

چند مثال

توابع

مبانی برنامه‌نویسی

(۱۳۹۱-۱۳۹۰-۱۳۸۹)

جلسه‌ی هجدهم



دانشگاه شهید بهشتی

پاییز ۱۳۹۲

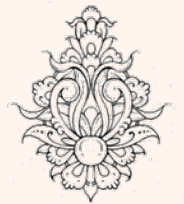
دانشکده‌ی مهندسی برق و کامپیوتر

احمد محمودی ازناوه

فهرست مطالب

- چند مثال

- تابع



توقف در حلقه‌های تودرتو

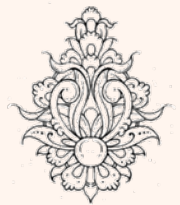
break Statement with Nested Loops

```
#include <iomanip> // defines setw()
#include <iostream> // defines cout
using namespace std;
int main()
{
for (int x=1; x <= 12; x++)
{
for (int y=1; y <= 12; y++)
if (y > x) break;
else cout << setw(4) << x*y;
cout << endl;
}
}
```

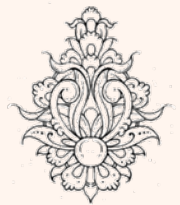
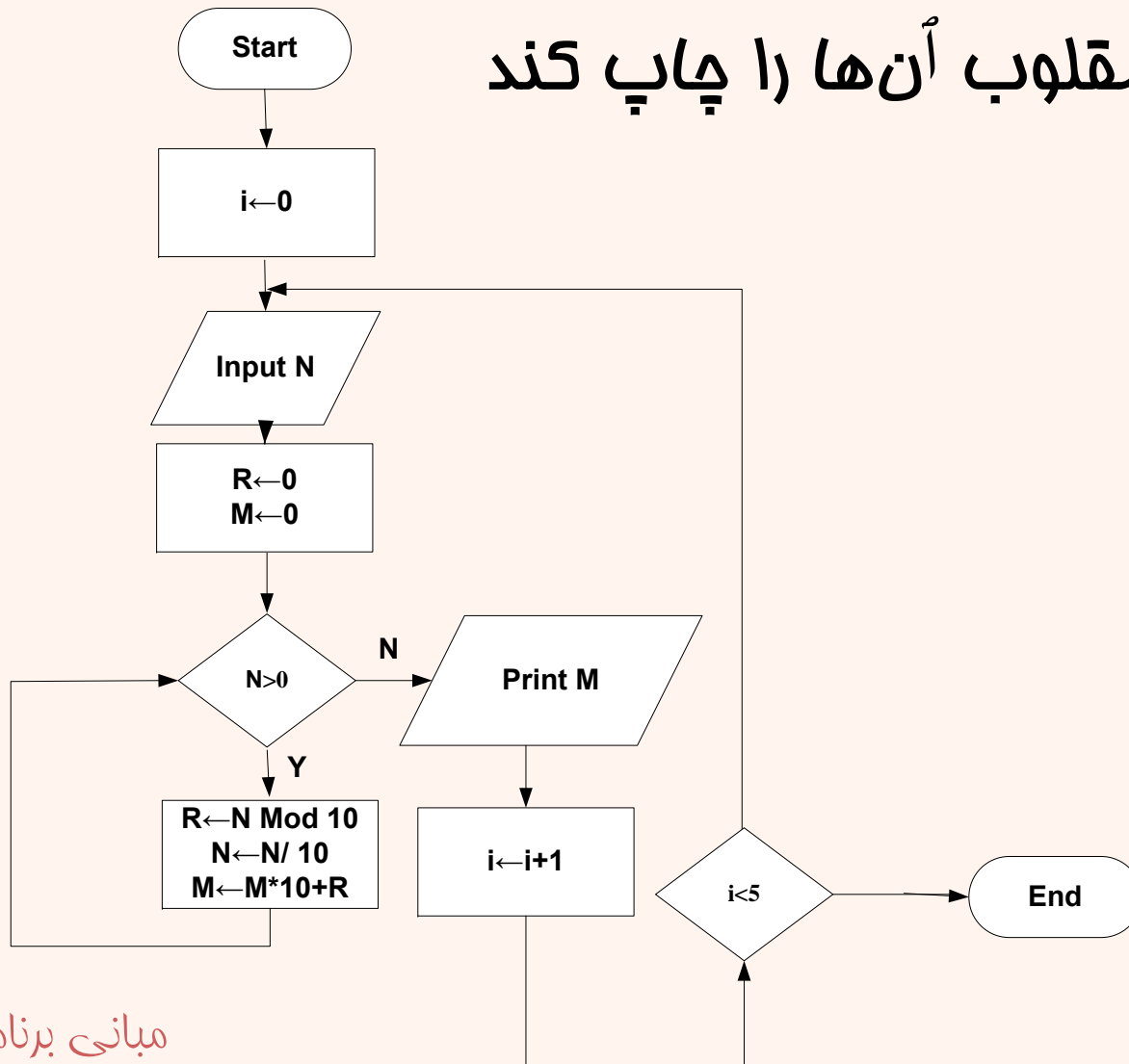
• خروجی برنامه زیر چیست؟

هنگامی که از **break** در حلقه‌های تودرتو استفاده می‌شود، خروجی از داخلی‌ترین حلقه صورت می‌گیرد.

```
1
2 4
3 6 9
4 8 12 16
5 10 15 20 25
6 12 18 24 30 36
7 14 21 28 35 42 49
8 16 24 32 40 48 56 64
9 18 27 36 45 54 63 72 81
10 20 30 40 50 60 70 80 90 100
11 22 33 44 55 66 77 88 99 110 121
12 24 36 48 60 72 84 96 108 120 132 144
```



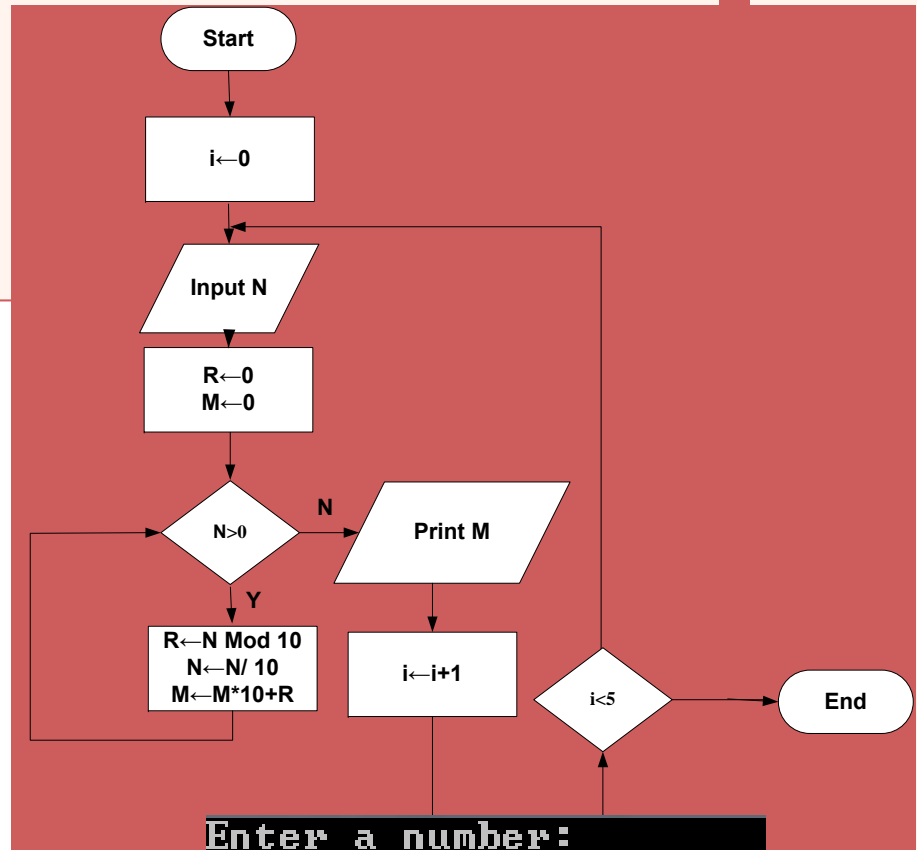
- روندنما و برنامه‌ای بنویسید که ۵ عدد را از ورودی گرفته مقلوب آن‌ها را چاپ کند



```

int main()
{
int i,r,m,n;
i=0;
for(i=0;i<5;i++)
{
m=0;r=0;
cout<<"Enter a number:\n";
cin>>n;
for(;n>0;){
r=n%10;
n=n/10;
m=m*10+r;
}
cout<< "The result is:" << m <<endl;
}
}

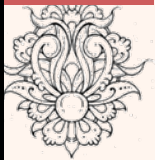
```



```

Enter a number:
167
The result is:761
Enter a number:
86
The result is:68
Enter a number:
555732
The result is:237555
Enter a number:
69700
The result is:796
Enter a number:
230128
The result is:821032

```



دانشگاه
سپهبد
بهشتی

تمرین

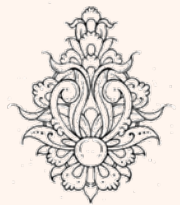
هدف، استفاده از یک حلقه برای انجام محاسبات است. برنامه به گونهای نوشته شود که حاصل جمع به ترتیب محاسبه شود

- برنامه ای بنویسید که سری زیر را تا n محاسبه و چاپ کند (n عددی فرد است):

$$sum = 1 - 3 + 5 - 7 + 9 \dots n$$

```
Enter a number:9
The result is:5
```

```
#include <iostream>
using namespace std;
int main()
{
    int sum=0,i,f=1,n;
    cout<<"\nEnter a number:";
    cin>> n;
    for(i=1;i<=n;i=i+2)
    {
        sum = sum+ f*i;
        f=-f;
    }
    cout<<"The result is:" <<sum <<endl;
}
```



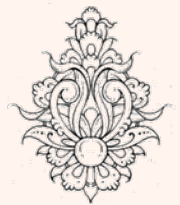
تمرین

- برنامه ای بنویسید که n را خوانده و سری زیر را تا n محاسبه و چاپ کند (بدون استفاده از تابع توان پیاده سازی شود).

```
int main()
{
    long int sum=0,i,j,p,n;
    cout<<"\nEnter a number:";
    cin>> n;
    for(i=1;i<=n;i++)
    {
        p=1;
        for(j=1;j<=i;j++)
            p=p*j;
        sum = sum + p;
    }
    cout<<"The result is:" <<sum <<endl;
}
```

$$\text{sum} = 1^1 + 2^2 + 3^3 + \dots + n^n$$

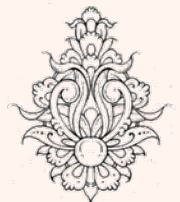
```
Enter a number:4
The result is:288
```



- برنامه‌ای بنویسید که تعداد «۱»های یک عدد صحیح در نمایش دودویی را بشمارد.

```
#include <iostream>
using namespace std;
int main(){
    int InpNum;
    unsigned char NumOfOnes=0;
    cout<<"please enter a number"<<endl;
    cin>>InpNum;
    for(int i=0; i<32; i++,InpNum>>=1)
        NumOfOnes+=InpNum&1;
    cout<<"The number of '1' is \n";
    cout<<(int)NumOfOnes<<endl;
}
```

```
please enter a number
-1
The number of '1' is
32
```



برنامه‌های بنویسید که عددی صحیح و مثبت از ورودی بخواند و تعیین کند توانی از ۲ است یا نه؟

۱

```
#include <iostream>
using namespace std;
int main()
{
    unsigned int num;
    char flg=0;
    cin>>num;
    for(int i=1; i<32 && (flg<=1); i++,num>>=1)
        flg+=num&1;
    cout<<(flg==1?"It is power of two":"It is NOT power of two")<<endl;
}
```

۲

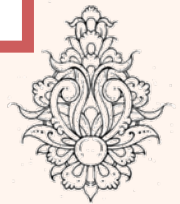


```
#include <iostream>
using namespace std;
int main()
{
    unsigned int num;
    char flg=0;
    cin>>num;
    flg = num && !(num & (num - 1));
    cout<<(flg==1?"It is power of two":"It is NOT power of two")<<endl;
}
```

• تکه برنامه‌ی زیر چه می‌کند؟

```
#include <iostream>
using namespace std;
int main(){
    int num;
    cout<<"please enter a number"<<endl;
    cin>>num;

    num = num - ((num >> 1) & 0x55555555);
    num = (num & 0x33333333) + ((num >> 2) & 0x33333333);
    cout<<(((num + (num >> 4)) & 0x0F0F0F0F) * 0x01010101) >> 24)<<endl;
}
```



پیش از این نیز تا حدی از توابع استفاده کردیم،
برنامه‌های که می‌نوشتیم، تابعی به نام main بود،
همچنین از توابع کتابخانه‌های استفاده کردیم
در این بخش خواهیم آموخت چگونه خودمان توابع
مورد نیاز را بنویسیم

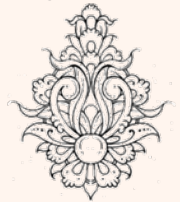
توابع



FUNCTIONS

تعریف تابع

- در بیشتر زبان‌های برنامه‌نویسی، برنامه‌ها به بخش‌های مختلفی تقسیم می‌شوند که به این بخش‌ها «زیربرنامه» گفته می‌شود.
- برای این که برنامه‌های بزرگ قابل مدیریت باشند، برنامه‌نویسان این برنامه‌ها را به زیربرنامه‌هایی بخش‌بندی می‌کنند.
- در زبان C به زیربرنامه «تابع» گفته می‌شود.
- توابع را می‌توان به طور جداگانه کامپایل و آزمایش نمود و در برنامه‌های مختلف بارها از آن‌ها استفاده کرد.



<نوع فروری>

<نام تابع>

<انواع ورودی>

{

.....

}

استفاده از توابع برنامه‌نویسی را ساده‌تر می‌کنند



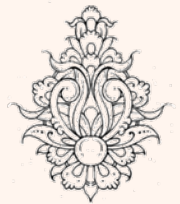
توابع کتابخانه‌ای C++ استاندارد

- «کتابخانه‌ی استاندارد C++» مجموعه‌ای است که شامل توابع از پیش تعریف شده و سایر عناصر برنامه است. این توابع و عناصر از طریق **header** ها قابل دستیابی‌اند.
- مثال: تابع **sqrt** در کتابخانه‌ی **cmath** موجود است.

```
#include <iostream>
#include <cmath>
using namespace std;
int main(){
    double number;

    cout << "Please enter a number: ";
    cin>>number;
    cout << "The sqrt of " << number<<" is: "<<sqrt(number)<<endl;
    return 0;
}
```

```
Please enter a number: 9
The sqrt of 9 is: 3
```

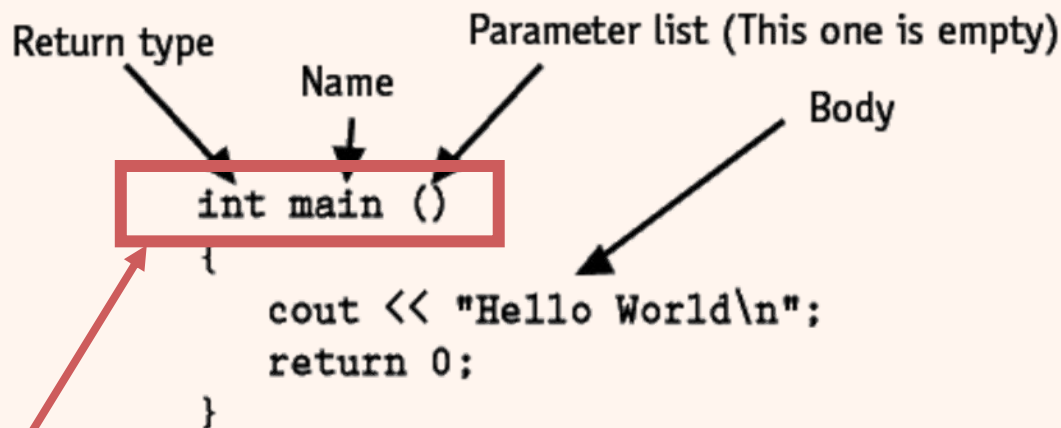


بخش‌های مختلف یک تابع

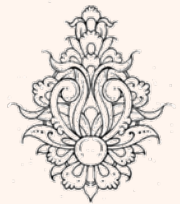
Defining and Calling Functions

• هنگامی که یک تابع را تعریف می‌کنید می‌باید قسمت‌های زیر را در نظر بگیرید:

- نام
- آرگومان‌های ورودی
- بدنه
- نوع بازگشتی



طرز نوشتن به سرآیند تابع (function header) مشهور است



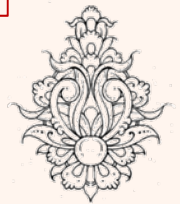
تابع void

Void Functions

- همه‌ی توابع به مقدار بازگشتی نیاز ندارند!
- تابعی که مقدار بازگشتی ندارد را **void** می‌نامیم.

```
void displayMessage()  
{  
    cout << "Hello from the function displayMessage.\n";  
}
```

اسم توابع باید **لوی** باشند،
به گونه‌ای که از نام یک تابع بتوان به عملکرد آن پی برد



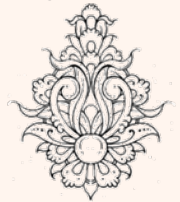
تابع void

```
void displayMessage()  
{  
cout << "Hello from the function displayMessage.\n";  
}  
  
int main()  
{  
cout << "Hello from main.\n";  
displayMessage();  
cout << "Back in function main again.\n";  
return 0;  
}
```

```
Hello from main.  
Hello from the function displayMessage.  
Back in function main again.
```

```
void displayMessage()  
{  
    cout << "Hello from the function displayMessage.\n";  
}
```

```
int main()  
{  
    cout << "Hello from main.\n";  
    displayMessage();  
    cout << "Back in function main again.\n";  
    return 0;  
}
```




```

int cube(int x)
{ // returns cube of x:
return x*x*x;
}

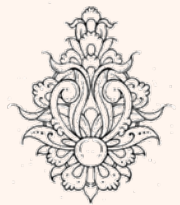
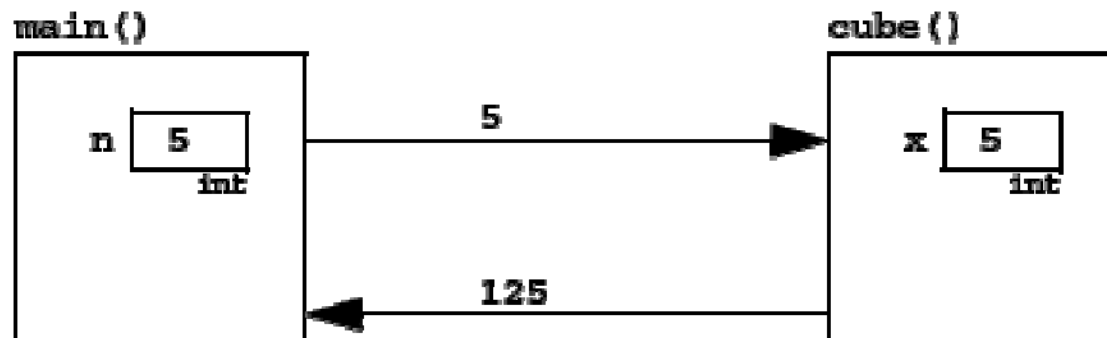
int main()
{ // tests the cube() function:
int n=1;
while (n != 0)
{
cout<<"Enter a number (0 for quit):";
cin >> n;
cout << "\tcube(" << n << ") = " << cube(n) << endl;
}
}

```

```

Enter a number (0 for quit):4
      cube(4) = 64
Enter a number (0 for quit):6
      cube(6) = 216
Enter a number (0 for quit):9
      cube(9) = 729
Enter a number (0 for quit):0
      cube(0) = 0

```

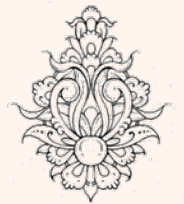


```
// function example
#include <iostream>
using namespace std;

int addition (int a, int b)
{
    int r;
    r=a+b;
    return (r);
}

int main ()
{
    int z;
    z = addition (5,3);
    cout << "The result is " << z;
    return 0;
}
```

The result is 8



چگونگی کارکرد

- ابتدا تابع main اجرا می‌شود.
- دو مقدار ۵ و ۳ در متغیرهای a و b کپی شده و به تابع ارسال می‌گردند.
- مقدار مورد نظر در تابع محاسبه شده و مقدار ۸ باز می‌گردد.
- متغیرهای a و b تنها در محدوده‌ی تابع معتبر هستند.

```
int addition (int a, int b)
z = addition ( 5 , 3 );
```

