



مجموعه آموزشی پی استور

# جزوه دست نویس برنامه نویسی سی پلاس پلاس

# C++

استاد مربوطه: مهندس امین جلیل زاده رزین

تهیه و تنظیم: فرشته داداش زاده

Subject :  
Year. Month. Date.

برنامه نویسی : ۱- الگوریتم ۲- برنامه نویسی

الگوریتم : گذر انجام دادن کار به صورت قدم به قدم و ترتیبی به صورتی منظم و جل به

منظم شود و لزوماً منجمد و غیر زنده نباشد.

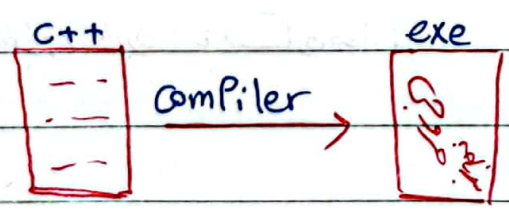
برنامه نویسی : روش پیاده سازی الگوریتم به کمک زبانهای برنامه نویسی.

برنامه نویسی : ۱- syntax (ساختار) ۲- semantic (معنوم).

زبان برنامه نویسی C++ : نسخه ارتقا یافته زبان C می باشد که در آن ویژگیهای

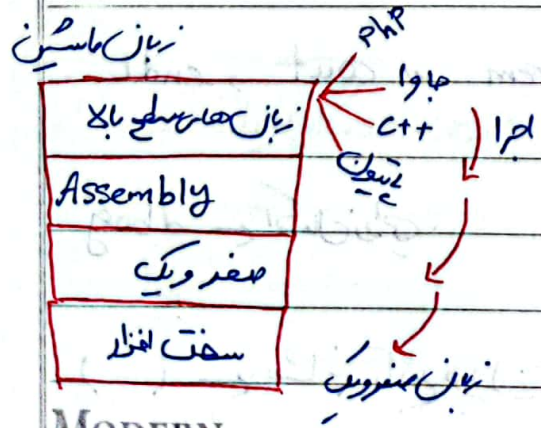
بسیار مهمی اضافه شده است؛ بنابراین زبان C++ دارای خصوصیات زبان C

دارد می باشد و علاوه بر آن نسخه شی براس دارد.



میانگین مصنف داریم (برای C++ ← DevC++)  
(مثال دیگر: borland - turbo)

حوزه برنامه نویسی در صنعت software.



۶۵٪ از کادری ...

MODERN



Subject: # سار # نامبرستان  
Year. Month. Date.

① برنامه نوشتن Hello world در صفی خروجی

```
#include <iostream>
```

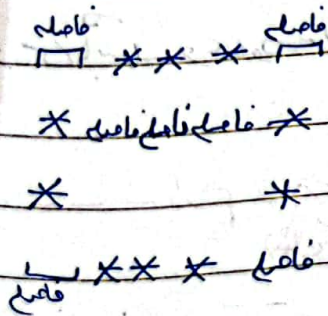
```
using namespace std;
```

```
int main ( )  
{  
  cout << " Hello world";  
}
```

shift → " "

اجرای

```
cout << "*" << endl; ← ***  
cout << " *** " << endl; ← *** **  
*** ** *
```



اجرای در خط سطرها: ۰۱۳

• cout و endl در iostream تعریف شده اند.

چون ما به آن اشاره کردیم. ← عمل در خطی

( ) ← نشانه تابع بودن

MODERN

temp ← متغیر



Subject :

Year. Month. Date.

متغیر و انواع آن

تعریف متغیر: متغیر از حافظه (RAM) که دارای نام، سائز، نوع، آدرس است و

برای ذخیره داده در آن استفاده می‌شود. (نوع و نام مهم است.)

نام متغیر: حتما باید با حروف شروع شود و می‌تواند شامل عدد یا خط تیره باشد. (از خط تیره)

در انتخاب نام متغیر دقت کنید، یعنی نام باید با مصداق و با همخوانی باشد.

در C++: ① متغیر اعداد: A صحیح ② متغیر شناور ③ متغیر کارائری

④ متغیر بولی: True - False ⑤ enum

نوع short = ۲ بایت int = ۴ بایت long = ۸ بایت	هم + و هم -	انواع صحیح بر اساس سائز:
---	-------------	--------------------------

فقط برای اعداد +	unsighde short = unsighde int = unsighde long =	۱ بایت ←
		۲ بایت ←

نظام متغیر X : نوع : صحیح / سائز : ۴ بایت / آدرس : خودکامیله در انتخاب می‌کنیم!  
نام : int

MODERN



هم اعداد + وسم -	} ۴ بایتی	float	} ۱۱	متغیر شناور :
		double		
فقط برای اعداد +	} ۸ بایتی	unsighnde float	۱۳	
		unsighnde double	۱۴	

char : کاراکتری

bool : بولی

enum : enum

۱۴، ۲۷

حاصل ۲

متغیرهای منطقی نیاز به (صریح / غلط) و (بلیه / ضد) و... (است) از بولی استفاده میکنند

0 = False

1 = True

در واقع تمام اعداد جز صفر True یا اشتباه هستند

نکته: داخل " " معنی عبارت احوال هستند. بدون " " مصوباً داخل متغیر

```
cout << "x" = "x";
```

تلفظ      تلفظ

```
cout << "x" << "x" << "x";
```

تلفظ      عدد      تلفظ



Subject :

Year. Month. Date.

• متغیری که داخل main تعریف شود ← محلی Local

• خارج = = ← سراسری Global

• تغییر محلی مقدار همی الهی این توسط کاربر انجام می شود، اگر انجام نشود مقداری از حافظه را

در خود رزده می کند ← در این مقدار تعریف انتخاب خواهد شد.

در تغییر سراسری هم درک پس از این برای تغییر هائی عددی صفر و برای پارامتری

na می باشد.

② در تغییر x و y را بخواه عددی کنید و حاصل جمع آنها را در متغیر z بنویسید

```

#include <iostream>
using namespace std;
main ( )
{
  int x, y, z;
  cout << "Please enter x: ";
  cin >> x;
  cout << "Please enter y: ";
  cin >> y;
  z = x + y;
  cout << "Sum is: " << z;
}

```

x = 2  
 y = 1  
 z = x + y  
 z = 1 + 2  
 z = 3  
 3

MODERN



Subject :

Year.

Month.

Date.

cout << اخرج و شروع به حرف اول کلمه اول (خروجی) ← خروجی

cin >> ← برعکس cout

تعداد رهن اطمینان ← initialization متغیر ← Variable

در تقسیم، متغیری که باید استفاده شود ← float / اعداد اعشاری ← خروجی

دانش آموزان ۴ گروه در هر درس را از طریق کلاس و معلمان کلاس متغیر

```
#include <iostream>
```

محاسبه و چاپ کنید:

```
using namespace std;
```

```
main()
```

```
{
```

```
float x, y, F, w;
```

```
float S
```

```
cout << "Please enter x: " << endl;
```

```
cin >> x;
```

```
cout << "Please enter y: " << endl;
```

```
cin >> y;
```

```
S = x * y * F * w;
```

```
cout << "result: " << S << endl;
```

MODERN



Subject : \_\_\_\_\_  
 Year. \_\_\_\_\_ Month. \_\_\_\_\_ Date. \_\_\_\_\_

۴) حاصل نویسی در شعاع دایره از نظر تغییر در صیغه و ساختار چاپ کنند:

```

float r, P, S;
cout << "enter r: ";
cin >> r;
P = (2 * 3.14 * r);
S = 3.14 * r * r;
cout << "result: " << P << endl;
cout << "result: " << S;
    
```

شکل	صیغه	ساختار
m <sub>1</sub> مثلث	مجموع ۳ ضلع	۲ / (ارتفاع × قاعده)
m <sub>2</sub> دایره	۲ π r	π r <sup>2</sup>
m <sub>3</sub> لوزی	۲ × کِضلع	۲ / (قطر کوچک × قطر بزرگ)
m <sub>4</sub> زوزنه	مجموع ۳ ضلع	ارتفاع پا × (قاعده کوچک + قاعده بزرگ)
m <sub>5</sub> متوازی الاضلاع	طول + عرض × ۲	قاعده × ارتفاع
m <sub>6</sub> مستطیل	(عرض + طول) × ۲	طول × عرض
m <sub>7</sub> مربع	۴ × ضلع	ضربش × ضلع
m <sub>8</sub> بیضی	میانگین ۲ شعاع × ۲ × π	شعاع کوچک × شعاع بزرگ × π

فصل ۲ : دستورات شرطی (انتخابی) : condition

```

if ( condition )
{
    Estatement ( دستورات )
}
else در غیر این صورت
{
    دستورات
}

if (a) b
{
    cout << "a";
}
else
{
    cout << "b";
}
    
```

MODERN





Subject :

Year.

Month.

Date.

```
if (a > b) : else if
{ cout << "a"; : else if
}
else if (a < b)
{ cout << "b";
}
else
{ cout << "mosavi";
}
```

برنامه ای بنویسید که سه عدد را از کاربر بگیرد و بزرگترین عدد را چاپ کند: توسعه

```
float a, b, c;
cout << "enter a : ";
cin >> a;
cout << "enter b : ";
cin >> b;
cout << "enter c : ";
cin >> c;
```

```
if (a > b && a > c)
cout << "a";
else if (b > a && b > c)
cout << "b";
else cout << "c";
```

MODERN



Subject :

Year. Month. Date.

if (.....)

False داخل else همیشه

{ True

}

= if = True

else

{ False

if (..... or.....)

(یا)

if (a == b)

}

جدول :

x	not !	x	y	or	x	y	and &&
0	1	0	0	0	0	0	0
1	0	0	1	1	0	1	0
		1	0	1	1	0	0
		1	1	1	1	1	1

a = 5.

نشان برای not

if (a != 5.)

Yes جواب

cout << a;

else

cout << "yes";

(4) برنامه بنویسید که عدد از طرف کاربر و مشخص

کنند زوج است یا فرد؟ عدد تقسیم بر 2 و باقیمانده زوج

int a;

cout << "enter a :";

cin >> a;

int a, b;

if (a % 2 == 0)

b = a / 2

cout << "zoj";

if (b == 0)

MODERN else

cout << "fard";



استوار بولیں!

عملدرستی اور افراسی و اٹھسی:

1) پیش وندری ++x و --x : اول اٹھسی و افراسی بعد حساب!

2) پس وندری x-- و x++ : اول حساب بعد افراسی یا اٹھسی!

```
int x = 1
```

```
cout << "x";
```

```
x = x + 1; → x++ / x+ = 1
```

```
x = x - 1; → x-- / x- = 1
```

```
int x
```

```
x = 12
```

```
x /= 5;
```

```
cout << x
```

مثال:

→ جواب x = 2

if

elseif

else

if ( صفر )

{

}

else

{

}

صفر صفر False

اسے پس در else میں نہیں

✓

```
if ( ... && ... && ... )
```

```
if ( ... || ... || ... )
```

```
if ( ... ! = )
```

```
if ( == )
```

```
if ( ... > ) و if ( ... < ... )
```

```
if ( ... < = ) و if ( ... > = )
```

MODERN



Subject :

Year. Month. Date.

نشانهای نویسی در ضربات معادله درجه اول با ضرایب صحیح از طریق یکسره در دست آورده شد: (۷)

$$ax + b = c$$
$$x = \frac{c-b}{a}$$

```
float a, b, c;
cout << "enter a: ";
cin >> a;
```

```
x = c - b / a;
cout << "result: " << x;
```

۱۵ - ۱

نشانهای نویسی در سه ضرایب معادله درجه دوم از طریق یکسره در دست آورده شد: (۸)

```
int a, b, c;
int delta;
cout << "enter a: ";
cin >> c;
```

$$ax^2 + bx + c = 0$$
$$\Delta = b^2 - 4ac$$
$$x = \frac{-b \pm \sqrt{\Delta}}{2a}$$
$$x = \frac{-b - \sqrt{\Delta}}{2a}$$

```
delta = (b * b) - 4 * c * a;
```

```
if (delta < 0)
```

```
{ cout << "rishenadard";
```

```
}
```

```
else if (delta == 0)
```

```
{ cout << "-b + sqrt(delta) / a x 2 << endl;
```

```
cout << "-b - sqrt(delta) / a x 2;
```

```
}
```

```
else
```

درون برنامه باید `#include <math.h>` sqrt

```
{ cout << -b / (a * 2);
```

```
}
```

```
}
```

MODERN



Subject :

Year.

Month.

Date.

9) برنامه‌ای بنویسید که ۳ عدد را بگیرد و بزرگترین آن را نمایش دهد.  
 c, b و a

```

if (a+b > c) || (b+c > a) || (c+a > b)
{ cout << "b, a و c بزرگترین عدد است"; }
else
{ cout << "c بزرگترین عدد است"; }

```

• اگر تعداد if ها زیاد باشد از دستور switch استفاده می‌شود.

switch (متغیر)

```

{ case 1 : دستورات ; break ;
  case 2 : دستورات ; break ;
  case 3 : دستورات ; break ;
  ...
}

```

break ← قطع کردن، خارج شدن  
 اگر break نوشته نشود، برنامه اجرا را  
 خود رها کند.

مثلاً ← برنامه‌ای که اماها را بگیرد و با سی (one, two, three) ...  
 را چاپ کند.

MODERN



Subject :

Year.      Month.      Date.

• اگر هیچ برنامه‌ای اجرا نشود ← default اجرا خواهد شد.

default :

cout << "numbers not in Range 1-7"; (مثال برای حالت default)

19) برنامه‌ی بنویسید که عددی را بگیرد و اساس هفته‌ی آن را چاپ کند :

```
int a;
```

```
cin >> a;
```

```
switch(a)
```

```
{ case 1 : cout << "shanbeh";
```

```
break;
```

```
case 2 : cout << "yek shanbeh";
```

```
...
```

```
case 7 : cout << "jomeh";
```

```
}
```

// در صورتی که حافظت (برای حافظت نویسی) \*

MODERN



دستور حلقه (تکرار)

انواع : ① while ② do-while ③ for ④ goto

goto : در این تکرار برنامه نویسی استفاده نمی کنیم.

① while :  
 while (شرط) {  
 دستورات ;  
 }  
 مثال برنامه ای که ۱۰۰ بار Ali چاپ کند :  
 int x = 0 ;  
 شرط روی اولین شرط

```
int i = 0 ;
while ( i < 100 )
{ cout << "Ali" << endl ;
  i ++ ;
```

② برنامه ای که اعداد زوج کمتر از ۱۰۰ را چاپ کند :

```
#include <iostream>
using namespace std;
main ( )
{ int i = 2 ;
  while ( i < 100 )
  { cout << i << endl ;
    i += 2 ;
  }
}
```

```
{ int i = 1
  while ( i < 100 )
  { cout << i << endl ;
    i += 2 ;
  }
}
```

اعداد فرد را چاپ کند :



Subject :

Year.

Month.

Date.

```

int i = 0;
int sum = 0;
while (i < 10)
{
    sum = sum + i;
    i++;
}
cout << sum;

```

برای هر مجموع اعداد متناهی صداهای اول:

$$Sum = \frac{n(n+1)}{2}$$

i	sum
0	0
1	0
2	1
3	3
4	6
5	10
...	...

```

int i = 2;
do

```

do-while :

نقطه! جوی while ← و یادت نرو!

```

{
    cout << i << endl;

```

```

    i++

```

برای اعداد زوج کمتر از ۱۰۰ ←

```

} while (i < 100);

```

```

}

```

```

int i = 0;

```

do-while :

```

do {

```

```

    i++

```

تفاوت while و do-while :

```

} while (شرط);

```

حلقه while ممکن است شروع برقرار نشود و برنامه اجرا نمی شود ولی

do-while حتماً یکبار اجرا می شود حتی در صورت برقرار نشدن شرط.

MODERN





for (بهره‌رسانی و شرط ; شماره دفعی اولی) : for, ۱۳

```
{
    ...
}
```

نظا اعداد زوج کمتر از ... با جمله for :

```
for (int i=1; i<=n; i++)
{
    cout << i << endl;
}
```

۱۳) برای این نویسی که مقسوم علیه های عدد وارد شده توسط کاربر اچال کند:

```
{
    int n;
```

```
cin >> n;
```

```
for (int i=1; i<=n; i++)
```

```
{
    if (n%i == 0)
```

```
{
    cout << i << endl;
}
```

i	n	n/i
1	12	12
2	12	6
3	12	4
4	12	3
5	12	2.4

۱۴) برای این که عدد را از کاربر بگیرد مشخص کند آن عدد نام است یا صفر:

```
int n;
```

```
cin >> n;
```

```
int sum = 0;
```

```
for (int i=1; i<=n; i++)
```

```
{
    if (n%i == 0)
```

```
{
    sum = sum + i;
}
```

```
}
```

```
if (sum == n)
```

```
{
    cout << "tam";
}
```

```
}
```

```
else
```

```
{
    cout << "not tam";
}
```

MODERN



برای هر عدد کوچکتر از ... را برسی کند و نام بوزن یا بوزن آن را مشخص کند

```

for (int n=1; n<100; n++)
{
    int sum=0;
    for (int i=1; i<n; i++)
    {
        if (n/i==0)
        {
            sum+=i;
        }
    }
    if (sum==n)
    {
        cout<<n<<endl;
    }
}

```

برای هر عددی که در جدول ضرب ظاهر شود

```

for (int i=1; i<10; i++)
for (int j=1; j<10; j++)
{
    cout<<i*j<<" " << " ";
    cout<<endl;
}

```

جواب سوال 1) مانتی تقوینا

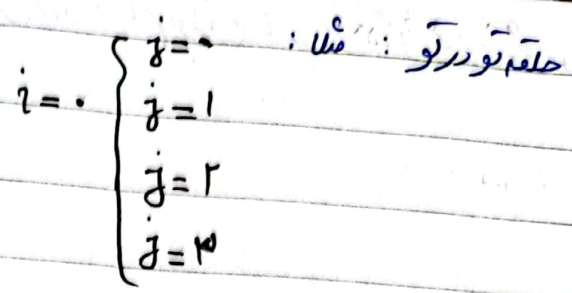
i	j	m
0	0	0
1	0	1
1	1	1
2	0	2
2	1	2
2	2	4
3	0	3
3	1	3
3	2	6
4	0	4
4	1	4
4	2	8
4	3	12
5	0	5
5	1	5
5	2	10
5	3	15
5	4	20
6	0	6
6	1	6
6	2	12
6	3	18
6	4	24
6	5	30
7	0	7
7	1	7
7	2	14
7	3	21
7	4	28
7	5	35
7	6	42
8	0	8
8	1	8
8	2	16
8	3	24
8	4	32
8	5	40
8	6	48
8	7	56
9	0	9
9	1	9
9	2	18
9	3	27
9	4	36
9	5	45
9	6	54
9	7	63
9	8	72

m = 9 جواب ←



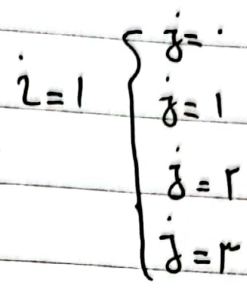
```
for (i = 0; i < 5; i++)
```

```
for (j = 0; j < 4; j++) →
```



```
cout <<
```

سپس کتا j و کتا i را چاپ می کند  
کتاب cout ، کتا کتا



الآن کتا i = 4

۱۷) برنامه ای بنویسید که یک عدد را از ورودی بگیرد و مشخص کند عدد اول است یا نه!

```

{ int n;
  cin >> n;
  int flag = 0;
  for (int i = 2; i < n; i++)
  { if (n % i == 0)
    { flag = 1;
    }
  }
  if (flag == 0)
  cout << " adad Aval ";
  else
  cout << " Aval nist ";
}

```

شماره ۱۵  
۱۵/۲  
۱۵/۳  
۱۵/۴  
۱۵/۵  
۱۵/۶  
۱۵/۷  
۱۵/۱۴



Subject :

Year. Month. Date.

```

int Flag;
for (int n=2; n<100;n++)
{
  Flag = 0;
  for (int i=2; i<n;i++)
  {
    if (n%i==0)
    {
      Flag = 1;
    }
  }
  cout << n << " , " ;
}

```

برایمان که اعداد اول تا ۱۰۰ را بررسی کند: (۱۸)

```

for (int i=.; i<100; i++) : continue و break دستور
{
  if (i==30)
  {
    continue;
  }
  cout << i << " , " ;
}

```

↓ ادامه دادن حلقه      ↓ شکستن حلقه

در این حالت تمام اعداد یک تا ۳۰ بیخیز ۳۰ چاپ می شود.

```

for (int i=.; i<100; i++)
{
  if (i==30)
  {
    break;
  }
  cout << i << " , " ;
}

```

در این حالت نیز حلقه در ۳۰ قطع می شود.

برایمان که اعداد اول کمتر از ۱۰۰ را جمع و چاپ کند:

n = 4975.3

n/2 2487.6

break 4975.3

در هر دو حالت تعداد اعداد یکسان است.

MODERN



برای محاسبه فاکتوریل عددی از ورودی کاربر، برنامه بنویسید. (۱۹)

```

{ int n;
cin >> n;
int fact = 1;
for (int i = 1; i <= n; i++)
{ fact = fact * i;
}
cout << fact;
}

```

$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$

برای محاسبه اعداد فیبوناچی، برنامه بنویسید. (۲۰)

```

// Fibonacci
long int F1 = 0, F2 = 1, F;

```

$F = F1 + F2$

$F1 = F2$

$F2 = F$

خطای overflow ← خطای استت و مقدار عددی

```

cout << F1 << " و " << F2 << " و " << F;

```

```

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

```

یک متغیر از طرفیت آن سر نیز می شود.

```

{ F = F1 + F2;

```

```

cout << F << " و " << F1 << " و " << F2 << " و " << F;

```

```

F1 = F2;

```

```

F2 = F;
}
}

```



Subject :

Year.

Month.

Date.

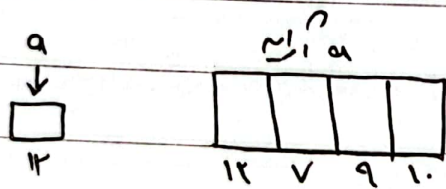
برنامه‌ای که حاصل عبارات زیر را حل دهد:

$$1 + (1/2!) + (1/3!) + \dots + (1/n!)$$

```

int sum = 0;
for (int a = 1; a <= u; a++)
{
  int fact = 1;
  for (int i = 1; i <= a; i++)
  {
    fact *= i;
  }
  sum = sum + (1/fact);
}
cout << sum

```



عبارت ۵، ۱۰

نمونه ۵: آرایه ها (Array)

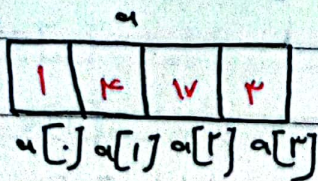
Declaration:

```
int a[ ];
```

آعداد خانه ها

اعلان یا تعریف آرایه :

بسیاری از عناصر آرایه : عددها بر اسم یک شماره هم دارد (خانه ها) هست:



تعریف آرایه : به معنیری که در این خانه های هم اشاره هم

```
a[2] = 17;
```

نوع در برنامهم ، آرایه نوشتند

```
MODERN cout << a[2];
```



مثال ۱: ارایه و اعداد عضو (عنصر) آن را بنویسید و سپس اعداد آن را در یک خط ارایه کنید

```
int x[5]; // (۱، ۲، ۳، ۴، ۵) : مقدار صده
```

```
x[0] = 15;
```

```
x[1] = 17; // به جای این → int x[5] = {15, 17, 19, 14, 1}
```

```
x[2] = 19; // می توانیم : مقدار دهی اولیه ارایه
```

```
x[3] = 14;
```

```
x[4] = 1;
```

```
int a[5] = {5, 14, 9, 77, 2};
```

```
cout << a[1];
```

↓  
for حلقه

```
int a[5] = {5, 14, 9, 77, 2};
```

```
int sum = 0;
```

```
for (int i = 0; i < 5; i++)
```

```
{ sum += a[i];
```

```
} cout << sum;
```

```
}
```

سیاسی ارایه

۲۲) برنامه ای بنویسید که اعداد ارایه را بخواند و سپس آن را در یک خط ارایه کند

```
{ int a[5] = {5, 14, 9, 77, 2};
```

```
for (int i = 0; i < 5; i++)
```

```
cout << a[i] << " , ";
```

```
}
```



Subject :

Year.

Month.

Date.

برنامه نویسی C++ // برای با انداز ۸ عنصری توقف کند طوری که طوری که عدد ۵ و ۱۰

```

int a[8];
for (int i=0; i<8; i++)
{cin >> a[i];}
for (int i=0; i<8; i++)
{cout << a[i] << " , ";}
}
}

```

```

int a[8];
cout << "Enter a number \n";
for (int i=0; i<8; i++)
{cout << "a[" << i << "] : ";
cin >> a[i];}
for (int i=0; i<8; i++)
{cout << a[i] << " , ";}
int item;
cout << "\n Enter item: ";
cin >> item;
for (int i=0; i<8; i++)
{if (a[i] == item)
{cout << "Found in index: " << i;}}
}
}
}

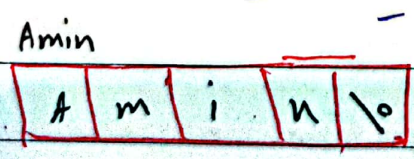
```

توقف کند ← n ← چقدر عدد ها  
n ←

رشته ها : String

توقف کند ← n ← چقدر عدد ها

نوعی رشته ها



```

char a;
a[0] = 'A';
a[1] = 'm';
a[2] = 'i';
a[3] = 'n';
a[4] = '\0';

```

توقف کند ← n ← چقدر عدد ها

\* gets → cin  
puts → cout



Subject :

Year.

Month.

Date.

۲۵) آیا حارس در این کد در این حالتی و پس کنند:

```
{ int i; char u[5];  
for (i = 0; u[i] != '\0'; i++)  
{ y[i] = u[i];  
} y[++i] = '\0';  
puts(y);  
char y[5];
```



۲۶) تمرین! عددی را تصویف / بویسته عددی را از عددی بجا بیاورید و این کار را در این

سهم (x) انجام کنند.



Subject :

Year. Month. Date.

۱۵/۵

علیم

۲۷) برنامه‌ای بنویسید یک عدد از طرف کاربر پس تعداد ارقام + معکوس، جمع ارقام

```

{ int n, b, sum = 0, i, m = 0;
  cin >> n;
  for (i = 0; i < n; i++)
  { b = n % 10;
    sum + = b;
    sum * = 10; // sum = sum * 10;
    n /= 10; // n = n / 10;
    m = (m * 10) + b;
  } cout << i << endl << m << endl << sum;
}

```

مسابره عالی کنند:

۲۸) برنامه‌ای بنویسید که شماره یک دانش آموز را از طرف کاربر بگیرد و میانگین نمره آن را حساب کند

```

{ float a[5];
  float sum = 0;
  for (int i = 0; i < 5; i++)
  { cin >> a[i];
  } for (i = 0; i < 5; i++)
  { sum + = a[i];
  } cout << sum / 5;
}

```

معدل آن را محاسبه کنند:



۲۹ برنامه‌ای بنویسید که رشته را از ابتدا برسد و مشخص کند که آیا پالیندروم است

```
{ char str[100];
```

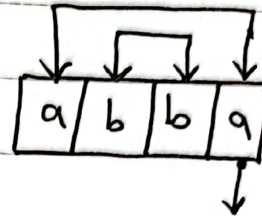
```
int flag = 0;
```

بازمانده!

```
cin.getline(str, 100);
```

```
// gets(str);
```

```
int n = strlen(str);
```



```
for (int i = 0; i < n/2; i++)
```

```
{ if (str[i] != str[(n-i)-1])
```

```
{ flag = 1;
```

$n = 4$  ← تعداد حروف

```
break;
```

$i = 0$

```
}
```

```
} if (flag == 0)
```

```
{ cout << "is palindrome";
```

```
}
```

```
else { cout << "is not palindrome";
```

```
}
```

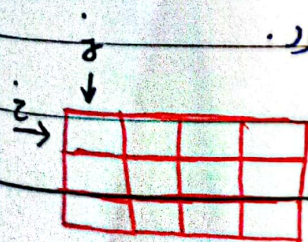
```
puts(str);
```

```
}
```

آیا یک عددی را می‌توانیم به صورت (مض) بود ...

آیا دو عددی: از نظر ستون تشکیل شده، معکوس برای اندیس گذاری سطرها از این

و برای اندیس گذاری ستون‌ها از مقادیر j استفاده می‌شود.



```
int a[i][j]
```

MODERN

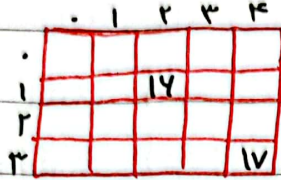


Subject :

Year.      Month.      Date.

اندیس‌ها از صفر شروع می‌شود - <sup>این دو بعدی همان ماتریس در ریاضیات است</sup>

برای دسترسی به عناصر <sup>اینست</sup> اندیس <sup>سروستون</sup> <sup>ایضا می‌توانیم</sup>



$\rightarrow$  `int a[4][5]`

`a[1][ ] = 14` ;

`cout << a[ ] [ ]` ;

نکته! برای پیچاس‌های <sup>این دو بعدی</sup> از جمله `for` و `while` استفاده می‌شود -

(۳) برنامه‌ای بنویسید که <sup>این دو</sup> در <sup>این</sup> از ورودی خوانده - در <sup>این</sup> ذخیره -

```
{ int a[5][5];
```

```
  for (int i=0; i<5; i++)
```

```
    for (int j=0; j<5; j++)
```

```
      { cout << "a[" << i << "][" << j << " ]";
```

```
        cin >> a[i][j];
```

```
    }
```

`setw` به اندازه‌ها در یک ردیف <sup>۴</sup>

```
  for (int i=0; i<5; i++)
```

```
    { cout << setw(4) << endl;
```

کاراکتر <sup>۴</sup> صحت می‌کند.

(iomanaliP)

```
    for (int j=0; j<5; j++)
```

```
      { cout << a[i][j] << " ";
```

```
        cout << a[i][j] << setw(4);
```

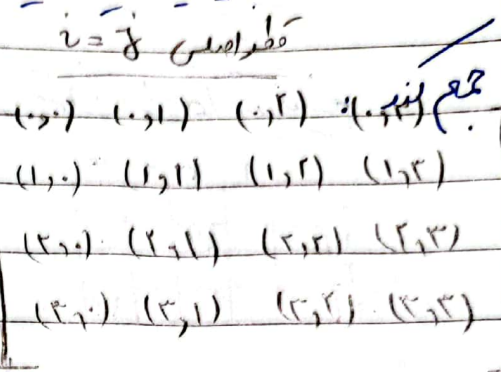
```
      }
```

MODERN



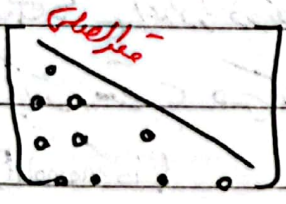
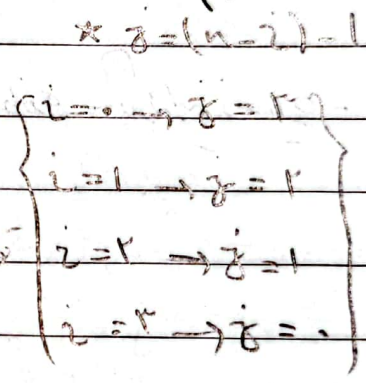
۳۱) برنامه‌ای بنویس که در یک ماتریس  $n \times n$  در نظر بگیرد و اعضای قطر اصلی را با هم جمع کند.

```
int sum = 0;
int m[i][j];
for (int i = 0; i < n; i++)
    for (int j = 0; j < n; j++)
        if (i == j)
            sum += m[i][j];
```



• اکنون مجموع اعضای قطر فرعی را حساب کنید:

```
int n = 4;
int j = 0;
int sum = 0;
for (int i = 0; i < n; i++)
    sum += m[i][j];
```



ماتریس بالا مثلثی: تمام ارقام های زیر قطر اصلی را صفر کنید:

```
if (i > j) m[i][j] = 0;
```

خطای درایم: عددی از یک اریبه در درستی جانمایی شود



Subject :  
 Year.      Month.      Date.

01.01.01 → تاریخ عددی

نوع اندازش برقرار است :  
 (اندازش)

A → 65  
 B → 66  
 C → 67  
 :  
 Z →  
 a = 97  
 b = 98  
 c = 99  
 :

char ch = 'A'  
 cout << int(ch) →

تبدیل دستوری نوع :

char (عدد) → تاریخ

خطاها: ① خطای زمان با صوابی : z x if

② خطای زمان اجرا : مثل تقسیم بر صفر یا خطای overflow

{ char str[100];

برنامه ای بنویسید که رشته ای را از ورودی بخواند و تعداد

int i, L=0, h=0, S=0;

cout << "enter string:";

تعداد حرف بزرگ، تعداد حرف صدادار و تعداد

gets(str);

for(i=0; str[i] != '\0'; i++)

انعام موجود در رشته را چاپ کند :

{ if (str[i] == 'a' || str[i] <= 'z')

L++;

if (str[i] >= 'A' || str[i] <= 'Z')

h++;

if (str[i] == 'A' || str[i] == 'a' || str[i] == 'E' || str[i] == 'e' ||

str[i] == 'I' || str[i] == 'i' || str[i] == 'o' || str[i] == 'o')

MODERN || str[i] == 'u' || str[i] == 'u')

S++

}



```

cout << "Lower: " << l << endl;
cout << "Higher: " << h << endl;
cout << "Sedade: " << s << endl;
cout << "tedadi: " << i << endl;
}

```

اطراف برابری:   
 $h = \frac{l+s}{2}$    
 $l = \frac{l+h}{2}$    
 $s = \frac{s+h}{2}$    
 $i = \frac{i+h}{2}$

حرف صواب: A, E, I, O, U

توانع آماره C++ و sqrt

تابع و function

توانع ساخت کاربر

توانع ساخت کاربر از دو جنبه نام و بدنه ساخته می شوند.

نوع تابع (باید مشخصا) نام تابع

```

{ return ....
}

```

نام قبل از main نوشته می شود.

```

int sq(int x)
{ return x*x;
}

```

```

#include <iostream>
#include <iomanip>

```

```

int fact(int n)
{ int f=1;
  for(int i=1; i<=n; i++)
  f*=i;
}

```

```

main() {
  int z=fact(5);
  cout << z;
  cout << endl << fact(7);
}

```

MODERN



Subject :

Year. Month. Date.

۳۳) تابع بنویسید که عددی را به عنوان ورودی بگیرد و اگر عدد اول باشد، ۱ برگرداند و در غیر این صورت ۰ برگرداند.

```
int cube (int n)
{
    return xxxxn;
}
```

```
main ( )
{
    cout << cube (۳);
}
```

return. در آنجا اختیاری / Void نوع برگشتی ندارد.

```
Void cube (int n)
{
    cout << xxxxn;
} main ( )
```

```
{ cube (۳);
}
```

۳۴) تابع بنویسید که عدد را به عنوان ورودی بگیرد و اگر عدد اول باشد، ۱ برگرداند و در غیر این صورت ۰ برگرداند.

```
int aVal (int n)
{
    int flag = 0;
    if (n <= 1)
    {
        return -1;
    }
    else {
```

اگر عدد اول باشد، ۱ برگرداند و در غیر این صورت ۰ برگرداند.

```
for (int i = 2; i <= n/2; i++)
{
    if (n % i == 0)
    {
        flag = 1;
        break;
    }
    if (flag == 0)
    {
        return 1;
    }
    else
```

```
main ( )
{
    int n;
    cin >> n;
    cout << AVal (n);
}
```

```
return 0;
}
```





```
void cube (int x)
```

```
{ cout << x * x * x << endl;
```

```
}
```

```
main ( )
```

```
{ int x = 1...
```

```
cube (x);
```

```
}
```

متغیری که در داخل تابع استفاده

یا ارجوند

این متغیر عملی آن

در سر اسیری

تابع می باشد و مقدار آن در داخل

یا ارجوان

همان تابع قابل استفاده خواهد بود

```
# include <iostream>
```

```
# include <iomanip>
```

```
:
```

```
int tavan (int x, int y)
```

```
{ int temp = 1;
```

```
for (int i = 1; i <= y; i++)
```

```
{ temp * = x;
```

```
}
```

```
return temp;
```

```
}
```

```
main ( ) {
```

```
int x, y;
```

```
cin >> x;
```

```
cin >> y;
```

```
cout << tavan (x, y);
```

```
}
```

تازه می توان نوشتیم دوباره

راستش با این می توان اول را نوشت

روش برساند



Subject :

Year.

Month.

Date.

ارسال با مقدار call by value

ارسال بر پایه مقدار

ارسال با ارجاع call by reference

ارسال به طریق مقدار ثابت می شود که متغیر همان برابر اصلی با main از تعریف اصل تابع

بسیار همانند اما بعضی مواقع می خواهیم محتویات متغیر فرستاده شده به تابع دسترس شود،

در این حالت از طریق ارجاع استفاده می کنیم. این کار باعث می شود تا جای اینده

لیستی که از آن ارجاع ایجاد کند، خود ارجاع می شود. هر تغییری که روی

با مقدار در تابع صورت گیرد روی برابر اصلی نیز انجام خواهد شد.

تفاوت ارسال از طریق مقدار و ارسال از طریق ارجاع:

ارسال از طریق مقدار	ارسال از طریق ارجاع
<code>int x;</code>	<code>int &amp;x;</code>
با مقدار <code>x</code> یک متغیر محلی است.	با مقدار <code>x</code> یک ارجاع است.
<code>x</code> یک کپی از آرگومان است.	<code>x</code> مترادف با آرگومان است.
تغییر محتویات آرگومان ممکن نیست.	می تواند محتویات آرگومان را تغییر دهد.
آرگومان ارسال شده از طریق مقدار می تواند یک ثابت باشد.	آرگومان ارسال شده از طریق ارجاع فقط باید یک متغیر باشد.
آرگومان فقط خواندنی است.	آرگومان خواندنی و نوشتنی است.

MODERN



$$P(n, k) = \frac{n!}{(n-k)!}$$

(۳۶) استفاده از مثال زیر برای یادگیری

$$P(5, 2) = \frac{5!}{(5-2)!} = \frac{5!}{3!} = 12$$

تعداد  $P(n, k)$  «کلیت» را می‌توانیم

# include <iomanip>

```
int fact (int n)
```

```
{ int f = 1;
```

```
for (int i = 1; i <= n; i++)
```

```
{ f * = i;
```

```
} return f;
```

```
} main ( ) {
```

```
int n, k, p;
```

```
// P(n, k)
```

```
cin >> n >> k;
```

```
P = ( fact (n) / fact (n-k) );
```

```
cout << P;
```

```
}
```

حرف : محدودیت برای ورودی استفاده از

تأثیرهای حرف دارد.

مثال برای برنامه مقابله «اعداد +»

نصف فقط اعداد + قبول نمی‌کند

صورت وارد کردن اعداد - را اجرا شود.

```
int cinPos ( )
```

```
{ while (true) {
```

```
cin >> n;
```

```
if (n >= 0)
```

```
return n;
```

```
else
```

```
cout << " n is not positive Plz
```

```
enter again: ";
```

```
}
```

حلقه while

```
while (true)
```

```
for ( ; ; )
```

MODERN



Subject :

Year.

Month.

Date.

```
int fact (int n)
```

در صورتی که برای اعداد بزرگتر

```
{ int F=1;
```

```
for (int i=1; i<=n; i++)
```

اصولاً حاصل است :

```
{ F *= i;
```

```
} return F;
```

```
} int cinPos ( )
```

```
{ int x;
```

```
while(true){
```

```
cin >> x;
```

```
if (x <= 0)
```

```
return x;
```

```
else
```

```
cout << " Plz Enter Positive Number: " << endl;
```

```
}
```

```
} main ( ) {
```

```
int n;
```

```
n = cinPos ( );
```

```
cout << fact (n);
```

```
}
```



آشاره کرد: نوعی متغیر چندین عدد خود را می‌تواند دارد. خود را می‌تواند یکی اشاره می‌کنند.

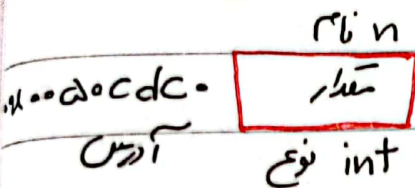
• حافظه کامپیوتر را می‌توان تصور کرد که از یک بزرگ زنجیره از بیت‌ها تشکیل شده است. برای این کامپیوتر را می‌توان تصور کرد.

بهر درجه آدرس ۰ تا ۲۵۵ مه‌باشد. ۳ و ۴ برای آسانی به متغیر داده می‌شود.

نوع متغیر، نام متغیر و آدرس حافظه. صلاً برای آسان  $n$   $int$  نام  $n$  آدرس

چند خانه از حافظه به مقدار  $n$  در این قرار می‌گیرد به یادگیری مرتبط هستند.

بیشتر کامپیوتر نوع  $int$ ، ۴ بایت (۸ بیت) حافظه اشغال می‌کند.

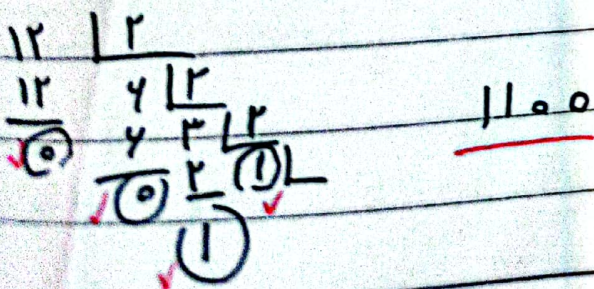


صنایعها: ۲ و ۳ و ۴ و ۵ و ۶ و ۷ و ۸ و ۹ و ۱۰ و ۱۱ و ۱۲

۲: صبیحانینی  $octal$ : ۸  $hex$ : ۱۶ (دهی فال (D): ۱۰  $dec$ : ۱۰  $hex$ : ۱۶: گندادسی مال

نخستین صبیحانینی حافظه صبیحانینی گندادسی مال (۱۶):

(A) ۱۰، (B) ۱۱، (C) ۱۲، (D) ۱۳، (E) ۱۴، (F) ۱۵



تقسیمات متوالی:



Subject :

Year. Month. Date.

$$\begin{array}{r} 174 \overline{) 14} \\ 14 \phantom{0} \\ \hline 0 \end{array}$$

معنای متوالی :

روش جدید : (معنای بانبری) :

$$2^{10} \quad 2^9 \quad 2^8 \quad 2^7 \quad 2^6 \quad 2^5 \quad 2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0$$

$$2.48 \quad 1.24 \quad 512 \quad 256 \quad 128 \quad 64 \quad 32 \quad 16 \quad 8 \quad 4 \quad 2 \quad 1$$

معنای باب ۲ :

$$(174)_{10} \rightarrow (101100)_{2} \quad / \quad (114)_{10} \rightarrow (111001)_{2}$$

$$(172)_{10} \rightarrow (10110000)_{2} \quad / \quad (24)_{10} \rightarrow (11000)_{2}$$

معنای ۲ به ۱۰ :

$$(110101)_{2} \rightarrow (53)_{10}$$

$$(11001)_{2} \rightarrow (15)_{10}$$

در روز بالا از چپ به راست ترجمه می شود

علامه انتهای و در صورت وجود ۱ در غیر این صورت

مفرد جانوری به سگ و پس از آن ۲ به ۱۰ از چپ به راست مفردین را در خود بودی مفردی

MODERN

و طبع هر سگ



صنای ۸ ← سه بیت.

حاصل بیت ۲ = ۸

عدد	۰	۱	۲	۳	۴	۵	۶	۷
صنا	۰۰۰	۰۰۱	۰۱۰	۰۱۱	۱۰۰	۱۰۱	۱۱۰	۱۱۱

سوم حاصل بیت ←  
 $(\frac{011}{1} \frac{001}{1})_2 \rightarrow (31)_8$

صنای ۸ ← اول بیت ۲ عدد ۸  
 $(\frac{11}{1} \frac{11}{1})_2 \rightarrow (10101010)_8$

یعنی صنای ۲ ← واسطه است.  
 $(25)_{10} \rightarrow (14)_8$

صنای ۸ ← اول بیت ۲ عدد ۸  
 $(25)_{10} \rightarrow (11001)_2 \rightarrow (14)_8$

صنای ۸ ← اول بیت ۲ عدد ۸  
 $(97)_{10} \rightarrow (1)_8$   
 غلط ✓ چرا؟ چون ۹ و ۷ صنای ۸ طاقی است  
 و باید کمتر از ۸ باشد.

$(27)_{10} \rightarrow (110111)_2$   
 $(27)_{10} \rightarrow (15)_8$



Subject :

Year.      Month.      Date.

میتا ۱۶ ← ۴ بیت ← ۲ بیت ← ۱ بیت ← ۱۶ ← ۲ = ۱۶

عدد	میتا	
۰	۰۰۰۰	
۱	۰۰۰۱	$(11011011)_2 \rightarrow (DB)_{16}$
۲	۰۰۱۰	$D \quad B$
۳	۰۰۱۱	
۴	۰۱۰۰	$(A2)H \rightarrow (10101010)_2$
۵	۰۱۰۱	
۶	۰۱۱۰	$(2B)H \rightarrow (43)_{10}$
۷	۰۱۱۱	اولی ۲ بیت ۱۰
۸	۱۰۰۰	$(2B)H \rightarrow (11010101)_2$
۹	۱۰۰۱	
A ۱۰	۱۰۱۰	$(24)_{10} \rightarrow (1A)H$
B ۱۱	۱۰۱۱	
C ۱۲	۱۱۰۰	$(2E)_{10} \rightarrow (110000)_2$
D ۱۳	۱۱۰۱	$(10101010)_2$
E ۱۴	۱۱۱۰	
F ۱۵	۱۱۱۱	

میتا ۱۶ بیت ۸ و برعکس:

$$(B15)H \rightarrow (2EFA)_{16}$$

$$(B15)H \rightarrow (1011010101010101)_2$$

d    E    ۲    A

$$(242A)_{10} \rightarrow (B1A)H$$

**MODERN**  $(2E2A)_{10} \rightarrow (1011010101010101)_2$

B    ۱    A



Subject :

Year.

Month.

Date.

• کوچکترین واحد در کامپیوتر : Bit ← □ ← صفر یا یک

Byte ← □□□□□□□□ ← 8Bit \* b ← بیت \* B ← بایت

$2^{10} = 1024$  KB کیلو بایت

$2^{30} = 1073741824$  GB گیگا بایت

$2^{20} = 1048576$  MB مگا بایت

$2^{40} = 1099511627776$  TB ترا بایت

• تغییر ارجاع در C++ برای بایت آدرس آدرس یک متغیر از عملگر ارجاع & استفاده می شود.

```

استفاده کنیم. (عبارة n)
{ int n = 44;
  cout << "n = " << n << endl;
  cout << "&n = " << &n << endl;
}

```

• در برنامه نویسی، n و rn هم‌معنی برای متغیر هستند.

```

int n = 44;
int &rn = n;
// rn is a synonym for n

```

• افزایش یا کم کردن rn نیز افزایش یا کم کردن n را در بر می گیرد.

```

--n;
cout << "n = " << n << ", rn = " << rn << endl;

```

```

rn * = 2;
cout << "n = " << n << ", rn = " << rn << endl;

```

MODERN



Subject :

Year. Month. Date.

```

{ int n = 44;
  int & m = 44; // ERROR: 44 is not a variable;
}

```

برنامه بالا! ارجاع ها در هنگام اعلان متغیرها اولی می شوند و فقط می توانند

متغیر قبول کند یعنی که بالا اشتباه است.

۲۴

۱۲

```
using namespace std;
```

```
void swap(int &x, int &y) {
```

```
  int temp;
```

```
  temp = x;
```

```
  x = y;
```

```
  y = temp;
```

```
} main()
```

```
{ int x = 10;
```

```
  int y = 12;
```

```
  cout << "x: " << x << " y: " << y << endl;
```

```
  swap(x, y);
```

```
  cout << "x: " << x << " y: " << y << endl;
```

```
}
```

توجه! عمل اشاره گر، برای اینج است. حافظه ی متغیر را برمی گرداند و به زمره ی نیمه بالا از اشاره گر

استفاده نمی. برای این منظور از عبارت "سپار" \* - برای اعلان متغیر استفاده نمی.

MODERN

(Pointer)



Subject :

Year.

Month.

Date.

```

int *x : این متغیر آدرس بران ذخیره می شود - اشاره کننده می شود - متغیر
{ int n = 44;
  cout << "n = " << n << " , &n = " << &n << endl;
  int *pn = &n ; // pn holds the address of n
  cout << " pn = " << pn << endl;
  cout << "&pn = " << &pn << endl;
}

```

- نام اشاره کننده اشاره : آدرس را برایش چاپ می کند و با اشاره ، مقبول است چاپ می کند.

• **آرایه پویا Dynamic Array** : در این اجزای برنامه (بهره از کامپایلر) اندازه را به دست می آوریم.

• **آرایه پویا** : آرایه پویا اعمال با تعریف کنیم.

```

int n;
cin >> n;
int *x = new int [n];
for (int i = 0; i < n; i++)
{ cout << "x[" << i << "]:";
  cin >> x[i];
} cout << "-----";
for (int i = 0; i < n; i++)
{ cout << "x[" << i << "]:" << x[i] << endl;
}
}

```

MODERN



Subject :

Year. Month. Date.

```
void display(int a[], int n)
```

۳۸ برنامه ای که در آن آرایه را بصورت عمود نمایش دهد

```
{ for(int i=0; i<n; i++)
  { cout << "a[" << i << "]: "
    << a[i] << endl;
  }
}
```

چاپ این آرایه از طریق تابع display انجام شود.

```
} main()
```

نکته! آوردن ارسال آرایه به روش ارجاع به

```
{ int n;
```

```
cin >> n;
```

تابع عمل کنیم مقدار آرایه ارسال و آرایه

```
int * a = new int [n];
```

```
for (int i=0; i<n; i++)
```

ظرف تابع این تغییر کند هر دو این آرایه

```
{ cout << "a[" << i << "]: "
```

```
cin >> a[i];
```

ها تغییر خواهند کرد.

```
} cout << "\n";
```

```
display(a, n);
```

```
#include <conio.h>
```

۳۹ برنامه ای بنویسید رشته از ورودی خوانده

```
#include <string.h>
```

```
void del(char s[])
```

و کلمات مابین آن ها را از رشته حذف کند.

```
{ int i, j;
```

```
for (i=0; s[i]; i++)
```

رشته در یک آرایه ای و حذف کلمات در تابع

```
for (j=i+1; s[j]; j++)
```

main - - - - -

```
{ if (s[i] == s[j])
```

int main() {

```
{ strcpy(s[j], s[j+1]);
```

const int n = 10;

```
j = i;
```

char s[n];

```
}
```

cin.get(s, n);

```
} MODERN
```

del(s);

getch();

return 0; }



const ← ثابت همدردی هم به هم تغییر نمی کنند.

$strlen(s)$  ← طول  $s$  را برمی گرداند.  
 $strcpy(s, t)$  ←  $t$  را در  $s$  کپی می کنند.  
 $strcat(s, t)$  ← عمل الحاق

توالی بازگشتی: توالی در داخل خودش، خودش را صدا می زند.

مانند عملی توالی به باها کار دریم به صورت غیر بازگشتی نوزده - انوش مرخو احم

$$f(n) = \begin{cases} f(0) = 1 & n < 1 \\ f(1) = 1 & n = 1 \\ f(n-1) + f(n-2) & n > 1 \end{cases}$$

توالی بازگشتی اصل توالی فیبوناچی

$f(5) = ?$

$f(5) = f(4) + f(3) =$

$f(3) + f(2) + f(2) + f(1) =$

$= f(2) + f(1) + f(1) + f(1) + f(1) + f(1) + 1 =$

$= f(1) + f(1) + f(1) + f(1) + f(1) + f(1) + f(1) +$

$f(0) + 1 = 8$

پس برنامه اعداد فیبوناچی را هم بصورت

توالی بازگشتی و هم بصورت غیر بازگشتی (۲۰)

فونستیم

برنامه را هم مقابله

```

#include <iostream>
#include <conio.h>
#include <string.h>
int fib(int n)
{ if(n==1 || n==0)
  return 1;
  else
  return fib(n-1)+fib(n-2);
}
main()
{ cout << fib(8);
}

```



Subject :  
Year. Month. Date.

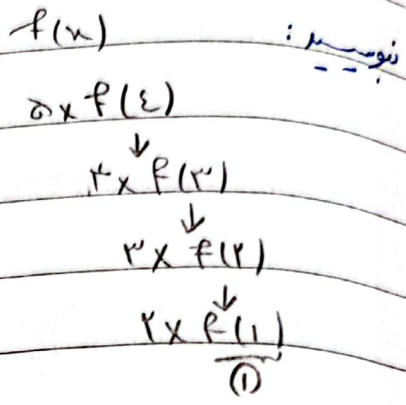
برنامه فاکتوریل را به صورت بازگشتی

```

int fact (int n)
{
    if (n == 1)
        return 1;
    else
        return (n * fact (n-1));
}

main () {
    cout << fact (5);
}

```



جمع اعداد ۱ تا n به صورت بازگشتی

```

int sum(int n)
{
    if (n == 1)
        return 1;
    else
        return (n + sum(n-1));
}

main () {
    cout << sum(5);
}

```

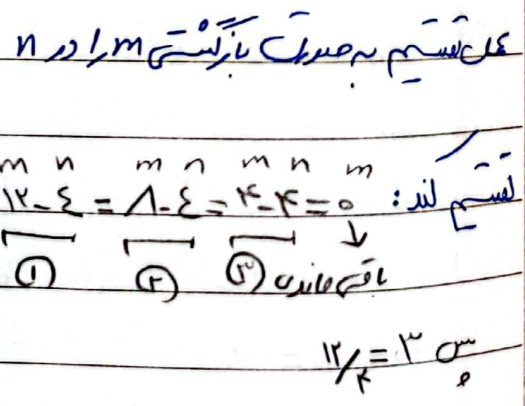
تابع نویسی در جدول استفاده

```

int div (int m, int n)
{
    if (m < n)
        return 0;
    else
        return 1 + div(m-n, n);
}

main () {
    cout << div(12, 4);
}

```



MODERN

برای آن فقط عمل جمع و تفریق استفاده



Subject :

Year.

Month.

Date.

```
int zarb (int m, int n)
```

۴۳ نامی نویسد که بدون استفاده از عملیات

```
{ if (n==1)
```

```
return m;
```

ضرب m را در n بصورت بازگشتی

```
else
```

```
return m+zarb (m, n-1)
```

ضرب کند:  $4 \times 2 = 11$

```
} main() {
```

$4 + z(4, 2)$

```
cout << zarb(4, 2);
```

$4 + z(4, 1)$

```
}
```

به ۴ هم جمع می شود و در نهایت ۱۱ را بر می گرداند

```
int mod (int m, int n)
```

۴۴ نامی نویسد که بدون عمل تقسیم

```
{ if (m < n)
```

```
return m;
```

باقی مانده m بر n بصورت بازگشتی چاپ

```
else
```

```
return mod (m-n, n);
```

کند:

```
} main() {
```

```
cout << mod (11, 3);
```

```
}
```

```
int tavan (int m, int n)
```

۴۵ نامی نویسد که توان را بصورت بازگشتی

```
{ if (n==1)
```

```
return m;
```

چاپ کند:  $2^5 = 2 \times 2 \times 2 \times 2 \times 2$

```
else
```

```
return m * tavan(m, n-1); }
```

```
main() {
```

```
cout << tavan(2, 5);
```

```
}
```

MODERN



Subject :  
Year. Month. Date.

مرتب سازی (Sorting) : در این مبحث خواهیم آموختیم اعداد بصورت صعودی یا نزولی مرتب

### Bubble Sort ← مرتب سازی حبابی

مثال : اعداد مقابل به صورت صعودی مرتب شوند :

۷ ۸ ۴ ۱ ۲

۷ ۸ ۴ ۱ ۲

① ۷ ۴ ۸ ۱ ۲

۷ ۴ ۱ ۸ ۲

۷ ۴ ۱ ۲ ۸

۴ ۷ ۱ ۲ ۸

② ۴ ۱ ۷ ۲ ۸

۴ ۱ ۲ ۷ ۸

۴ ۱ ۲ ۷ ۸

۱ ۴ ۲ ۷ ۸

③ ۱ ۲ ۴ ۷ ۸

۱ ۲ ۴ ۷ ۸

۱ ۲ ۴ ۷ ۸

در این صورت اعداد در خط اول  $7 > 8$  ؟ خیر ؛ پس

در این تغییر به مرحله بعد می رود. در خط دوم، آری

در  $7 > 2$  ؟ بله ؛ پس جای آن عوض می شود.

برنامه مثال مقابل :

```

main()
{
  int n = 5;
  int temp;
  int a[n] = {7, 8, 4, 1, 2};
  for (int i = 0; i < n; i++)
    for (int j = 0; j < n - i - 1; j++)
      if (a[j] > a[j + 1])
      {
        temp = a[j];
        a[j] = a[j + 1];
        a[j + 1] = temp;
      }
}
for (int i = 0; i < n; i++)
  cout << a[i] << " ";
}

```

MODERN





Subject :

Year.

Month.

Date.

## استدلال (رولود - struct) :

ما تاکنون هر چه می خوانیم هم نوع رسم ساختار بود. اکنون می خواهیم چند نوع

متغیر (مانند int, float, char, ...) را در کنار هم استفاده کنیم. در این صورت

int	long
string	

با struct با تعریف می کنیم: مثال :

#include <iostream> می خواهیم شماره دانشجویی، نام و نام خانوادگی

using namespace std;

struct student {

معدل دانشجویی و جای نهم :

int id;

char name[10];

می دانیم شماره دانشجویی از نوع int - نام و

char family[10];

نام خانوادگی از نوع char (رشته) و معدل

float avg;

};

main() {

struct student S1;

S1.id = 12;

strcpy(S1.name, "Amin");

strcpy(S1.family, "jalilzadeh");

S1.Avg = 19.2;

cout << S1.id << endl;

S1.name << endl << S1.family

<< endl << S1.Avg;

• struct قبل از main

تعریف می شود!

باید تعریف struct ← { و بعد از آن می توانیم از آن استفاده کنیم

همین طرز کار همانگونه می شود :

struct {  
?  
};

MODERN



Subject :

Year.

Month.

Date.

SI ای کے درجہ ہال میں تعریف کریم ← Avg, family, name, id

```
#include <iostream>
```

برنامہ اس بنویسید کہ مستحقان سے دانشجو

```
#include <cstdio>
```

```
using namespace std;
```

اہم از شماره دانشجو ہے، نام و خانوادہ معقول

```
struct student {
```

```
int id;
```

شمارہ تماس، از کار پر بلبر، اس میں مستحقان

```
char name[11];
```

```
char family[20];
```

دانشجو با بیشترین معقول کار کند:

```
float avg;
```

```
int phone; };
```

در برای دریافت پتہ، کارکن از cin استفادہ فرستیم و بجای آن از

```
main() {
```

gets استفادہ فرستیم:

```
struct student s1, s2, s3;
```

S2 و S3 نیز در صورتی : نام برام

```
cout << "s1 info\n";
```

نوشتہ می شوند تا زمانیکہ if برای

```
cout << s1.id << " ";
```

مستحق نام برام تعریف فرستیم:

```
cin >> s1.id;
```

```
if (s1.avg > s2.avg || s1.avg > s3.avg)
```

تہ!

```
cin.ignore();
```

```
cout << s1.id << endl << s1.name << endl <<
```

```
cout << "s1 name: ";
```

```
s1.family << endl << s1.avg << endl <<
```

```
gets(s1.name);
```

```
s1.phone << endl;
```

```
cout << "s1 family: ";
```

```
if (s2.avg > s1.avg || s2.avg > s3.avg)
```

```
gets(s1.family);
```

```
cout << s2.id << endl << s2.name << endl <<
```

```
cout << "s1 Avg: ";
```

```
s2.family << endl << s2.avg << endl << s2.phone << endl
```

```
cin >> s1.avg;
```

```
if (s3.avg > s2.avg || s3.avg > s1.avg)
```

```
cout << "s1 phone: ";
```

```
cout << s3.id << endl << s3.name << endl <<
```

```
cout << s1.phone;
```

```
s3.family << endl << s3.avg << endl <<
```

```
s3.phone << endl;
```



Subject :

Year.

Month.

Date.

نقشه برنامه مندرج در صورت عملی ignore و بعد از cin برابر اینج می توانیم از

صورت gets استفاده کنیم یا سین ignore کنیم. تا ما با این شخصیت

به صورت در اختیار تابع gets قرار نگیرد پس از دستور (cin.ignore) استفاده می کنیم.

۵۳

حجم ۱۵

s1	s2	s3	s4

آرایه ای از structure ها :

```
struct student {
```

```
int id;
```

```
char name[10];
```

```
char family[20];
```

```
float Avg;
```

```
};
```

```
main ()
```

```
{ student struct S[10];
```

دسترسی به عناصر آرایه از این طریقها:

```
S[0].id = 12
```

```
cout << S[0].id;
```

MODERN



Subject :

Year.

Month.

Date.

```
#include <io stream>
```

برنامه ای بنویسید که مشخصات دانشجو (۴۷)

```
using namespace std;
```

```
struct student {
```

(شماره دانشجویی - نام و نام خانوادگی و معدل) هر

```
int id;
```

```
char name [10];
```

دانشجو را از ورودی گرفته پس آنها را به صورت

```
char family [10];
```

```
float avg; };
```

در یک خط چاپ کند:

```
main()
```

```
{ int n;
```

برای

```
for (int i=0; i<n; i++)
```

```
cout << "enter number of student: ";
```

```
{
```

```
cin >> n;
```

```
cout << s[i].id << " ";
```

```
struct student s[n];
```

```
cout << s[i].name << " ";
```

```
for (int i=0; i<n; i++)
```

```
cout << s[i].family << " ";
```

```
{ cout << "s[" << i << "].id: ";
```

```
cout << s[i].avg << " ";
```

```
cin >> s[i].id;
```

```
cout << endl;
```

```
cout << "s[" << i << "].name: ";
```

```
}
```

```
cin.ignore();
```

```
}
```

```
gets (s[i].name);
```

```
cout << "s[" << i << "].family: ";
```

```
gets (s[i].family);
```

```
cout << "s[" << i << "].avg: ";
```

```
cin >> s[i].avg;
```

```
cout << endl;
```

```
} cout << "Id Name Family Avg\n";
```

MODERN

→ ادامه



Subject :

Year.

Month.

Date.

• سراسری تعریف کردن استرکچر: (فقط قبل از main)

این نخواهیم استرکچر را به صورت سراسری تعریف کنیم، به روشی که این عمل انجام می شود.

```

1.
struct student {
    int id;
    char name [10];
    char family [20];
    float Avg;
};
student s[2];
main ( ) {

```

```

2.
struct student {
    int id;
    char name [10];
    char family [20];
    float Avg;
} s [2];
main ( ) {

```

• استفاده از استرکچر در تابع:

```

};
void Printstruct (student a)
{
    cout << a.id << " ";
    cout << a.name << " ";
    cout << a.family << " ";
    cout << a.avg << " ";
    cout << endl;
}

```

من دو اسم استرکچر را به عنوان پارامتر

تابع در نظر گرفته و از آن استفاده کنیم:

```

} main ( ) { struct student s;
    s.id = 1;
    strcpy (s.name, "Amin");
    strcpy (s.family, "Jalilzadeh");
    s.avg = 19.2;

```

MODERN

cout << ...



Subject :

Year.

Month.

Date.

مقاله از جیب آرایه ها از استوار پی استوار پی

```
void Printstruct (struct student *a, int i)
```

```
{ cout << a[i].id << " " ;
```

```
  cout << a[i].name << " " ;
```

```
  cout << a[i].family << " " ;
```

```
  cout << a[i].avg << " " ;
```

```
  cout << endl ;
```

```
}
```

```
main ()
```

```
{ struct student s[2];
```

```
  for (int i=0; i<2; i++)
```

```
  { cin >> s[i].id;
```

```
    cin.ignore();
```

```
    gets(s[i].name);
```

```
    gets(s[i].family);
```

```
    cin >> s[i].avg;
```

```
  } cout << "Id name Family Avg\n";
```

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

```
    Printstruct (s, i);
```

```
  }
```

نکته! چرا \*a ؟ - چون s را به عنوان استرکچر قبول نکرد پس \*a

را به صورت اشاره تر تعریف می کنیم. (s نیز می تواند به جای و بنشیند).



Subject :

Year.

Month.

Date.

شی نواسی :

چهارمین جلدی بود که باره این از مشغلات برنامه نویسی راه حل داشت. این مضمون

از زبان فلسفه به جهان برنامه نویسی آمد و کمک کرد تا مفصلات تولید و پشتیبانی

نرم افزار کمتر شود. اسبابی که توان با توجه به مشغلات فرقی که آنها داشته اند

کرد. در کتب شی نواسی به دسته ها «لاسی» می نویسند و نمونه های هر کدام

«لاسی» گفته می شود. مشغلات هر شی را «صفت» می نامند و به زبان های

هر شی «لامده» می نویسند.

برنامه نویسی شی گدا بر سه ستون استوار است :

الف. سبب بندی : یعنی اینکه داده های مرتبط با هم ترکیب شوند و فرسایات ساده سازی

محقق شود.

ب. وراثت : در زبان واکس، وراثت به این معناست که شی وقتی متولد می شود،

خصوصیات و ویژگی های را از والد خود جراثت می گیرد.

MODERN



Subject :

Year.

Month.

Date.

امتیاز وراثت در این است که از داده‌های مشترک استفاده می‌شود و علاوه بر

موتور از داده‌های قبلی استفاده مجدد کرده، زیرا نیز صرفه‌جویی شده و استفاده منطقی

برنامه هم افزایش می‌یابد.

ع. چند معنی: که به آن چند شکلی هم می‌گویند به معنای یک چیز بود و چند شکل داشتن

است. چند بخشی بسته در وراثت معنا پیدا می‌کنند.

اعضای کلاس‌ها :

که زیر اعلان یک کلاس نشان می‌دهد: class Ratio

{ Public:

void assign(int, int); اعلان کلاس با یک طریقی class شروع

void Print ();

Private: هر شود و سپس نام کلاس می‌آید.

int num, den;

};

اعضای اعلان کلاس درون یک بلوک انجام می‌شود و سرانجام یک سفیدکولون بعد از

بلوک نشان می‌دهد که اعلان کلاس با یک نایفتم است.

در هنگام تعریف شیء object از کلاس سازنده استفاده می‌کنیم:

MODERN

class → object → ویرجی باصفت





Subject :

Year.

Month.

Date.

عبارت **Public** و عبارت **Private**. هر عضو که زیر عبارت **Public** اعلان شود.

یک « عضو عمومی » محسوب می شود و هر عضوی که زیر عبارت **Private** اعلان شود

یک « عضو خصوصی » محسوب می شود.

### سازنده ها :

وظیفه تابع سازنده این است که حافظ لازم را بر اساسی جدید تخصیص داده و آن را

مقدار دهی نماید و با اجرای وظایفی که برای تابع سازنده منظور شده استی جدیدی برای

استفاده آماده کند. هر صلاص می تواند چندین سازنده داشته باشد. در صورتی

تابع سازنده می تواند چند شکلی داشته باشد.

{ Public:

Ratio () سازنده ها، از طریق فهرست پارامترها می توانند

{ num=0; den=1; }

Ratio (int n) از طریق تالیف می شوند. افزودن چند تابع سازنده

{ num=n; den=1; }

Ratio (int n, int d) { num=n; den=d; } « Ratio به سبب

Void Print () { cout << num << '/' << den; }

Private:

int num, den;

};

MODERN



Subject :

Year. Month. Date.

main.cpp

```
class sum
```

برای این نویسی در استفاده از class و ...

```
{ public:
```

```
sum() {
```

نامی با نام sum داشته باشد - به صورت سینتکس

```
start=1; end=100; }
```

```
sum(int n) {
```

اعداد را تا n جمع کند

```
start=0; end=n; }
```

```
sum(int n, int m) {
```

در تابع سازنده یا اکتوری باشد از ابتدا

```
start=n; end=m; }
```

```
void print()
```

برای هر مورد فقط جمع کند - و در تابع سازنده

```
{ int s=0;
```

```
for(int i=start; i<=end; i++)
```

```
s+=i;
```

```
cout<<s;
```

برای هر مورد چاپ کند :

```
}
```

```
private:
```

```
int start, end;
```

```
};
```

```
main()
```

```
{ // sum obj;
```

```
// sum obj(50);
```

```
sum obj(1, 10);
```

همه از این صواب است

```
obj.print();
```

باید بخواهیم

```
}
```

MODERN



Subject :

Year. Month. Date.

```
using namespace std;
```

```
class myclass {
```

```
public: // Public access specifier
```

در تعریف کلاس زفانس

• دسترسی به عناصر کلاس :

```
int x; // Public attribute
```

```
void change(int a)
```

تغییری به صورت عمومی تعریف شده باشد

```
{ y = a;
```

```
} void Print()
```

نمی توان مستقیماً به آن دسترسی پیدا کرد

```
{ cout << y << " " << x;
```

برای این کار می توانیم

```
private: // Private access specifier
```

```
int y; // Private attribute
```

مبداً (خاص) بصورت

```
};
```

```
int main() {
```

Public عمومی تعریف کنیم و از آن جا به

```
myclass myobj;
```

```
myobj.x = 25; // Allowed (Public)
```

بعضی از به دسترسی

```
// myobj.y = 50; // Not Allowed (Private)
```

```
myobj.change(50);
```

دسترسی داریم

```
myobj.Print();
```

```
return 0;
```

```
}
```



## Encapsulation (کپسول سازی) :

منظور از کپسول سازی یعنی اینکه داده‌های یک کلاس از طریق متدها پنهان هستند.

برای پنهان کردن متغیرها و متدها در کلاس‌ها و دسترسی به آن‌ها از طریق متدها.

متغیر آن‌ها از get و set استفاده می‌کنیم.

set → برای مقداردهی به اعضای خصوصی کلاس

get → برای دسترسی به اعضای خصوصی کلاس

مثال :

```
using namespace std;
class Employee {
public: // Set
    void setSalary (int s) {
        salary = s;
    } // get
    int getSalary () {
        return salary;
    }
// Private attribute
    int salary;
};
int main () {
    Employee obj;
// obj.salary = 5000; error because access from main to
// Private attribute is not true
    obj.setSalary (5000);
    cout << obj.getSalary ();
    return 0;
}
```

MODERN



Subject :

Year.      Month.      Date.

نابودگر یا نابود کننده : نشان ← ~

وقتی که یک شیء ایجاد می‌شود تابع سازنده به خود خودکار برای ساختن آن فراخوانی

می‌شود : وقتی که شیء به پایان زندگی اش برسد تابع عضو ریزی به صورت خودکار

فراخوانی می‌شود تا نابود کردن آن شیء را مدیریت کند. این تابع عضو،

نابود کننده نامیده می‌شود.

- سازنده و حذف‌دهنده باید با هم کار کنند تا اطمینان حاصل شود که همه منابع آزاد شوند

و آن منابع آزاد شوند.

• هر دو فقط یک نابود کننده دارد.



Subject :

Year.      Month.      Date.

```
using namespace std;
```

مَدَنی پور لنگه به همران و سببش به اعطای سبب

```
class Line {
```

```
public:
```

در خارج از main ::: ل

```
void setlength(double len);
```

```
double getlength(void);
```

```
Line(); // This is the constructor declaration
```

```
~Line(); // This is the destructor : declaration
```

```
private:
```

```
double length;
```

```
};
```

```
// member functions definitions including constructor
```

```
Line::Line(void) {
```

```
cout << "object is being created" << endl;
```

```
} Line::~~line(void) {
```

```
cout << "object is being deleted" << endl;
```

```
}
```

```
void Line::setlength(double len) {
```

```
length = len;
```

```
}
```

```
double Line::getlength(void) {
```

```
return length;
```

```
}
```

لر دامه برنامه صفحه بعد :

```
// main function for the program
```

**MODERN**



Subject :

Year.

Month.

Date.

```
int main () {  
    Line line;  
    // Set line length  
    line.setLength (6.0);  
    cout << "Length of line : " << line.getLength () << endl;  
    return 0;  
}
```

مثال از وراثت :

```
using namespace std;  
// Base class  
class Vehicle {  
public:    string brand = "Ford";  
    void honk () {  
        cout << "tuut , tuut ! \n";  
    }  
};  
// Derived class  
class car : public Vehicle {  
public:    string model = "Mustang";  
};
```

```
int main () {  
    car my car;  
    my car.honk ();  
    cout << my car.brand + " " + my car.model;  
    return 0;  
}
```

MODERN



Subject :

Year. Month. Date.

```
using namespace std;
class univercity {
public:
    void setuni (int id, char* name, char* add, int tell) {
        Vid = id;
        strcpy(Uni Name, name);
        strcpy(Uni Address, add);
        Phone = tell;
    }
    void getuni (void) {
        cout << Vid << " " << Uni name << " , " << Uni Address << " " << Phone ;
    }
private:
    int Vid;
    char Uni Name [20];
    char Uni Address [50];
    int Phone;
};
class dars {
public:
    void setdars (int id, char* name, int unitt, float score) {
        Did = id;
        strcpy(DName, name);
        unit = unitt;
        grade = score;
    }
    void getdars (void) {
        cout << endl << Did << " " << DName << " " << unit << " " << grade ;
    }
}
MODERN
```

برای نویسنده مشخصات دانشگاه

ایم از شماره دانشگاه - نام زنا خانوادگی - نمره !

تلفن ریس در دفتر مدیر - مشخصات دانشگاه

ایم از (شماره دانشگاه - نام - آدرس - شماره تلفن)

ایم از (شماره ریس - نام آدرس - واحد و نمره)

تلفن دانشگاه - مشخصات قبول شده و نام ریس ها در تمام اول

برای تهیه است - استفاده از سی لایسنس و مفهوم

کپیوم سازی و وقت این مسئله را حل کنید

ادامه صبحی بعد:



Subject :

Year.      Month.      Date.

```
s1.get uni ( );  
s1.get std ( );  
s1.get dars ( );
```

ادامه برنامه بعد :

float avg;

}; main()

{ student s1;

s1.set uni (123, "sahand", "tabriz", 91422);

s1.set std (2325, "fereshteh", "dadashzadeh", 70);

s1.set dars (1, "Programming", 3, 19);

ادامه صفحه بعد :

MODERN

