

Subject.

Year. Month. Date. ()

موضوع: برنامه نویسی زبان C++ ، دیتیل و دیتیل
مقدمه ، حل تمرین و الزاماتی

تمرین ۱ شماره

پرودته ۲ استادهزادرام ۱ شماره

میلاد اتم ۱ شماره

پیداوت اتم ۱ شماره

13... 42774

14... 593419

15... 274027

زبان C++

Load base

add over

store base

آریتیکال کد های زبان C++

زبان اصلی

اصول
مقدم

زبان C++

base = base + over; C++

حزبان C++ در کامپایلر کامپایلر زبان C++

قبل از زبان C در زبان وجود داشته
B ←

نوع وجودی است و متغیرها به نوع بودن متغیر type
زبان C تبدیل شده

سازمانده زبان C (استاندارد) کامپایلر کامپایلر زبان C

Subject:

Year: Month: Date: ()

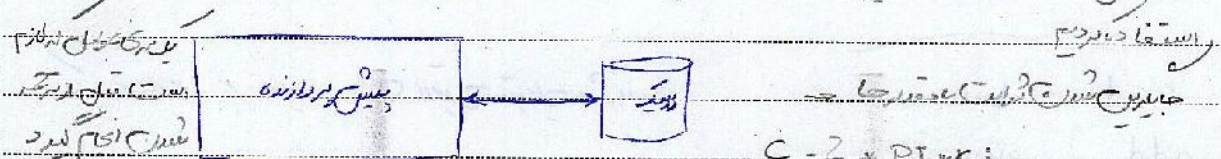
در این بخش به بررسی فرآیند تبدیل کد منبع به فایل اجرایی می‌پردازیم. این فرآیند شامل مراحل مختلفی است که در ادامه خواهیم دید.

در این بخش به بررسی فرآیند تبدیل کد منبع به فایل اجرایی می‌پردازیم. این فرآیند شامل مراحل مختلفی است که در ادامه خواهیم دید.

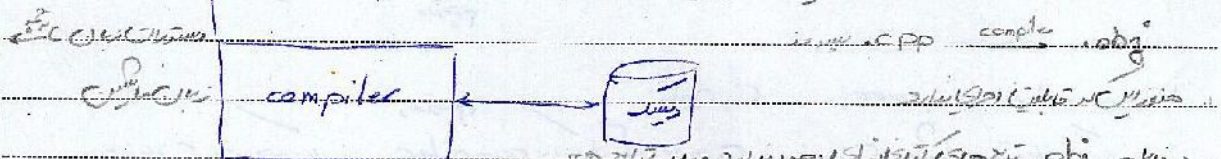
این فرآیند شامل مراحل مختلفی است که در ادامه خواهیم دید. در این بخش به بررسی فرآیند تبدیل کد منبع به فایل اجرایی می‌پردازیم.



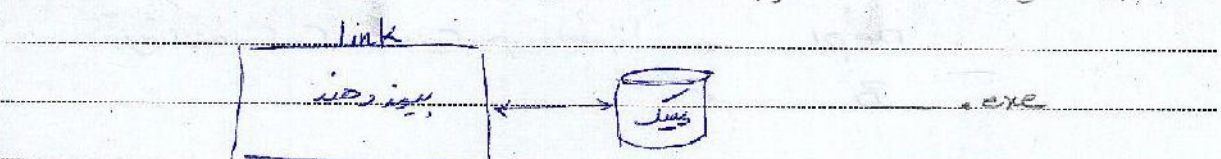
در این بخش به بررسی فرآیند تبدیل کد منبع به فایل اجرایی می‌پردازیم. این فرآیند شامل مراحل مختلفی است که در ادامه خواهیم دید.



در این بخش به بررسی فرآیند تبدیل کد منبع به فایل اجرایی می‌پردازیم. این فرآیند شامل مراحل مختلفی است که در ادامه خواهیم دید.

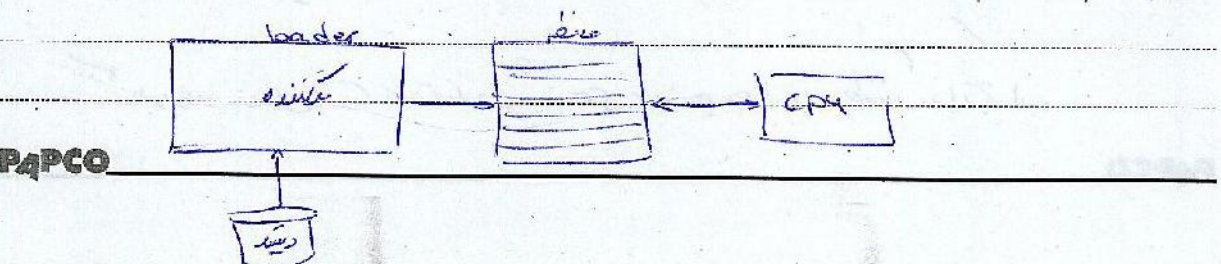


در این بخش به بررسی فرآیند تبدیل کد منبع به فایل اجرایی می‌پردازیم. این فرآیند شامل مراحل مختلفی است که در ادامه خواهیم دید.



در این بخش به بررسی فرآیند تبدیل کد منبع به فایل اجرایی می‌پردازیم. این فرآیند شامل مراحل مختلفی است که در ادامه خواهیم دید.

در این بخش به بررسی فرآیند تبدیل کد منبع به فایل اجرایی می‌پردازیم. این فرآیند شامل مراحل مختلفی است که در ادامه خواهیم دید.



Subject:

Year. Month. Date. ()

جای اولی بر بنام دو صفی زایش

First C++ program

این کد را در فایل main.cpp ذخیره کنید

```
/* This is a test */
```

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

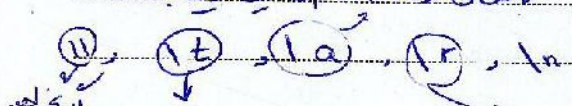
```
int main ( )
```

```
cout << "Hello! \n";
```

```
return 0;
```

```
{
```

```
}
```



این دو حرفی که قرار داده ایم چه کاره است؟
 این کاره است که در خروجی یک خط جدید ایجاد کند.
 این کاره است که در خروجی یک تب ایجاد کند.

توجه کنید که در خروجی یک خط جدید ایجاد می شود.
 این کاره است که در خروجی یک تب ایجاد کند.

```
Hello! \n world!
```


Subject:

Year. Month. Date. ()

چیزی که main return می‌دهد به سیستم عامل برمی‌گردد

اگر در return چیزی را برمی‌گردانیم و در جایی که return شده است تابع فراخوانده می‌شود (جوابی می‌دهد)

همان از cout استفاده می‌کنیم و از cin هم استفاده می‌کنیم

جایی که برای تغییر چیزی داریم، صاف می‌نویسیم و نیازی به تغییر دیگری نیست

```
# include <iostream>
```

برای تغییر و تغییر در سیستم استفاده می‌کنیم

```
int main ( )
```

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

اینجا در cout و cin استفاده می‌کنیم

```
cout << "Enter two integers: ";
```

```
cin >> i;
```

اینجا در cout و cin استفاده می‌کنیم

```
cin >> j;
```

```
sum = i + j;
```

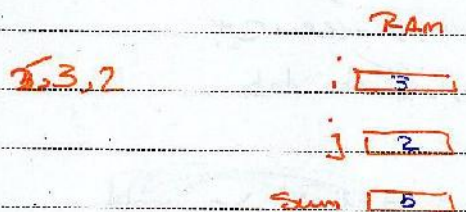
نتیجه حاصل جمع را می‌نویسیم

```
cout << "sum = " << sum << "\n";
```

عبارت خطی می‌نویسیم

```
return 0;
```

```
}
```



Subject:

Year. Month. Date. ()

اعمال ریاضی:

ضرب * $a * b$

تقسیم / a / b

جمع + $a + b$

تفریق - $a - b$

بازمانده % $a \% b$

مثال: $3 * 2 = 6$, $3 / 2 = 1.5$, $5 + 2 = 7$, $5 - 2 = 3$, $5 \% 2 = 1$

در عملیات ترتیبی از چپ به راست عملیات انجام می‌دهیم و اگر پرانتز باشد اول آن عملیات انجام می‌دهیم.

مثال: $(2 + 3) * 4 = 20$

اولیاً دارد

$$a + b * c = a + (b * c)$$

عملیات اولویت دارد

مثال: $a * b + c$

مثال: $3 = 5$ false

مثال: $a = b$ true

مثال: $a < b$ true

مثال: $a < b$ کو صحت دارد

مثال: $a > b$ کو صحت ندارد

مثال: $a = b$ کو صحت دارد

Subject: _____

Year. _____ Month. _____ Date. () _____

<= کورجیکلر یا مساوی
>= بزرگتر یا مساوی

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

(شماره)

```
int main ( )
```

```
{ int a,b;
```

```
  cout << " Enter two integer: ";
```

```
  cin >> a >> b;
```

```
  if (a == b)
```

```
    cout << " a is equal to be \n";
```

```
  if (a <= b)
```

```
    cout << " a is equal or less than b \n";
```

```
  return 0;
```

```
}
```

() چپ پرانتز است اظہار حقایق است

+ / :

+ -

>> << سبب خروجی

< <= > >=

= & ! =

⊕ → اکتیو و کونکاتنیشن چپ پرانتز است a = b = c = 5

دستور اجرای شرطی (شرط) if $(a > b)$ \leftarrow دستور اجرای شرطی

```

dستور) if (a > b)
    Max = a;
    if (a > b)
    {
        Max = a;
        a = a + 1;
    }
  
```

مربک \rightarrow دستور اجرای شرطی

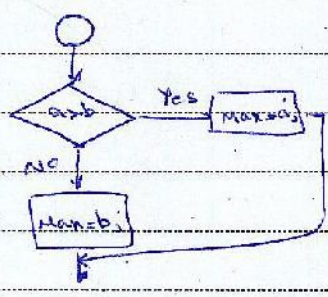
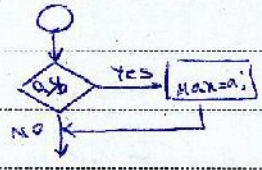
```

if (b > a)
  دستور 1
else
  دستور 2
  
```

```

dستور) if (a > b)
    Max = a;
else
    Max = b;
  
```

نمایش دستورات شرطی



Subject.

Year. Month. Date. ()

```
if (a > b)
```

```
if (c > a)
```

```
    Max = c;
```

```
else
```

```
    Max = a;
```

```
else if (c > b)
```

```
    Max = c;
```

```
else
```

```
    Max = b;
```

اگر اولی از آن‌ها بزرگتر باشد
اگر دومی بزرگتر باشد

اگر دومی بزرگتر باشد
اگر اولی از آن‌ها بزرگتر باشد

```
do { if (a > b)
```

```
    if (c > a)
```

```
        Max = c;
```

```
    }
```

```
else Max = a;
```

اینجا در صورتی که
اولی بزرگتر باشد
دومی بزرگتر باشد
اولی بزرگتر باشد

اینجا در صورتی که
اولی بزرگتر باشد
دومی بزرگتر باشد
اولی بزرگتر باشد

for ← while

while

do while

while (شرط)

بسته

اینجا در صورتی که
اولی بزرگتر باشد
دومی بزرگتر باشد
اولی بزرگتر باشد

```
do {
```

```
    int i = 0;
```

```
    int a = 0;
```

```
    while (i < 10)
```

```
        a = a + 2;
```

```
        i = i + 1;
```

```
    }
```

اینجا در صورتی که
اولی بزرگتر باشد
دومی بزرگتر باشد
اولی بزرگتر باشد

اینجا در صورتی که
اولی بزرگتر باشد
دومی بزرگتر باشد
اولی بزرگتر باشد

Subject:

Year. Month. Date. ()

۱- اوسط حساب کرنے کے لیے ہمیں ایک متغیر (sum) کی ضرورت ہے جس میں تمام اعداد جمع ہوں گے۔
 ۲- اعداد کو جمع کرنے کے لیے ہمیں ایک متغیر (counter) کی ضرورت ہے جس میں ہمیں اعداد کی تعداد بتانی جائے۔

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

```
int main (c)
```

```
{
```

```
int num, sum, counter = 0;
```

```
while (counter < 10)
```

```
{ cout << "Enter a number : "
```

```
cin >> num;
```

```
sum = sum + num;
```

```
counter = counter + 1 ;
```

```
}
```

```
cout << "Average = " << sum / counter << endl;
```

۱- ہمیں اعداد کو جمع کرنے کے لیے sum متغیر کی ضرورت ہے۔
 ۲- ہمیں اعداد کی تعداد کو جاننے کے لیے counter متغیر کی ضرورت ہے۔
 ۳- ہمیں اعداد کو پڑھنے کے لیے cin کی ضرورت ہے۔

```
Sum = sum / counter
```


Subject:

Year. Month. Date. ()

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

```
int main( )
```

```
{
```

```
int num, sumto, counter = 0;
```

```
cout << "Enter integers numbers, -1 to end" << endl;
```

```
cin >> num;
```

```
while (num != -1)
```

```
{
```

```
sum = sum + num;
```

```
counter = counter + 1;
```

```
cin >> num;
```

```
}
```

```
cout << "average = " << sum / counter << endl;
```

```
return 0;
```

```
}
```

```
counter = 0;
```

```
if (counter) << setprecision(2) << setiosflags(ios::fixed | ios::showpoint) <<  
cout << "average = " << static_cast<float>(sum) / counter << endl;
```

```
else
```

```
cout << "average = 0" << endl;
```

```
return 0;
```

```
}
```

Q.21

Conversion - default C++ 13.5

2.15.1

Subject :

Year . Month . Date . ()

تبدیل نوع (cast)

برای تبدیل یک نوع به نوع دیگر در C باید از `type cast` استفاده کرد

```
j = (float) i
```

نوع متغیر j به نوع متغیر i تبدیل می شود

تبدیل نوع در C

```
float j = static_cast<float>(i)
```

نوع متغیر j به نوع متغیر i تبدیل می شود

دستگاه اندازه گیری جریان

set precision

تعداد ارقام اعشاری را مشخص می کند

set iosflags

در C++ iosflags شامل ios::fixed و ios::scientific است

ios::fixed یا ios::scientific

```
set iosflags (ios::fixed | ios::showpoint)
```

نمایش نقطه اعشاری را فعال می کند

در C++

Subject:

Year. Month. Date. ()

برای هر عددی که در بازه 1 تا 100 قرار دارد

For حلقه

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

این حلقه را می توانیم به صورت دیگری هم بنویسیم

```
int i;
```

تغییر مقدار در ابتدای حلقه

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

انجام می شود

```
cout << "i=" << i << endl;
```

مقدور حلقه را هم می توانیم در یک خط بنویسیم

عدد 2 به عدد 1 می آید و 1 به 0 می آید
i=0, i=1, i=2, i=3, i=4, i=5, i=6, i=7, i=8, i=9, i=10

```
for ( int i=0, int j=0; i<10; i++, j+=2)
```

```
cout << "i=" << i << endl;
```

ی به 0 می آید و 0 به 1 می آید

```
for ( ; ; )
```

شرطی هیچ بار هم نشود true

```
cout << "test=" << endl;
```

```
int i=0;
```

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

```
cout << i;
```


Subject:

Year. Month. Date. ()

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

```
cout << i << " ";
```

برای چاپ اعداد 0 تا 9
کمی آسانتر است

while یا for

1. while

```
while (condition)
```

```
{
```

رشته
عبارة

```
}
```

2. while
for

برای حساب مبلغ بعد از n سال
 $A = P(1+r)^n$
amount = P * pow(1+rate, year)

year amount

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

حساب
ماتریس

```
int main()
{
    double amount, rate = 0.5, P = 1000.0;
    cout << "year" << setw(10) << "amount" << endl;
    for (int year = 1; year <= 10; year++)
    {
        amount = P * pow(1+rate, year);
        cout << year << setw(10) << setprecision(2) << setiosflags(ios::fixed) << setprecision(2) << amount << endl;
    }
}
```


Subject:

Year. Month. Date. ()

Set w در دستوری کننده جریان ^{جری} بهمانند علامه اول جاری شد و اگر چه جلوی سرد به سردی
را جواب می دهد یعنی می توانی بگویم که اندازگی آن را به خط می بندیم

یا حجم

از دست ^{ios} چه چیزی کند برای اینکه از صبح جواب بده

```
setw setflags (ios::left)
```

```
pow (x, y) = xy
```

مربوط به می خواهیم عملی را انجام دهیم در buffer بریزیم به buffer 20 است و ما در
کامل است 20 تا وارد کند و در دستوری که می بینیم در setw در دستوری که می بینیم
10 خزان

معنا و چند انتخابی (switch)

switch (تغییر انتخاب)

```
{
  case 1: دستورات ; break;
  case 2: دستورات ; break;
  case n: دستورات ; break;
  default: دستورات ; break;
}
```

به از دستورات break می توانیم از دستورات بعدی دور شویم و دستورات آن
اواخر تا آخر دستورات اجرا می شود

Default (تغییر انتخاب) اگر به دستورات هیچ کدام از case ها بر نرسد
در دستورات آخر می بینیم

Subject:

Year. Month. Date. ()

A, B, C, D, F

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

```
int main( )
```

```
{
```

```
int grade, a=0, b=0, c=0, d=0, f=0; Wahab
```

```
while( (grade = cin.get( )) != EOF) (ch, 2) C++
```

```
{
```

```
switch (grade)
```

```
{
```

```
case 'A': A++ break; Wahab
```

```
case 'a': a++; break; Case C++
```

```
case 'B': break; Case C++
```

```
case 'b': b++; break;
```

```
...
```

```
default: cout << "Input error" <<
```

```
{
```

```
}
```

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

```
return 0;
```

```
}
```


Subject:

Year. Month. Date. ()

Do while ← دو حلقه

```
do
{
    دو حلقه
} while (b2)
```

```
do
    دو حلقه
while (b2)
```

```
int sum = 0;
int num = 0;
do
{
    sum += num;
    cin >> num;
} while (num != -1);
```

دو حلقه ← break

```
int i = 0;
for ( ; i < (10; i++)
{
    if (i == 5)
        break;
    cout << "i = " << i << "n";
}
```


Subject: _____

Year. _____ Month. _____ Date. _____ ()

```

int i = 0;
for ( ; i < 10; i++)
{
    if (i == 5)
        continue;
    cout << i << "\n";
}

```

min ← continue

2. if (i == 5) stop → skip the rest of the loop

1. continue

```
if (a > b && a > c)
```

and and && and . 1

```
max = a;
```

```
if (a > b || c > b)
```

or or || or . 2

```
min = b;
```

```
if (!a)
```

```
if (a > b)
```

not ! not . 3

```
;
```

```
;
```

```
if (a < b)
```

طوری که اگر a و b مساوی باشند، true می شود

```
if (a == b) // a == b
```

در assignment ایسی می شود

true

false

Non-terminating

if, if, if ← true, false

توانیج

مادر برتشارت قنچه بیگ تاجی مانا روسته بایند
 کارهای الیپین کی برود در بایگ اولی سطل می شود و چند بار بایند و آنگاه هرگز برای زیج این سطل
 از توانیج استفاده در کی الیپین

توانیج قدرتی در در کی الیپین یک خردی بسیار است

به قدرتی نمی تواند بایند

در صحت بقیه توانیج بایند نوع خردی را مشخص کنیم

نوع بازگشتی توانیج
 ...
 توانیج پاره‌ها در در کی الیپین
 نوع پاره‌ها

توانیج قدرتی بلوغ پاره‌ها در در کی الیپین

```
int min ( int a , int b )
```

```
{
    if (a > b)
        return a ;
    else return b ;
}
```

قدرتی که توانیج باری الیپین در دستور
 return

استیمتور در زبان C از حروف و علامه underline به شکل شده
 یا علامه توانیج می شود

call یا فراخوانی توانیج

(نوع خردی در در کی الیپین) و اسم توانیج

min (x,y) ;

Subject:

Year: Month: Date: ()

تابع min در main تعریف شده و یک جا هم در main استفاده کنیم

```
int main ( )
```

```
{
```

```
int x=2, y=5;
```

```
int m;
```

```
m = min(x, y);      // این تابع در (مادهای شماره 2) در C++ تعریف شده
```

```
cout << m;
```

```
return 0;
```

دو زبان هستند برای هر تابع label از دسته‌ی بشود و بعد از اینکه از اجرای آن بشود اینند
cout (label) اجرای بشود و آن تابع اجرای بشود پس اجرای برنامه اجرا بشود

هر یک تابع را در main تعریف کنیم و در header تابع را در main تعریف کنیم

```
int a, b;      // Forward
```

```
int min (int, int)
```

در کامپایلر C++

تابع ریاضی کتابخانه

ceil(x) // گرد کردن به بالا ceil(9.3) = 10.0

floor(x)

گرد کردن به پایین floor(9.3) = 9

int

sin x cos x tg(x) pow(x, y) → x^y

log(x), ln(x), log₁₀(x), log₂(x), sqrt(x) √x, fabs(x) |x|

exp(x) = e^x

Subject

Year Month Date ()

انواع متغیرهای ابری آرایه

char

این متغیر برای اعداد و حروف

1 بیت

int

این متغیر برای اعداد صحیح (مثبت و منفی) و مقادیر کوچک دیگر
مثلاً 4 بیت

2 بیت

توصیف کننده

short

کوچکترین

long

بزرگترین

unsigned

بدون علامت

short int

int int ← 1 بیت

unsigned →

متغیرهای بدون علامت که فقط مقادیر مثبت را می‌توانند نگه دارند

این متغیرها برای اعداد صحیح مثبت و مقادیر کوچک دیگر استفاده می‌شوند

مثلاً 4 بیت (بزرگترین مقدار 255 است)

Float

2 بیت

توصیف کننده

double

float

4 بیت

تولید اعداد تصادفی

rand()

در کتابخانه stdlib.h

مقدار RAND_MAX

Subject:

Year. Month. Date. ()

$r = \text{rand}()$
 $a + r \% (b - a + 1)$

البرمجی ہمیں اعداد تصادفی میں a اور b کے درمیان

a b

اعداد میں اس وقت کوئی فرق

5000 بار اعداد تصادفی میں اعداد یک تا 4 دہا خصوصاً لیج جینڈر 1، 2، 3، 4، 5، 6، 7، 8، 9، 10

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

int main ()

{

int f1=0, f2=0, f3=0, f4=0, f5=0, f6=0, r;

for (int i=0; i<6000; i++)

$\text{rand}()$ تصادفی اعداد

$r = 1 + \text{rand}() \% 6;$

switch (r)

(time())

پہلے اس وقت تصادفی اعداد
کی ضرورت

تاکہ نتیجہ از 1990 تا 2010 آج

case 1: f1++; break;

case 2: f2++; break;

⋮

}

cout << "f1=" << f1 << "f2=" << f2 << ... << endl;

return 0;

}

subject:

Year. Month. Date. ()

نوع داده‌های عددی در C++ شامل integer و stand int است.
برای این اعداد تعدادی متغیر به نام متغیر عددی هم به کار می‌رود.

بزرگ‌ترین

enum status {CONTINUE, LOST, WON};

تعیین عدد متغیر است
که برای آن می‌تواند

status s;

0 اگر continue
وقتی که با انواع status تعیین می‌شود
lost, won, continue
چون دیگر نمی‌توانیم در آن تکرار دهیم

```
#include <iostream>
```

```
#include <string>
```

```
#include <time.h>
```

```
int RollDice ();
```

```
int main () {
```

```
{
```

```
enum status {CONTINUE, WON, LOST};
```

```
status s;
```

```
srand (time (0));
```

```
int r, RollDice (); e;
```

```
switch (r)
```

```
{
```

```
case 7:
```

```
case 11: s = WON; break;
```


Subject:
Year. Month. Date. ()

bagheri@cc-aut.ac.ir
or_bagheri@yahoo.com

```
case 2:  
case 3:  
case 12: S = LOST; break;
```

```
case 4:  
case 5:  
case 6:  
case 8:  
case 9:  
case 10: S = CONTINUE
```

Collection case continue
continue default

default: S = CONTINUE; exit;

```
}  
while (S == CONTINUE)  
{  
    r = rollDice();  
    if (r == 6)  
        S = WON;  
    else if (r == 7)  
        S = LOST;  
}  
if (S == WON)  
    cout << "WON" ;  
else (S == LOST)  
    cout << "LOST" ;  
return 0;  
}
```


Subject: _____

Year. Month. Date. ()

```
int RollDice ( ) :
```

```
{
```

```
  int r1, r2;
```

```
  r1 = 1 + rand() % 6;
```

```
  r2 = 1 + rand() % 6;
```

```
  return r1 + r2;
```

```
}
```


Subject:

Year. Month. Date. ()

$$Fib(0) = 0, \quad Fib(1) = 1$$

$$Fib(n) = Fib(n-1) + Fib(n-2)$$

DFS

```
int fib(int n)
```

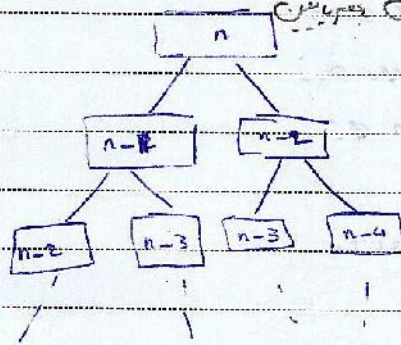
```
{
```

```
if (n == 0 || n == 1)
```

```
return n;
```

```
return fib(n-1) + fib(n-2);
```

```
}
```



Handwritten notes in Persian explaining the recursive process and stack usage.

inline

توابع

Handwritten notes in Persian explaining inline functions and their use in compile-time evaluation.

```
inline int sum(int a, int b) { return a+b; }
```

```
int main()
```

```
{
```

```
int x=2, y=3;
```

```
cout << "sum = " << sum(x,y) << endl;
```

```
return 0;
```

```
}
```

نوعی فراخوانی توابع

۱- فراخوانی استاندارد

Handwritten notes in Persian explaining function calls and stack frames.

Subject:

Year. Month. Date. ()

```
int f (int a)
```

فرضاً که به مقدار

```
{
```

```
cout << a++;
```

5

```
cout << a;
```

```
return a;
```

```
}
```

```
int main ()
```

خوبی

```
{
```

5

```
int x = 5;
```

x 5

5

```
cout << x;
```

6

```
f(x);
```

5

```
cout << x;
```

```
return 0;
```

```
}
```

فرضاً که به مقدار

فرضاً که به مقدار

```
int f (int &a)
```

```
{
```

چون که به مقدار 5 در دسترس است و در دسترس است

```
cout << a++;
```

5

```
cout << a;
```

5

```
return a;
```

```
}
```

خوبی

```
int x = 5;
```

5

```
cout << x;
```

5

```
f(x);
```

6

```
cout << x;
```

6

```
return 0;
```


Subject.

Year. Month. Date. ()

سفر صای به کلاس و تدریس در کلاس

```
int main,
```

```
{
```

```
int x;
```

اگرچه در اینجا x را به صورت عددی تعریف کرده ایم

```
int &a=x;
```

اینکه

```
a=2;
```

```
x=3;
```

a [2] ~ b = 3

```
cout << a;
```

```
return 0;
```

```
}
```

```
int x=2;
```

a [2]

```
int y=3;
```

y [3]

```
int &a=x;
```

در اینجا ما به متغیر x اشاره می‌کنیم و در حافظه آن را می‌بینیم

```
a=y;
```

در اینجا ما به متغیر y اشاره می‌کنیم و در حافظه آن را می‌بینیم

```
cout << x << a << y;
```

در اینجا ما به متغیر x اشاره می‌کنیم و در حافظه آن را می‌بینیم

Subject:

Year. Month. Date. ()

```

int
int f (int a=2, int b=3)
{
  cout << a << b << endl;
  return 0;
}

```

داده ها یا پارامترها مقدر میشوند و در صورتی که مقدار داده نشده باشد، مقدر میشود.

✓ int (a, b=3)
 ✗ int (a=2, b)

```

int main()
{
  cout << f(5,7);
  cout << f(10);
  cout << f();
  return 0;
}

```

در این مثال، مقادیر 5 و 7 برای پارامترهای a و b در تابع f استفاده شده است.

در خط دوم، فقط مقدار 10 برای پارامتر a در نظر گرفته شده است و مقدار b از مقدر (3) استفاده می‌شود.

در خط سوم، هیچ مقدار داده نشده است و مقادیر 2 و 3 از مقدر استفاده می‌شوند.

این مقادیر پیش از آنکه به کامپایلر C++ برسد، در حافظه قرار می‌گیرند.

error

```
int x = 3
```

```

int main()
{
  int x=2;
  cout << x;
  cout << x;
  return 0;
}

```

در اینجا، متغیر x در محدوده main تعریف شده است. اگر در یک فایل دیگر متغیری با نام x تعریف شود، آن متغیر global است و در تمام فایل‌ها قابل دسترسی است.

در اینجا، مقادیر 2 و 3 برای x در نظر گرفته شده است.

Subject:

Year. Month. Date. ()

تقریباً فصل ۲
۲۹
۲۹

در C++ می توانیم چند تابع هم نام داشته باشیم (Function overloading) که اینها را می توانیم
فقط با اضافه کردن آرگومان

int sum(int x, int y);

float sum(float x, float y);

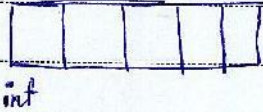
sum();

در Sum بار اولی اینم که در ری می بار اولی می بود
کدام Sum (ست یعنی) این int می باشد (دکتر) در

سوالم در تابع با اسم یکسان و پارامترهای یکسان داشته باشیم و اینها را می توانیم
در C++ error می دهیم یعنی بار اولی خود اینها نیست!

آرایه ها

تعداد حافظه های بیت که به اندازه 4 type یکسان دارند
 لازم آید آرایه ایست که خانه های آن قبلی بیت هم باشند



اندرین اول در آرایه در است

و [تعداد] اسم نوع

int a[3];

هر خانه int در حافظه
 حافظ آن int است



[اندرین] اسم آرایه

a[0] = 5;

محل آرایه در حافظه

برای دسترسی

برای تعیین آرایه حافظه ی آنیم از دو روش استفاده کنیم

{ int a[3]; or int a[3], b[3];
 int b[3]; }

برای آرایه های 10 تا 15 آرایه در حافظه

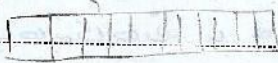
int main()

حافظه ی آرایه

{

int a[10];

حافظه ی آرایه



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

در C، compiler آرایه را اول نمی کند در این مثال

a[i] = 5;

مثلا برای آرایه ای [20] cout << a[20] از 20 حافظه ی آرایه

در هر سوال حافظه است چک کنند

Subject:

Year: Month: Date: ()

```
int a[10] = {0};
```

یک آرایه از 10 خانه با مقدار 0

در حافظه قرار می‌گیرد

```
int a[4] = {5, 1, 2, 7};
```

```
a[0] = 5    a[1] = 1    a[2] = 2    a[3] = 7
```

```
int a[4] = {5, 1};
```

```
a[0] = 5    a[1] = 1    a[2] = 0    a[3] = 0
```

```
int a[4] = {5, 1, 2, 7, 10};
```

این خط را در کامپایلر (compiler) خطا می‌دهد زیرا 5 عدد در براکت تعریف شده است.

در خط اول تعریف آرایه و در خط دوم مقداردهی آن انجام می‌دهیم. اگر در خط اول فقط آرایه تعریف شود و در خط دوم مقداردهی انجام ندهیم، کامپایلر خطا می‌دهد.

```
# Define size 10
```

constant

در خط اول آرایه را تعریف می‌کنیم و در خط دوم آن را مقداردهی می‌کنیم.

```
a[size]
```

آرایه را با نام آرایه تعریف می‌کنیم

همچنین می‌توانیم از آرایه استفاده کنیم

```
const int size = 10;
```

مقدار این ثابت در طول برنامه تغییر نمی‌کند

```
type    define    const
```

در define حتماً نوع آرایه را می‌نویسیم

و در type نوع آرایه را می‌نویسیم

مقدار آرایه را می‌نویسیم

Subject:

Year. Month. Date. ()

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

مقدور تغییر const تعیین می کند مقدار را ثابت نگه دارد

در 5 = عدد compiler قطعی می بیند

```
int a[] = {5, 7, 8};
```

```
{
a[3]
```

تعداد این اعداد مقدار را تعیین می کند و در حافظه 3 در نظر می آید

```
int main ()
```

```
{
```

```
int a[] = {3, 1, 4, 2, 7};
```

```
int total = 0;
```

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

```
total += a[i];
```

```
cout << "total = " << total << endl;
```

```
return 0;
```

```
}
```


Subject: _____

Year. _____ Month. _____ Date. _____ ()

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

```
int main ( )
```

```
{
```

```
int numbers[40] = {1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10, 10};
```

```
int f [10] = {0};
```

```
for (int i=0; i<40; i++)
```

```
    f[numbers[i]-1]++;
```

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

```
    cout << "f[" << i << "] = " << f[i] << endl;
```

```
return 0;
```

```
}
```

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

```
int main ( )
```

```
{
```

```
int num [ ] = {5, 2, 3, 4, 7, 8, 1, 2, 3, 2};
```

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

```
    { for (int j=0; j<num[i]; j++)
```

```
        cout << "#";
```

```
        cout << endl;
```

```
    }
```

```
return 0;
```

```
}
```

P4PCO

Subject:

Year. Month. Date. ()

۳-ی توان مستقیم اندکی خوانده شود

۴- مستقیماً در خروجی چاپ می شود

در `int` مستقیماً ترانسپایلر `(atoi)` را از دستوری `atoi` می خواند

~~`int a[10];`~~

~~`cin >> a;`~~

`char a[10];`

`cin >> a;`

عنوان دارد می شود `Enter` زده می شود از دستوری

خوانده می شود در `9` و `10` می شود کار می کند

حرف وارد می شود `9` و `10` می شود

انواع خاصی که `space` و `tab` است فقط `Ctrl` را می خواند

Ali Reza

مادر Ali می بیند

موتوری که `cin` می کنیم از `Keyboard` است از `Keyboard` خوانده می شود

نویسند `overwrite` می شود به جای آن `Ctrl` می شود

`cin >> setw(10) >> a;`

که `Ctrl` و `Tab` می خواند

`setw [n]` `n` حرف می خواند از دستوری خوانده می شود

اگر `1000` تا حرف وارد کرده باشد `Enter` نیز می خواند `Ctrl` آن خوانده می شود

خفیه ای `ignore` می شود

Subject:

Year. / Month. Date. ()

ایمانی چند لایه ای

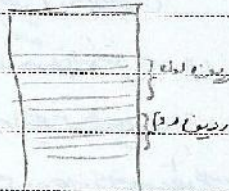
int a[10];



int a[3][4];



RAM



از این سفر

a[1][2] = 5

↓
اینجا مستقر

برای مقدار دهی به این آرایه از این for استفاده کنیم

```
for (int i = 0; i < 3; i++)
```

```
for (int j = 0; j < 4; j++)
```

```
a[i][j] = 0;
```

int a[3][4] = { { 1, 2, 7, 0 }, { 3, 4, 5, 6 }, { 0, 0, 0, 0 } }

اینجا به این آرایه مقدار دهی می‌کنیم

در اینجا به این آرایه مقدار دهی می‌کنیم

	0	1	2	3
0	2	7	0	0
1	3	4	5	6
2	0	0	0	0

Subject:

Year: Month: Date: ()

int f (int a[], int size) ←

↑
array size array element

تایم اریب اریب در یک کند (ریسک)

int a[3][4]; ←

این تایم اریب است

int f (int x[][4])

همه اریب باید در یک تایم اریب

باید در یک تایم اریب

{

کند تایم اریب است

x[2][3] = 5;

تایم اریب این اریب است

تایم اریب است

x[i][j] → x + i * functional + j

include <iostream.h>

void printArray (int l[][3]);

int main ()

{ int a1[2][3] = { {1, 2, 3}, {4, 5, 6} }

1	2	3
4	5	6

a1

a2[2][3] = { {1, 2, 3}, {4, 5, 6} }

1	2	3
4	5	6

a2

a3[2][3] = { {1, 2, 3}, {4, 5, 6} }

1	2	3
4	5	6

a3

cout << "array 1: " << endl;

printArray (a1);

cout << "array 2: " << endl;

printArray (a2);

cout << "array 3: " << endl;

printArray (a3);

return 0;

}