریتم درست، برای برنامه و یا گاهی در اثر درنظر نگرفتن بعضی شرایط خاص در برنامه ، ایجاد می‌شوند. متأسفانه این دسته خطاها در زمان کامپایل اعلام نمی‌شوند و در زمان اجرای برنامه ، خود را نشان می‌دهند. بنابراین، این خود برنامه‌نویس است که پس از نوشتن برنامه باید آن را تست کرده و خطاهای منطقی آن را پیدا و رفع نماید. متاسفانه ممکن است یک برنامه‌نویس خطای منطقی برنامه خود را تشخیص ندهد و این خطا پس از مدتها و تحت یک شرایط خاص توسط کاربر برنامه کشف شود.

مثلا برنامه ای برای تشخیص شماره ملی نوشته اید ، که ورودیش از نوع صحیح است . همان طور که می‌دانید در نوع صحیح تعداد صفر قبل عدد حساب نمی‌شود. پس در این صورت اگر کد ملی با صفر شروع شود عملیات مورد نظر روی آن انجام نمی شود.شاید با چند کد ملی متفاوت امتحان کنید و برنامه به ظاهر مشکلی نداشته باشد ، اما در یک مورد خاص مانند بالا با دردسر روبرو شوید.

خطا در عبارت متنی

```
"Hello"=5
```

در مثال بالا نمی توان داخل عبارت متنی عددی قرار داد.

توضیحات چیست؟

عباراتی که برای فهم مسئله نوشته و فقط جنبه نمایشی دارند و همچنین در کد نویسی هیچ تاثیری ندارند را توضیحات می‌گویند.البته در هر زبان برنامه نویسی متفاوت می‌باشد.

C /* */

Vb '

pascal { }

دیگه توضیحات کافیه بهتر بریم رو برنامه نویسی.

قبل از حل مسئله باید مسئله را به :

۱. ورودی

۲. خروجی

۳. شرط

۴. حلقه

شکست.

تعریف متغیر:

متغیر ظرفیست که در آن اطلاعات قرار می‌گیرد . و همان طور که می‌دانید چون ما برای هر غذا از ظرف مخصوص استفاده می‌کنیم برای انواع متغیر ها هم باید از ظرف مخصوص خودش استفاده کنیم ، مثلا اگر عدد صحیح استفاده می‌کنیم از نوع int اگر اعشاری ،Float و...

هر گاه پارامتری را نوشتیم و خط زدیم یعنی نیاز به متغیر داریم.

برای نوشتن برنامه در C++ قبل از هر کاری باید از هدر هایی استفاده کنیم. هدر ها مجموعه بسته هایست که درون آن ها دستورات برنامه نویسی است. نگران نباشید این هدر ها آنقدر تکرار می‌شود که نیازی به حفظ کردن آن ندارید.

چار چوب C++

```
#include <iostream.h>
Void  main()
{
}
```

هدر

سوال:

برنامه بنویسید **شعاع را دریافت** و **مساحت** و **محیط دایره** را محاسبه کند.

جواب :

ابتدا کاربر باید **عددی** را به عنوان **ورودی** وارد کند (ورودی).

سپس **فرمول مساحت و محیط** را محاسبه و چاپ کنیم(خروجی).

پس **متغیری** برای شعاع نیاز داریم.

دستور **دریافت ورودی** cin>>

ودستور **چاپ** cout << می باشد.

برنامه

```
#include<iostream.h>
void main()
{
    int x;  //تعریف متغیر برای دریافت عدد

    cout<<"masahat ra vared kon:" ;//چاپ پیغام ورود مساحت
    cin>> x;     //دریافت ورودی وریختن در متغیر

    cout<<"masahat" ;          //چاپ پیغام مساحت
    cout<< x*x*3.14;           //چاپ مساحت

    cout<<"mohit" ;            //چاپ پیغام محیط
    cout<< x+x*3.14;           // چاپ محیط
}
```

برنامه ای بنویسید که **دو عدد** را بخواند بعد **جابه جا** کند.

راه حل اول:

ابتدا نیاز به سه ظرف داریم ، ظرف A , B , C

ظرف A , B حاوی عدد و ظرف C به عنوان واسط است.

اول باید ظرف A را در C بریزیم.

بعد ظرف B را در A

و در آخرظرف C را در B

حالا اعداد جا به جا شد.

A=10 **B=5**

1) C=A ⟶ C=10

2) A=B ⟶ A=5

3) B=C ⟶ B=10

B=10 **A=5**

راه حل دوم

ابتدا دو ظرف نیاز داریم .باید در ظرف اول ،ظرف دوم را هم اضافه کردیعنی:

x=x+y

سپس در ظرف دوم ،ظرف اول را از ظرف دوم کم کرد.

y=x-y

و در آخر در ظرف اول ظرف اول را از ظرف دوم کم کرد.

x=x-y

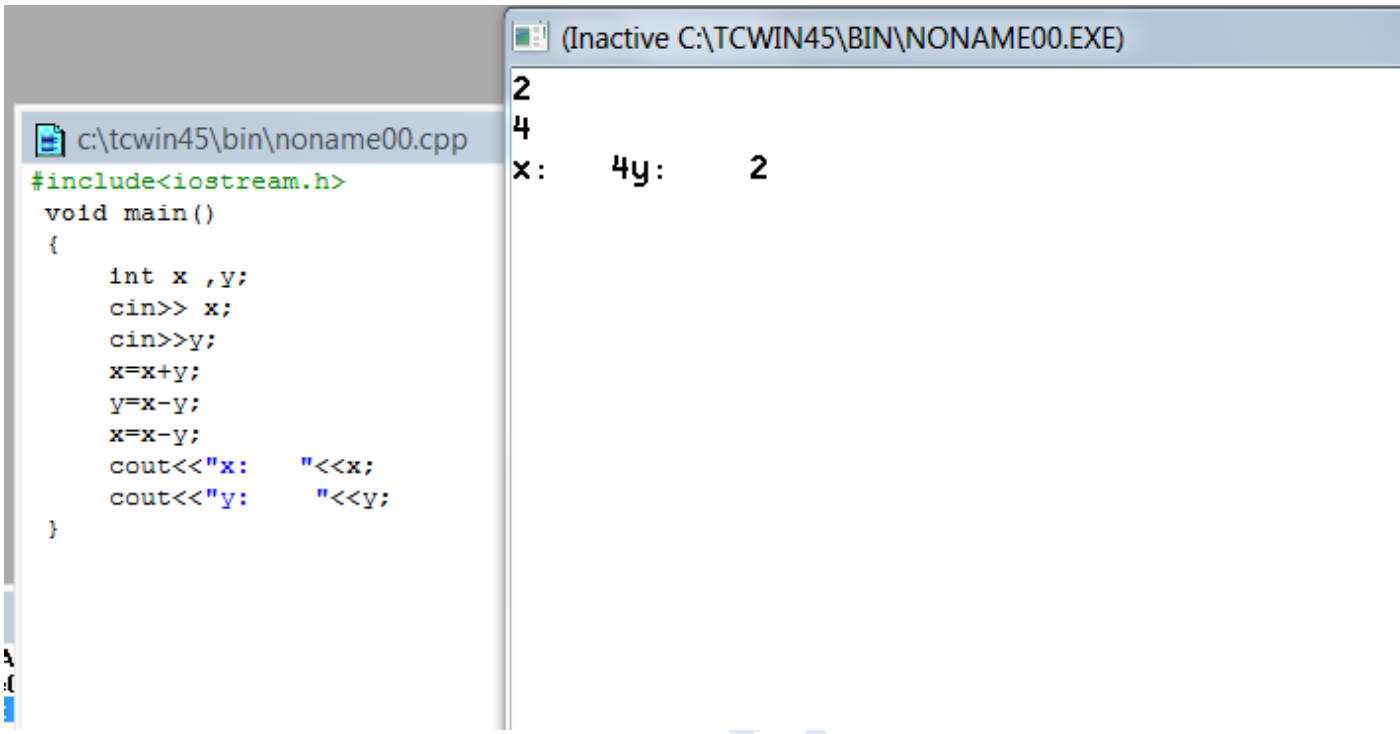
x=2 **y=4**

x=x+y x=2+4 6 ⟶ x=6 y=4

y=x-y y=6-4 2 ⟶ x=6 y=2

x=x-y x=6-2 = 4 ⟶ x=4 y=2

x=4 **y=2**



برنامه بنویسید دو عدد دریافت هر کدام **بزرگ تر** بود آن را چاپ کند.

دریافت دو عدد

مقایسه دو عدد

چاپ بزرگترین

پس نیاز به **شرط** داریم.

```
If (شرط)
Else
{
}
```

برنامه:

```
#include<iostream.h>
void main()
{
    int x;   //تعریف متغیر برای دریافت عدد
    int y;   //تعریف متغیر برای دریافت عدد

    cout<<"adad aval ra vared kon:   "//چاپ ورود اولین عدد
    cin>>x;  //دریافت اولین عدد

    cout<<"adad aval ra vared kon:   "//چاپ ورود دومین عدد
    cin>>y; // دریافت دومین عدد

    if(x>y)  //شرط بزرگ بودن
    cout<<"x: "<<x;  //اگر عدد اول بزرگ بود آن را چاپ کند
    else              // وگرنه
    {
        cout<<"y:   "<<y;  //عدد دوم راچاپ کند
    }

}
```

فصل دوم

حلقه ها

هدف های رفتاری:

- ✓ تعریف حلقه را فرا می گیرید.
- ✓ متغیر جمع ،ضرب و حافظه را یاد می گیرید.

در پایان هم با حلقه های وابسته و تو در تو آشنا می‌شوید.

حلقه ها

هرگاه یک فرآیند تکراری با روند مشخص در مسائله باشد ، از حلقه استفاده می‌کنیم.

در حلقه باید به سه سوال جواب داد:

۱- **از کجا**

۲- **تا کجا**

۳- **چند تا چند تا**

```
For(چند تا چند تا;تا کجا ;از کجا)
For(int i=0 ;i<=100;i++)
```

برنامه ای بنویسید که **اعداد زوج** بین ۱ تا ۱۰۰ را چاپ کند.

این حلقه ،یک حلقه ی ساده است ، که در آن باید از **شرط** استفاده کنیم.

اگر i زوج باشد، چاپ زوج و چاپ عدد i وگرنه فرد و چاپ عدد i .

```
#include<iostream.h>

void main()
{
    int i;   //تعریف متغیر

    for(i=1;i<=100;i++)  //حلقه ۱ تا ۱۰۰
    {
        if(i%2==0)   //بررسی زوج بودن
            cout<<i<<"zoj";   // در صورت برقراری چاپ اندیس و کلمه زوج
        else //وگرنه
        {
            cout<<i<<"fard";//در صورت برقراری چاپ اندیس و کلمه فرد
        }
    }
}
```

نکته: اگر بعد از متغیر = بگذاریم به معنای انتساب است . یعنی آن آیتم(مثلا عدد) را درون آن متغیر می گذاریم. اما برای مقایسه باید از == استفاده کنیم .

توجه!: داخل حلقه شمارنده را تغییر ندهید.

حافظه

مکانی که اعداد ، رشته و... درون آن قرار می گیرد.

متغیرجمع

متغیری که در یک مکان حافظه قرار می‌گیرد ،قبل از حلقه صفر، داخل حلقه اضافه و خارج حلقه استفاده می‌شود.

متغیر ضرب

متغیری که در یک مکان حافظه قرار می‌گیرد ،قبل از حلقه یک،داخل حلقه اضافه و خارج حلقه استفاده می‌شود.

هر وقت در صورت مسئله **جمع** داشتیم ،به **متغیرجمع** و هر وقت **ضرب** داشتیم به **متغیرضرب** نیازمندیم.

برنامه ای بنویسید که **۱۰ عدد از کاربر دریافت** کند اگر **عدد زوج** بود آن را **جمع** کند و در پایان **جمع کل** را نمایش دهد.

کار تکراریست(دریافت ۱۰ عدد)⟵ **حلقه**

زوج بودن عدد ⟵ **شرط**

جمع اعدا زوج ⟵ متغیر جمع

چون در این حلقه شرط و دریافت ورودی وجود دارد باید دستورات داخل آن بین **دو آکولاد** قرار گیرد.

حالا شما این چند نمونه سوال را حل کنید...

۱-برنامه بنویسید که **معادله** **ax+bx=c** را چاپ کند.

راهنمایی:در قیقت معادله بالا بعد از حل شدن به صورت **x=b/a** است .در اینجا **سه متغیر** داریم که **b,a** را کاربر وارد می‌کند ,مجموع آن ها در حافظه جمع **x** قرار می‌گیرد.

۲-برنامه ای بنویسید که **دو عدد** را خوانده **بزرگترین** آن را چاپ کند.

راهنمایی: نیاز به **سه متغیر** **x,y,z** داریم .باید با یک **شرط** مشخص کنیم **x** **بزرگ** است یا **y** ،هر کدام بزرگ بود در **z** قرار گیرد و نمایش یابد.

۳-برنامه ای بنویسید که **اعداد فرد** ۱ تا ۱۰۰ را جمع کند و **میانگین** آن را نمایش دهد.

راهنمایی: ابتدا :نیاز به **حلقه** داریم : از کجا:از1 ، تا کجا: تـا ۱۰۰ ، چنـدتا چنـد تا:دو تا دوتا،

دوم :نیاز به **متغیر جمع** داریم که در آن شمارنده حلقه را جمع کند

```
x=x+i
```

سوم: در خارج **حلقه** نیاز به متغیری برای **میانگین**

```
y=x/50
```

حلقه های تو در تو

هر گاه حرکت به صورت ماتریسی بود یعنی طول و عرض را نیاز داشتیم از حلقه تو در تو استفاده می‌کنیم.در هر کدام از مکان ها می توانیم دستور بنویسیم.

```
For(int i=0 ;i<=10; i++)
    {
    دستور
    For(int j=0 ;j<=10; j++)
        {
    دستور

        }
    دستور
    }
    دستور
```

برنامه ای بنویسید که **شکل** زیر را چاپ کند.

```
****
****
****
```

به **دو حلقه** نیاز داریم که داخل حلقه دوم **ستاره ها** چاپ و خارج حلقه دوم اینتر خورده می‌شود.

```
#include<iostream.h>
void main()
{
    for(int i=0 ;i<=3;i++)//حلقه ای ۴ تایی برای چاپ ستاره
    {
        for(int j=0 ;j<=3;j++)//حلقه ۴ تایی داخلی برای چاپ ستاره
        {
            cout<<"*"  ;//چاپ ستاره
        }
        cout<<endl  ;//بعدا از یک دور چاپ باید اینتر زده شود.
    }
}
```

حل حلقه:

i	j	خروجی
۰	۰و۱و۲و۳	****
۱	۰و۱و۲و۳	****
۲	۰و۱و۲و۳	****
۳	۰و۱و۲و۳	****

برنامه ای بنویسید که شکل زیر را چاپ کند.

۱۱۱

۲۲۲

۳۳۳

حل حلقه:

خروجی	j	i
۱۱۱	۱و۲و۳	۱
۲۲۲	۱و۲و۳	۲
۳۳۳	۱و۲و۳	۳

از حل این حلقه نتیجه می‌گیریم که فقط i چاپ می‌شود

برنامه

```
#include<iostream.h>
void main()
{
    for(int i=1 ;i<=3;i++)   //حلقه ۴ تایی
    {
        for(int j=1;j<=3;j++)//حلقه ۴ تایی داخلی
        {
            cout<<i  ;   //چاپ اندیس
        }
        cout<<endl  ;    //زدن اینتر
    }
}
```

برنامه ای بنویسید که شکل زیر را چاپ کند.

۱۲۳

۲۴۶

۳۶۹

حل حلقه:

i	j	خروجی	
۱	۱و۲و۳	۱۲۳	i*j
۲	۱و۲و۳	۲۴۶	

۳۶۹ ۱و۲و۳ ۳

از حل این حلقه نتیجه میگیریم که **j*i** شده است.

برنامه

```
#include<iostream.h>
void main()
{
    for(int i=1 ;i<=3;i++)   //حلقه ۴ تایی
    {
        for(int j=1 ;j<=3;j++)   //حلقه ۴ تایی داخلی
        {
            cout<<i*j  ;  //ضرب دو اندیس حلقه ها
        }
        cout<<endl  ;  //زدن اینتر
    }
}
```

حالابرنامه های زیر شما بنویسید.

خروجی زیر را چاپ کنید.

۱)

۱۰۳

۰۰۰

۳۰۹

راهنمایی:

حل حلقه:

خروجی	j	i
۱۰۳	۱و۲و۳	۱
۰۰۰	۱و۲و۳	۲
۳۰۹	۱و۲و۳	۳

در نتیجه اگر i=2 یا j=2 بود باید عدد صفر چاپ شودوگرنه باید .

```
If(i==2 || j==2)
```

۲)

100

120

123

حل حلقه:

خروجی	j	i
100	۱و۲و۳	۱

120	۱و۲و۳	۲
123	۱و۲و۳	۳

اگر i<j عدد صفر وگرنه j را چاپ کند.

حلقه تودر توی وابسته

هرگاه شکل ماتریسی متقارن نبود ،یا اصطلاحا هـم i و هـم j را نیـاز داشـتیم ماتریس ما تو در توست.

برنامه ای بنویسید که شکل زیر چاپ کند.

```
*
**
***
```

حل حلقه:

خروجی	j	i
*	۱	۱
**	۱و۲	۲

۳ ۱و۲و۳ ***

```
#include<iostream.h>
void main()
{
    for(int i=1 ;i<=3;i++)   //حلقه ۴ تایی
    {
        for(int j=1 ;j<=i;j++)   //حلقه ۴ تایی داخلی
        {
            cout<<"*"  ;          //چاپ ستاره
        }
    cout<<endl  ;         //زدن اینتر
    }
}
```

برنامه های زیر را شما بنویسید.

۱)

۱

۱۲

۱۲۳

حل حلقه:

خروجی j i

	1	۱	۱
	۱۲	۱۲	۲
	۱۲۳	۱۲۳	۳

حلقه وابسته مثل بالا که باید j چاپ شود.

۲)

۱

۲۲

۳۳۳

حل حلقه:

	خروجی	j	i
۱			
		۱	۱
	۲۲	۱۲	۲
	۳۳۳	۱۲۳	۳

حلقه وابسته مثل بالا که باید i چاپ شود.

۳)

۱

۲۴

۳۶۹

حل حلقه:

خروجی	j	i
۱	۱	۱
24	۱۲	۲
369	۱۲۳	۳

حلقه وابسته مثل بالا که باید i*j چاپ شود.

۴)

۱

**

۳۳۳

*****

حل حلقه

خروجی	j	I
۱		
1	1	۱

۲	۱و۲	**	
۳	۱و۲و۳	333	
۴	۱و۲و۳و۴	****	

حلقه وابسته مثل بالا که بایددر آن شـرط زوج بـودن i بررسـی شـود و در ایـن صورت *چاپ شود.

فصل سوم

آرایه ها

هدف های رفتاری:

- ✓ تعریف آرایه و نحوه استفاده از آن را یاد می گیرید.
- ✓ با انواع آرایه های دو بعدی و سه بعدی آشنا می‌شوید.
- ✓ مفهوم ماتریس خلوت و فلگ را استفاده می کنید.

آرایه

هر گاه به تعداد زیادی متغیر از یک جنس نیـاز داشـته باشـیم از آرایـه اسـتفاده می‌کنیم. فکر کنید نیاز به گرفتن نمره ۵ دانش آموز دارید ، در این صورت مجبورید ۵ متغیر تعریف کنید. حالا اگر این ۵ دانش آموز به ۵۰ دانـش آمـوز تغیـر کنـد بـه دردسر جدیدی برمی‌خورید دیگر نمی توانید ۵ متغیر تعریف کنید پس بهتر است به سراغ یک راه حل منطقی برویم. آرایه مانند یک بسته می باشد که هـر چقـد نیـاز دارید تعریف می‌کنید.

پس:

آرایه فضای ذخیره سازی است که در آن باید سه عمل انجام شود:

۱- ورود اطلاعات

۲- پیمایش

۳- چاپ

آرایه از صفر شروع می شود و دسترسی به اطلاعات آن با ایندکس(**شمارنده حلقه**) می‌باشد.

X(1)	x(2)	x(3)	x(4)	x(5)
10	20	30	40	50

تعریف آرایه در C++

مثلا: int x[5]

در++c آرایه از صفر شروع می شود .آرایه بالا ۶ خانه را شامل می‌شود .

در نوشتن آرایه به حلقه نیازمندیم چون باید بین خانه های آرایه حرکت کنیم و به شماره خانه ها دسترسی داشته باشیم .که یک بار باید بنویسیم ویک بار بخوانیم.

برنامه ای بنویسید **۵ عدد را بخواند** سپس چاپ کند.

```
#include<iostream.h>
void main()
{
    int x[4];
    for(int i=0 ;i<=4;i++)
    {
        cin>>x[i];
    }
    for( i=0 ;i<=4;i++)
    {
    cout<< x[i];
    }
}
```

برنامه ای بنویسید که **خانه های زوج آرایه** را مقدار دهی کند.آرایه ۵ تایی می‌باشد.

در این سوال فقط به خانه های زوج نیازمندیم پس باید شرط گذاشته شود .

```
#include<iostream.h>
void main()
{
    int x[4];    //تعریف آرایه
    for(int i=0 ;i<=4;i++)   //حلقه ۵ تایی
    {
        if(i%2==0)        //بررسی زوج بودن شمارنده حلقه
         cin>>x[i];          //در صورت زوج بودن خانه آرایه ای که با اندیس زوج برابر است را مقداردهی می کنیم//
    }
    for( i=0 ;i<=4;i++)  //حلقه ۵ تایی
    {
        if(i%2==0)   //بررسی زوج بودن شمارنده حلقه
        cout<< x[i];//چاپ خانه های زوج مقداردهی شده
    }
}
```

برنامه ای بنویسید که ۵ عدد را **خوانده** و **جمع و میانگین** آن را چاپ کند.

نکته این سوال **میانگین** است . میانگین نباید داخل **حلقه** باشد بلکه در پایان حلقه محاسبه می‌شود چون باید جمع کل به دست آید و درآخر میانگین محاسبه شود.

```
#include<iostream.h>
void main()
{
    int x[4];     //تعریف آرایه ۵ تایی
    int y=0;      //تعریف متغیر برای میانگین
    for(int i=0 ;i<=4;i++)   //حلقه ۵ تایی
    {
        cin>>x[i];   //پر کردن آرایه
        y=y+x[i];    //جمع کل خانه های آرایه
    }
    cout<<y/5;       //چاپ میانگین
}
```

حالایه ذره سوالات رو سخت تر میشه...

نمرات ۵ دانش آموز را دریافت و

۱- کمترین نمره

۲- بیشترین نمره۳

۳- نمرات کمتر از ۱۰

۴- میانگین نمرات

در این سوال باید سه کار انجام شود:

۱-مقدار دهی اولیه

۲-مقایسه

۳- استفاده

در سوال ۱و۲ نیاز به متغیری داریم که اولین نمره در آن قرار گیرد و آن متغیر با خانه های بعدی مقایسه اگر کمتر یا بیشتر بود با جایگزین شود.

مثال: شما اعداد روبرو را وارد می‌کنید.

X[0]	X[1]	X[2]	X[3]	X[4]
4	2	5	9	1

به طور پیش فرض c=0 است.بعد از پر شدن حلقه c اولین خانه آرایه می‌شود.

c با تک تک خانه ها **مقایسه** می‌شود.

i	x[i]	c	out low
0	4	4	4
1	2	4	2
3	5	2	2
4	9	2	2
5	1	2	1

نتیجه ۱ می‌شود. برای بزرگتر هم همین کار را می‌کنیم.

```
#include<iostream.h>
void main()
{
     int x[4];     //تعریف آرایه ۵ تایی
     int  min;     //تعریف متغیر
     min=0;
     for(int i=0 ;i<=4;i++)          //حلقه ۵ تایی
     {
          cin>>x[i];          //پر کردن آرایه
     }
       min=x[0];      ریختن اولین خانه آرایه در یک متغیر به منظور
                                                          //مقایسه
     for( i=0 ;i<=4;i++)//حلقه ۵ تایی
     {
          if(min>x[i])      // بررسی بزرگتر بودن متغیر
          min=x[i];          اگر متغیر بزرگ باشد یعنی خانه آرایه
                                کوچک است پس باید متغیر تغیر کند.//
     }
     cout << min;               چاپ متغیر که الان کوچک ترین عدد
//است
}
```

نکته:x[i] بیرون حلقه مفهومی ندارد.چون i در حلقه استفاده می‌شود.

قسمت۲:

```
#include<iostream.h>
void main()
{
     int x[4];     //تعریف آرایه ۵ تایی
     int max;      //تعریف متغیر
     for(int i=0 ;i<=4;i++)
     {
          cin>>x[i];   //پر کردن آرایه
     }
       max=x[0];      ریختن اولین خانه آرایه در یک متغیر به منظور
                      مقایسه//
     for( i=0 ;i<=4;i++)       //حلقه ۵ تایی
     {
          if(max<x[i])        اگر متغیر کوچک باشد یعنی خانه آرایه
                              بزرگ است پس باید متغیر تغیر کند.//
           max=x[i];          ریختن اولین خانه آرایه در یک متغیر
                              به منظور مقایسه//
     }
     cout << max;          //چاپ متغیر که الان بزرگ ترین عدد
}
```

قسمت ۳:

```
#include<iostream.h>
void main()
{
    int x[4];    //تعریف آرایه ۵ تایی
    for(int i=0 ;i<=4;i++)   //حلقه ۵ تایی
    {
        cin>>x[i];        //پر کردن آرایه
    }
    for( i=0 ;i<=4;i++)  //حلقه ۵ تایی
    {
        if(x[i]<10) // بررسی کوچکتر از ۱۰ بودن خانه آرایه
        cout <<x[i];      اگر خانه آرایه کوچکتر از ۱۰ بودچاپ
                          می شود.//
    }
}
```

قسمت۴:

```
#include<iostream.h>
void main()
{
    int x[4];    //تعریف آرایه ۵ تایی
    int c;   //تعریف متغیر
    for(int i=0 ;i<=4;i++)   //حلقه ۵ تایی
    {
        cin>>x[i];   //پر کردن آرایه
    }
    c=0;     //متغیر را صفر می کنیم(متغیر جمع)
    for( i=0 ;i<=4;i++)       //حلقه ۵ تایی
    {
        c=c+x[i];    //جمع خانه های آرایه
    }
    cout << c/5;     //میانگین خانه های آرایه
}
```

حالا نوبت شماست...

۱)آرایه ای ۵ تایی را **مقدار دهی** کنید ،سپس **یک عدد را بخوانید** ، و برنامه اعلام کند عدد در کدام خانه قرار گرفته.

در این سوال ابتدا در یک حلقه ۵ تایی یک آرایه ۵ تایی را مقدار دهی می کنید، سپس خارج حلقه یک عدد را از کاربر می گیرید . دوباره در یک حلقه دیگر آن متغیر دارای عدد را با تک تک خانه ها بررسی می‌کنید . اگر شرط برقرار بودشماره i را چاپ کند.

شرط:

```
if (c=x[i])
cout<< i;
```

۲) برنامه ای بنویسید که آرایه ۵ تایی را مقدار دهی و سپس اعداد زوج لیست را برابر صفر کند.

این سوال هم در حلقه دوم بایـد شـرط زوج بـودن را بررسـی و آن خانـه را در صورت زوج بودن صفر می‌کنیم.

```
If(x[i]%2=0)
x[i]=0;
```

۴) برنامه ای بنویسید که در یک آرایه ۵ تایی اگر محتوا بـا انـدیس برابـر بـود . محتوا را چاپ کند.

شرط

```
if(x[i]=i)
cout<<x[i];
```

۴) در همان آرایه ۵ تایی عددی از کاربر بگیرد در صورت تکراری بـودن تعـداد ، آن را مشخص کند.

بعد از پر کردن آرایه در خارج حلقه عددی از کـاربر دریافـت مـی‌کنید.سـپس در حلقه بعد با شرطی آن را با خانه های آرایه مقایسه می‌کنید ،اگر برابر بـود بـه یـک شمارنده اضافه می‌کنید.

```
if(c= x[i])
m=m+1;
```

۵) برنامه ای بنویسی که تعدا خانه های زوج و خانه های فـرد در یـک آرایـه ۵ تایی را چاپ کند.

دقیقا مانند مثال قبلیست.

جمع آرایه متناظر

اگه دو تا آرا یه را پر کنیم و نیاز به جمع خانه ها باشد ، باید چه ترفنـدی را بـه کار ببریم به مثال زیر دقت کنید.

شماره خانه	1	2	3	4	5
آرایه x	10	20	50	20	10
آرایه y	7	10	20	30	4
	17	30	70	50	14

برای حل این سوال باید توسط یک حلقه دو آرایه را پر کنیم و توسط حلقه دیگر و یک متغیر آن ها را جمع و نمایش دهیم.

```
#include<iostream.h>
void main()
{
    int x[4];     //تعریف آرایه ۴ تایی
    int y[4];     //تعریف آرایه ۴ تایی
    int  c;       //تعریف متغیر برای جمع دو آرایه

for(int i=0 ;i<=4;i++)    //حلقه ۵ تایی
    {
        cin>>x[i];        //پر کردن آرایه اول
        cin>>y[i];        //پر کردن آرایه دوم
    }
    c=0;          //متغیر جمع که قبل از حلقه باید صفر باشد.

for(int i=0 ;i<=4;i++)        //حلقه ۵ تایی
    {
        c=y[i]+x[i];      //جمع دو آرایه در متغیر جمع
        cout <<c;         //جمع دو آرایه
    }
}
```

نکته:اگر جمع آرایه را خارج حلقه می گذاشتیم ،جمع تمام خانه دو آرایه را می‌دیدیم.در حالی که ما قصد دیدن جمع خانه های متناظر را داریم.

معکوس کردن آرایه در یک آرایه دیگر:

در این برنامه باید آرایه ای دریافت کنیم و در آرایه دیگر آن را معکوس کنیم.

X[]	3	2	1
Y[]	1	2	3

حل:

I	x[i]	y[i]
0	3	1
1	2	2
2	1	3

در حقیقت:

Y[0]=x[2]

Y[1]=x[1]

Y[2]=x[0]

I			
0	2-0=2	y[0]	x[2]
1	2-1=1	y[1]	x[1]
2	2-2=0	y[2]	x[0]

پس برای دانستن خانه X باید از یک مقدار ثابت کم شود .که این مقدار ثابت آخرین خانه آرایه است.

برنامه ای بنویسید آرایه ۵ تایی را مقدار دهی و در آرایه دیگر معکوس کند.

```
#include<iostream.h>
void main()
{
    int x[4];          تعریف آرایه ۴ تایی//
    int y[4];          تعریف آرایه ۴ تایی//
    for(int i=0 ;i<=4;i++)          حلقه ۵ تایی//
    {
        cin>>x[i];    پر کردن آرایه اول//
    }
    for(int i=0 ;i<=4;i++)    حلقه ۵ تایی//
    {
        y[i]=x[4-i];       معکوس کردن آرایه اول در دوم//
        cout <<y[i];       چاپ آرایه دوم//
    }
}
```

فلگ (FLAG)چیست؟

فلگ نشانه ایست که وضعیت موجود در داخل حلقه را به ما نشان می دهد .

مثال: فکر کنید می‌خواهید بدانید در یک آرا عدد زوج وجود دارد یا خیر .

در این صورت مجبورید یک متغیر را در خارج حلقه صفر کنید . در داخل حلقه بررسی زوج بودن را انجام دهید و در خارج حلقه آن متغیر را بررسی کنید و در صورت تغییر گزارش دهید.

```
#include<iostream.h>
void main()
{
     int x[4];     تعریف آرایه ۴ تایی//
     int f=0;      فلگ//
for(int j=0;j<=4;j++)     حلقه ۵ تایی//
          {
               Cin>>x[i];    // پر کردن آرایه
          }

     for(int i=0;i<=4;i++)     حلقه ۵ تایی//
          {
               if(x[i]%2=0)  بررسی زوج بودن خانه آرایه//
                    f=1       در صورت برقراری شرط فلگ ۱ می شود//
          }
     if (f==1)          بررسی فلگ//
     cout<<"ok";        در صورت برقرای شرط چاپ متن//

}
```

چند نکته در مورد فلگ:

۱-فلگ کنترل کننده وضعیت است. در برنامه هایی که خطوط مرزی دارند وضعیت دوحالته است پس نیاز به فلگ داریم.

۲-فلگ را هیچ وقت با else استفاده نکنید.

۳-از فلگ برای شرط نقض استفاده می شود.

۴- فلگ را باید بعد از حلقه و با یک شرط بررسی کنید.

بهتر است برای فلگ از عدد استفاده شود چون ممکن است رشته کوچک و بزرگ نوشته شود .

آرایه ۲ بعدی و ۳بعدی:

آرایه دوبعدی آرایه ایست که مانند حلقه های تو در تو ماتریسی عمل می کند. و آرایه سه بعدی علاوه بر طول و عرض دارای عمق نیز می‌باشد.

تعریف: `int x[3][3]` یا `int[2][3][5]`

برای پر کردن آرایه باید به تناسب آرایه از حلقه استفاده کنیم.مثال زیر پر کردن و چاپ کردن یک آرایه ۲*۲ می‌باشد.

```
#include<iostream.h>
void main()
 {
    int j;   //تعریف متغیر برای حلقه
    int i;   //تعریف متغیر برای حلقه
    int x[1][1];      //تعریف آرایه دو بعدی ۲*۲
    for(i=0;i<=1;i++)     //حلقه ۲ تایی خارجی
    {
        for(j=0;j<=1;j++)     //حلقه ۲ تایی داخلی
        {
             cin>>x[i][j];          //پر کردن آرایه ۲*۲
        }
     }
    for(i=0;i<=1;i++)     //حلقه ۲ تایی خارجی
    {
        for(j=0;j<=1;j++)     //حلقه ۲ تایی داخلی
            {
                cout<<x[i][j];    //چاپ آرایه ۲*۲
            }
    }
}
```

برنامه ای بنویسید که جمع هر سطر آرایه دو بعدی ۳*۳ را چاپ کند.

شماره خانه ها	۰	۱	۲	جمع
۰	۲	۵	۳	۱۰
۱	۹	۷	۵	۲۱
۲	۰	۱	۳	۴

شماره سطر	out	j	I
(0,0)	۲	0	0

(0,1)	۵	۱	۰
(0,2)	۳	۲	۰
(1,0)	۹	۰	۱
(1,,1)	۷	۱	۱
(1,2)	۵	۲	۱
(2,0)	۰	۰	۲
(2,1)	۱	۱	۲
(2,2)	۳	۲	۲

اگر توجه کنید در هر دوره i ثابت است.

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

```
void main()
 {
    int j;   //تعریف متغیر برای حلقه
    int i;   //تعریف متغیر برای حلقه
    int x[1][1]ک
    int c=0;//تعریف متغیر جمع
    for(i=0;i<=1;i++)     //حلقه ۲ تایی خارجی
    {
        for(j=0;j<=1;j++)     //حلقه ۲ تایی داخلی
        {
            cin>>x[i][j];     //پر کردن آرایه ۲*۲
        }
    }
    for(i=0;i<=1;i++)     //حلقه ۲ تایی خارجی
    {
        for(j=0;j<=1;j++)     //حلقه ۲ تایی داخلی
        {
            c=c+x[i][j];      //جمع آرایه ۲*۲
        }
    }
}
```

برای جمع ستون j ثابت می‌شود.

ماتریس خلوت:

ماتریسی که تعداد خانه های خالی(صفر) بیشتر خانه های پر باشد ماتریس خلوت است.در برنامه ماتریس خلوت بعد از پر کردن ماتریس از دو شمارنده باید استفاده کنیم یکی برای شمارش اعدا بالای صفر و دیگری برای اعداد صفر و درآخر آن ها را مقایسه می‌کنیم اگر خانه های صفر بیشتر بود ماتریس خلوت است.

```
#include<iostream.h>
void main()
{
     int j;          //تعریف متغیر برای حلقه
     int i;          //تعریف متغیر برای حلقه
     int c=0;        //تعریف متغیر جمع
     int m=0;        /تعریف متغیر جمع
     int x[3][3]; //تعریف آرایه ۴*۴
     for (i=0;i<=3;i++)    //حلقه ۴ تایی خارجی
     {
          for (j=0;j<=3;j++)    //حلقه ۴ تایی داخلی
          {
               cin>>x[i][j];     //پر کردن آرایه ۴*۴
          }
     }
     for (i=0;i<=3;i++)    //حلقه ۴ تایی خارجی
     {
          for (j=0;j<=3;j++)    //حلقه ۴ تایی داخلی
          {
               if (x[i][j]=0)          //شرط صفر بودن خانه آرایه
               c=c+1; //اگر خانه آرایه صفر باشد،یکی به شمارنده صفر اضافه می شود
               Else      //وگرنه
               {
                    m=m+1; //یکی به شمارنده صفر نبودن اضافه می شود
               }
          }
     }
     if(c>m) //اگر شمارنده صفر از شمارنده عدد بزرگ بود
cout<<"khalvat";}//یعنی ماتریس خلوت است
```

سوالات مربوط به شما :

برنامه بنویسید که اگر در ماتریس ۳*۳ قطر اصلی صفر بود پیغام okچاپ کند .

در این سوال شماره خانه های قطر اصلی را پیدا و برای آن شرط بگذارید.نمونه این سوال در بالا حل شده.

فصل چهارم

کلاس

هدف های رفتاری:

تعریفی از کلاس و توابع را می آموزید.

با حلقه شرطی آشنا می شوید.

مفاهیم اشاره گر ها و کاربرد آن را فرا

می‌گیرید.

در حلقه شرطی کار تکرایست و سه سوال ما به این صورت جواب داده می‌شود:

از کجا: شروع حلقه باید عددی مخالف صفر باشد.

تا کجا: معلوم نیست(مبهم)شرط توقف داشته باشیم.

چند تا چند تا : یکی یکی

در forچون انتها را می دانستیم پس برای تعداد مشخص استفاده می‌شد ،امـا در while به علت ندانستن شروع و پایان برای تعداد مبهم استفاده می‌شود.

برنامه ای بنویسید که تا زمانی که صفر وارد نکردیم از کاربر عدد دریافت کند.

دراین برنامه به علت ندانستن پایان حلقه نیاز به یک شرط داریم(شرط مخـالف صفر)

```
#include<iostream.h>
void main()
{
    int c=1;      //مقدار اولیه
    while(c!=0)  //شرط
    {
        cin>> c;  //دریافت عدد به منظور ادامه حلقه
    }
}
```

کلاس چیست؟

روند برنامه نویسی در گذشته به صورتی بود که همه برنامه رو یکجا می‌نوشتن . این نوع برنامه نویسی دو تا عیب بزرگ داره . یکی اینکه اگه قسمتی از برنامه رو بخوایم عوض کنیم باید کل برنامه تغییر کنه مثلا تغییر یه فرمول تو قسمتی از برنامه، کل برنامه رو به هم می‌ریزه دوم اینکه اگه کار های تکراری تو کل برنامه دائم تکرار میشه. فرض کنید قراره یه برنامه ماشین حساب بنویسید ،اگه همه ی دستوراتو پشت سر هم بنویسید مطمئنا برنامتون شلوغ و گیج کننده میشه و دائم باید دستورات تکراری مثل جمع و تفریق و ضرب و تقسیمو تایپ کنید . میدونم فکر کردن به این مسئله خودش عذاب آوره اما اینم یه راه داره . اگه می خوای برسی به راه حل بقیه مطلبو بخون .

روند برنامه نویسی همینجوری ادامه پیدا نکرد. برنامه نویسی به سمت پیمانه شدن پیش رفت .یعنی کد و داده کنار هم قرار گرفت .مبحث کلاس ها به وجود آمدند ،که می‌شد در هر کلاس چندین توابع تعریف و هر تابع را در هر جایی مورد

استفاده قرار داد. تو مثال بالا کافیه یک بار هـر عمـل رو تعریـف کنیـد و جاهـای مختلف فقط فراخونیش کنید.

یه مثال تو دنیای واقعی:

مثلا زمانی که شما قراره یه میوه بخورید به کارد ، بشقاب و میوه نیازمندید. کـه هر کدوم از اینا یه جایی تو آشبزخونه قرار گرفته .دیگه شما نیاز ندارید اول کـارد و درست کنید بعد بشقاب ،بعد میوه .چون یه بار اینا درست شدن و تو کارای مختلـف شما ازش استفاده می کنید.

کلاس ها هم بخاطر این مسئله به وجود اومدن. تو کلاس توابعی تعریف میشه تا کارای مختلفی رو انجام بده. بیرون کلاس واسه هر تابع دستورات نوشته میشه و در بدنه اصلی فقط فراخونی میشه.

شکل کلی کلاس:

```
Class     نام کلاس//
{
Public:
Int x ;   // تعریف متغیر

void  tavan (void)
};
```

کلمه Public به مفهوم عمومی بودن متغیر و توابع می باشدو هر جایی قابل استفاده می باشد.کلمه void به مفهوم نداشتن است.یعنی این تابع ورودی و خروجی ندارد . و فقط برای کاری ساده مورد استفاده قرار می‌گیرد.البته توابع براساس کار کردشان می توانند ورودی و خروجی داشته باشن . که در مباحث آینده به آن می‌پردازیم.

نحوه نوشتن دستورات داخل توابع:

```
void نام کلاس :: نام تابع (void)
{
دستورات
}
```

نحوه استفاده در بدنه:

```
void main()
{
    x classname          با این متغیر به توابع آن کلاس دسترسی داریم//
x.نام تابع ();
}
}
```

هر تابع می تواند شرایط زیر را داشته باشد.

- **نه ورودی نه خروجی**

```
void    نام تابع  (void)
```

- **فقط ورودی**(مثلا نوع صحیح)منظور دستور cin در تابع نیست.

```
void    نام تابع  (int x, int y)
```

- **فقط خروجی**

```
int    نام تابع  (void)
```

- **هم ورودی هم خروجی**

```
int نام تابع  (int x)
```

برنامه کنترل دما

دمای هوای محیط را **دریافت** و با توجه به دمای دریافتی **خروجی مناسب** دهد.

دمای بین ۰ تا ۱۰ =سرد

دمای بین ۱۱ تا ۲۰ =خوب

دمای بین ۲۱ تا ۳۰ = گرم

در این سوال نیاز به ورودی داریم . پس موقعه تعریف تابع برای آن ورودی تعین می کنیم و در بدنه اصلی این ورودی را با cin دریافت می‌کنیم و در تابع جایگزین می‌کنیم.

```
#include<iostream.h>
    Class dama   //کلاس دما
    {
        Public:
        void temp(int a);     //تابع دما
    };
    void dama::temp(int a)    //استفاده از تابع .این تابع دارای ورودیست

    {
        if (a>0 &&a<=10) //اگر ورودی کمتر از ۱۰  و بیشتر از صفر بود
        {
            cout<<"cold";//چاپ کن سرد
        }
        Elseif(a>=11  && a<=20)//وگرنه اگر ورودی بیشتر از ۱۱ و کمتر از ۲۰ بود
        {
            Cout<<"good";//چاپ کن خوب
        }
        Elseif(a>=21  && a<=30)//وگرنه اگر ورودی بیشتر از ۲۱ و کمتر از ۳۰ بود
        {
            cout<<"hold";//چاپ کن گرم
        }
    }
    Void main()
    {
        int t=1;//تعریف متغیر برای حلقه شرطی
        Dama x;//تعریف متغیری از نوع کلاس برای دسترسی به توابع آن
        While(t!=0)//استفاد از متغیر بالا برای شرط توقف
        {
            Cin>>x;// دریافت عدد
            x.dama(x);//استفاده از عدد در تابع دما
        }
}
```

در بدنه یک متغیر تعریف کردیم تازمانی کاربر عددی غیر صفر وارد کـرد برنامـه ادامه یابد همین که صفر وارد کرد برنامه خاتمه یابد .سپس متغیری از نوع کـلاس برای دسترسی به توابع آن قرار دادیم. و درآخر متغیـر ی دیگـر بـرای ورودی تـابع نوشتیم.

کلاسی طراحی کنید دو عدد **دریافت** و**توابع جمع، تفریق ،ضرب** را پیـاده سازی کند.

```
#include<iostream.h>
    class calc          // کلاس ماشین حساب
    {
        Public:
        int x;    //تعریف متغیر برای دریافت عدد
        int y;    //تعریف متغیر برای دریافت عدد
        void  jam(void);       //تابع جمع
        void zarb (void);      //تابع ضرب
        void tafrigh(void);    //تابع تفریق
        void daryaft (void);        //تابع دریافت
        void menu (void);      //تابع چاپ منو
    };
    void calc:: menu(void)
    {
        cout<<"1-jam"<<"2-zarb"<<"3-tafrigh";//چاپ منو
    }
    void calc:: daryaft(void)
    {
        cout<<"adad aval ra vared konid";
        cin>> x;      // دریافت عدد
```

```
    cout<<"adad dovam ra vared konid";
    cin>>y;        //دریافت عدد
}
void calc:: zarb(void)
{
    cout<<x*y;    //چاپ ضرب دو  عدد دریافتی بالا
}
void calc:: jam(void)
{
    cout<<x+y;    //چاپ جمع دو  عدد دریافتی بالا
}
void calc:: tafrigh(void)
{
    cout<<x-y;    //چاپ تفریق دو  عدد دریافتی بالا
}
void main()
{
    calc a;  //تعریف متغیری از نوع کلاس
    char y=0;      //برای انتخاب عدد منو
    a.menu();      //فراخوانی تابع منو
    a.daryaft();        //قبل ازانجام عملیات دو عدد دریافت می شود
    a=getch();    //ابتداعدد منورامیزنیم
    do
    {
        if(a=='1')  //اگر عدد ۱ زده شود عملیات جمع
            y.jam ();
        elseif(y=='2')//وگرنه اگر عدد ۲ زده شود عملیات ضرب
        {
            y.zarb ();
        }
        elseif(y==3')//وگرنه اگر عدد ۳ زده شود عملیات تقسیم
        {
            y.tafrigh();
        }
```

```
        getch();//این دستوربرای دریافت عدد بعدیست
        }(while=='5');    // شرط پایان
}
```

کلاسی طراحی کنید:

۱-تابعی که دو عدد را **دریافت** و جمع آن را **برگرداند**.

۲-تابعی که دو عدد را **دریافت** و عدد بزرگتر را **برگرداند**.

```
#include<iostream.h>
    class omome  //کلاس عمومی
    {
        public:
        int  jam(int x, int y);//تابع جمع دارای ورودی و خروجی
        int  max(int x, int y);//تابع عدد بزرگتر دارای ورودی و خروجی
    };
    int omome::jam(int x, int y)
    {
        return(x+y);      //باز گردانی جمع دو عدد
    }
    int omome::max(int x, int y)
    {
        return((x>y)?x:y);    //بازگردانی بزرگترین عدد
    }
    void main()
    {
        int a;   //تعریف متغیر  برای دریافت عدد
        int b;   //تعریف متغیر برای دریافت عدد
        omome c  //تعریف متغیراز نوع کلاس
```

```
        cin >>a>>b;//دریافت دو عدد
        cout<<c.jam(a,b);      //چاپ بازگردانی دو عدد
        cout<<c.max(a,b);//چاپ باز گردانی بزرگترین عدد
}
```

اشاره گرچیست؟

برای دسترسی مستقیم به حافظه از اشاره گر استفاده می‌کنیم.

تعریف اشاره گر:

```
int *c;
char *c;,,...
cin >> c;        یک رشته می خواند
cin>>*c;         یک حرف می خواند
```

برای دریافت رشته از اشاره گر استفاده می‌کنیم و تعریف آن به صورت زیر است.

```
char *c;
c=new (char);
```

تابعی بنویسید که رشته ای را **دریافت** کند.

```
void daryaft::reshteh(void)
{
    char *c;        //تعریف اشاره گر
    c=new(char);        //آزاد سازی فضای حافظه
    cin>>c;        //دریافت رشته
```

```
}
```

تابعی بنویسیدرشته ای دریافت و حرف A را در آن بشمارد.

```
void daryaft::reshteh(void)
{
     char *c;       //تعریف اشاره گر
     c=new(char);       //آزاد سازی فضای حافظه
     cin>>c;        //دریافت رشته
     int i=0;       //تعریف متغیر برای حلقه شرطی
     while(*(c+i)!='\0')//تازمانی که متن به پایان نرسد
     {
          i++;// شمارش
     }
     cout i;//چاپ تعداد
}
```

تابع سازنده چیست؟

تابعی با نام کلاس که مقدار پیش فرض هر چیز را داخل آن قرار می‌دهند و قبل از اجرا برنامه به صورت خود کار اجرا می‌شود.

مثال:

سیستم کنترل سرعت

دریافت **نوع** و **سرعت** وسیله

سرعت **بالای ۶۰** کیلومتر تخلف

تعداد وسایل نقلیه براساس نوع

```
#include<iostream.h>
    class soarat      //کلاس سرعت
    {
         public:
        soarat();     //تابع سازنده
        void menu(void);       //تابع منو
        void daryaft(void);          //تابع دریافت
        void takhalof(void);         //تابع تخلف
    };
    soarat::soarat()//تابع سازنده برای مقداردهی اولیه
    {
        m=0;
        a=0;
        d=0;
    }
    void soarat::daryaft(void)
    {
        int s;   //تعریف متغیر برای دریافت عدد سرعت
        char v;  //تعریف متغیر برای دریافت وسیله
        cout<<" vaziat";          //چاپ وضعیت
        cin>>s;  //دریافت سرعت
        if(v=='m'&& s>60)//اگر وسیله موتور و سرعتش بیشتر ۶۰بود
            M=m+1;//شمارنده موتور
        elseIf(v=='a'&& s>60)//وگرنه وسیله پیکان و سرعتش بیشتر ۶۰بود
```

```
        {
            A=a+1;//شمارنده پیکان
        }
        elseIf(v=='b && s>60)//وگرنه وسیله سنگین و سرعتش بیشتر ۶۰بود
        {
            b=b+1;//شمارنده ماشین سنگین
        }
Void soarat::takhalof(void)
{
    Cout<<"motor"<<m;      //چاپ موتور و تعداد آن
    Cout<<"paykan"<<a;     //چاپ پیکان و تعداد آن
    Cout<<"sangin"<<b;     //چاپ سنگین و تعداد آن
}
Void soarat::manu(void)  //تابع چاپ منو
{
    Cout<<"1-daryaft";
    Cout<<"2-takhalof";
}
Void main()
{
    Soarat x;     //تعریف متغیری از نوع کلاس
    Char z;  //برای انتخاب عدد منو
    Do{
        x.menu();     //فراخوانی تابع منو
        z=getch();    //ابتداعدد منورامیزنیم
        if (z=='1')  //اگر عدد ۱ زده شود عملیات دریافت
            x.daryaft();
        elseif(z=='2')    //وگرنه اگر عدد ۲ زده شود عملیات تخلف
        {
            x.takhalof();
        }
        Getch();
    }while(z!='3');
}
```

فصل پنجم

خواندن و نوشتن در فایل

هدف های رفتاری:

با فایل ها آشنا می شوید.

قادر به نوشتن و خواندن در فایل خواهید بود.

نوشتن و خواندن در فایل:

برای نوشتن و خواندن در فایل باید ابتدا اشاره گری از نوع فایل تعریف کرد که از هدر #include<stdio.h> استفاده می کند.،سپس فایل مورد نظر را با دستور fopen در مسیری ایجاد کنیم. حالا می توانیم درون فایل بنویسیم.

نکاتی در مورد فایل

دستور تعریف اشاره گر فایل به صورت زیر است .در این دستور اشارگری با نام f درست می‌شود.حتما باید فایل را با حروف بزرگ نوشت.

```
FILE *f;
```

بعد از تعریف فایل باید آن را با دستور fopen باز کرد.

```
p=fopen("c:\\f.txt","w");
```

حالا نوبت تعریف متغیری از نوع کاراکتر است که بتوانیم ورودی را از کاربر بگیریم.چون نیاز به گرفتن کاراکتر به کاراکتر را داریم باید از دستور getche(); استفاده کنیم .که هدرآن #include <conio.h> است.

```
char c;
```

```
c=getche()
```

برای نوشتن در فایل از دستور putc(c,p); استفاده می کنیم که c همان حرف و p فایلیست که در آن می‌نویسیم. و در آخر بستن فایل الزامیست.

```
fclose(f);
```

برنامه نوشتن در فایل:

```
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
    void main()
    {
        FILE *p;        //تعریف اشاره گری از  نوع  فایل
        char c;         //تعریف متغیری از نوع رشته
        p=fopen("c:\\f.txt","w");//باز کردن فایل برای نوشتن در آن
        c=getche();//دریافت اولین حرف
        while(c!='\r')//شرط توقف: زدن اینتر
        {
            putc(c,p);// نوشتن حرف در فایل
            c=getche();//دریافت حرف بعدی
        }
        fclose(p);//بستن فایل
        getch();// زدن یک حرف برای پایان
```

}

در این برنامه چون نیاز به گرفتن تک تک حروف داریم و از طرفی نمی‌دانیم که کاربر چه تعداد حروف وارد می کند پس باید از یک حلقه شرطی استفاده کنیم من قبل از حلقه ابتدا کاراکتری را دریافت کردم سپس با شرط حلقه بررسی کردم که اینتر زده شود.درون حلقه آن کاراکتر را داخل فایل نوشتم و حرف بعدی را دوباره گرفتم.و در آخر هم فایل را بستم.

در دستور fopen:

```
f=fopen("d:\\x.txt","a");
```

اشاره گر فایل=f

مسیر فایل"d:\\x.txt" :

""پارامتر تعیین کننده خواندن و نوشتن که w یعنی نوشتن و r یعنی خواندن و a اضافه کردن به انتهای فایل(اگر از دو مورد قبل استفاده کنیم با هر بار اجرا فایل از اول نوشته می شود اما با این گزینه در ادامه فایل نوشته می شود)اگرr+یا w+ نوشته شود هم می‌توان نوشت هم می‌توان خواند.

خواندن از فایل :

برای خواندن از فایل از دستور c=getc(k); استفاده می کنیم.که در آن c متغیریست که اطلاعات فایل k در آن ریخته می شود.. برای خواندن از فایل هم نیاز به حلقه شرطی داریم که انتهای فایل را بررسی کند.

```
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
void main()
{
     FILE *k;//تعریف اشاره گری از  نوع  فایل
     char  c;//تعریف متغیری از نوع رشته
     k=fopen("c:\\f.txt","r");//باز کردن فایل برای نوشتن در آن
     c=getc(k);//دریافت اولین حرف
     while(c!=EOF)دستوربررسی انتهای فایل
     {
          cout<<c;چاپ کاراکتر
           c=getc(k);دریافت کاراکتربعدی
      }
      getch();// زدن یک حرف برای پایان
      fclose(k);//بستن فایل
}
```

برنامه ای بنویسید که فایلی را بخواند اگر حرف a در آن بود در فایلی دیگر با حرف b جایگزین کند.

در برنامه ما نیاز به دو اشاره گر داریم یکی برای خواندن فایل اول دومی برای نوشتن فایل دوم .اولین حرف فایل اول را می‌خوانیم در حلقه شرط پایان فایل را می‌زنیم و درون حلقه بررسی می‌کنیم اولین حرف دریافتی a هست یا نه اگر بود حرف b را چاپ و در فایل دوم بنویس وگرنه همان حرف a را چاپ و در فایل دوم می‌نویسد.

```
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
void main()
{
     FILE *k;//تعریف متغیری از نوع اشاره گر
     FILE *b;//تعریف متغیری از نوع اشاره گر
     char c;//تعریف متغیری از نوع رشته
     k=fopen("c:\\f.txt","r");//باز کردن فایل برای خواندن از آن
     b=fopen("c:\\f1.txt","w");//باز کردن فایل برای نوشتن در آن
     c=getc(k);//دریافت اولین حرف
     while(c!=EOF)دستوربررسی انتهای فایل
     {
          if(c=='a')// اگر حرف گرفته شده برابر حرف مورد نظر بود
          {
               cout<<'b';//حرف روبرو را چاپ
               putc('b',b);//حرف مورد نظر را در فایل  مورد نظر بنویس
          }

     Else//وگرنه
      {
           putc(c,b);//حرف خوانده شده را بنویس
           cout<<c;//و چاپ کن
      }
      c=getc(k);دریافت کاراکتربعدی
      }
```

```
        getch();// زدن یک حرف برای پایان
        fclose(k);// بستن فایل
        fclose(b);// بستن فایل

}
```

برنامه ای بنویسید تعدا حروف فایلی را بخواند و چاپ کند.

در برنامه کافیست که در حلقه یک شمارنده گذاشت.

```
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
void main()
{
     FILE *k;//تعریف اشاره گری از  نوع  فایل
     char c;//تعریف متغیری از نوع رشته
     int x=0;// تعریف متغیری از نوع عدد برای شمارش
     k=fopen("c:\\f.txt","r");//باز کردن فایل برای خواندن
     c=getc(k);//دریافت اولین حرف
     while(c!=EOF)دستوربررسی انتهای فایل
     {
          c=getc(k);دریافت کاراکتربعدی
          x=x+1;//شمارش کارکتر
      }
      cout<< x;چاپ کاراکتر
      getch();// زدن یک حرف برای پایان
      fclose(k);// بستن فایل
      fclose(b);// بستن فایل

}
```