درس «مبانی کامپیوتر و برنامه‌سازی»

حلقه

صادق علی‌اکبری

سرفصل مطالب

- حلقه‌ها
  - حلقه while
  - حلقه do while
  - حلقه for
- حلقه‌های تودرتو
- دستورات break و continue

حلقه‌ها (Loops)

حلقه

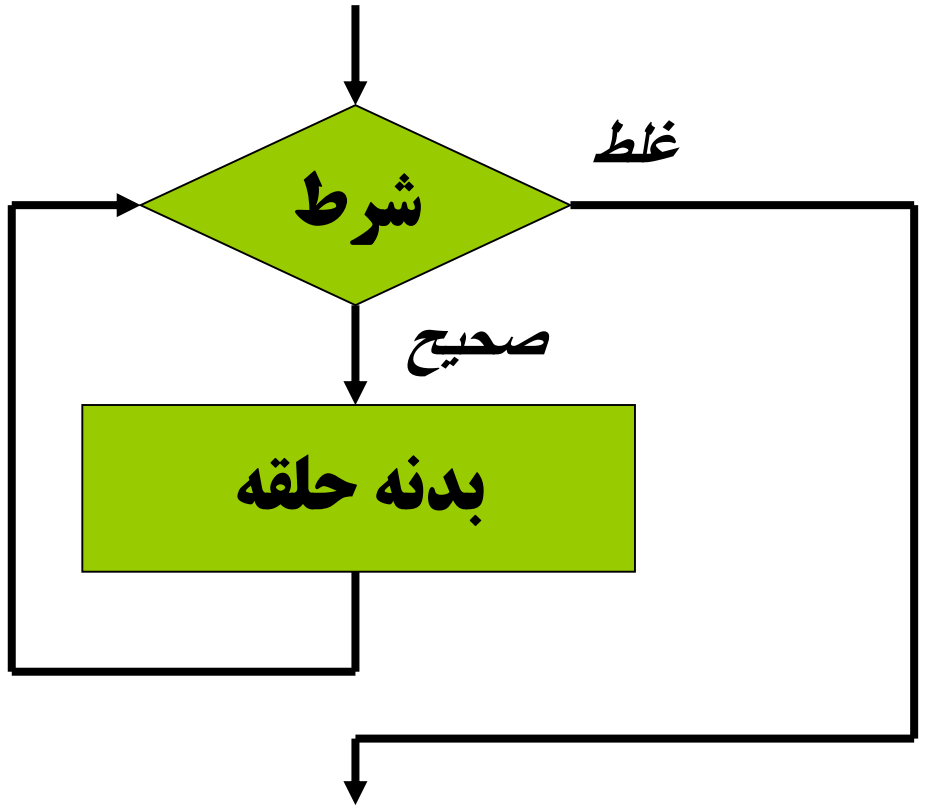
- امکانی برای انجام چندباره یک عملیات
- تعداد مشخصی از تکرار، یا تکرار تا زمان حصول یک شرط
- مثال:
  - نمره ۱۰۰ دانشجو از ورودی خوانده شود
  - تا زمانی که کاربر مقدار ۱- را وارد نکرده، مقدار نمرات از کاربر گرفته شود
- سه گونه حلقه در زبان C:
  - for
  - while
  - do-while

حلقه while

- تا زمانی که یک شرط برقرار باشد، بدنه حلقه while اجرا می‌شود

```
while (boolean expression)
        statement to repeat;
```

```
while (boolean expression)
{
        statement1;
        statement2;
        statement3;
}
```



مثال برای حلقه while

مواظب باشید:
بعد از دستور while از
سمیکالن استفاده نکنید

```
int counter=1;
while(counter<10)
{
    cout << counter ;
    counter++;
}
```

بلوک

- خروجی این برنامه؟
- چاپ اعداد ۱ تا ۹

تمرین

- برنامه‌ای بنویسید که عدد صحیح و مثبت n را از ورودی بگیرد،
  و اعداد n تا ۱ را به صورت نزولی چاپ کند

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

cout << "Enter a number:";
unsigned int n;
cin >> n;
while(n>0)
{
    cout<<n <<"\n";
    n--;
}

return 0;
}
```

تمرین

- برنامه‌ای بنویسید که عدد صحیح و مثبت n را از ورودی بگیرد،

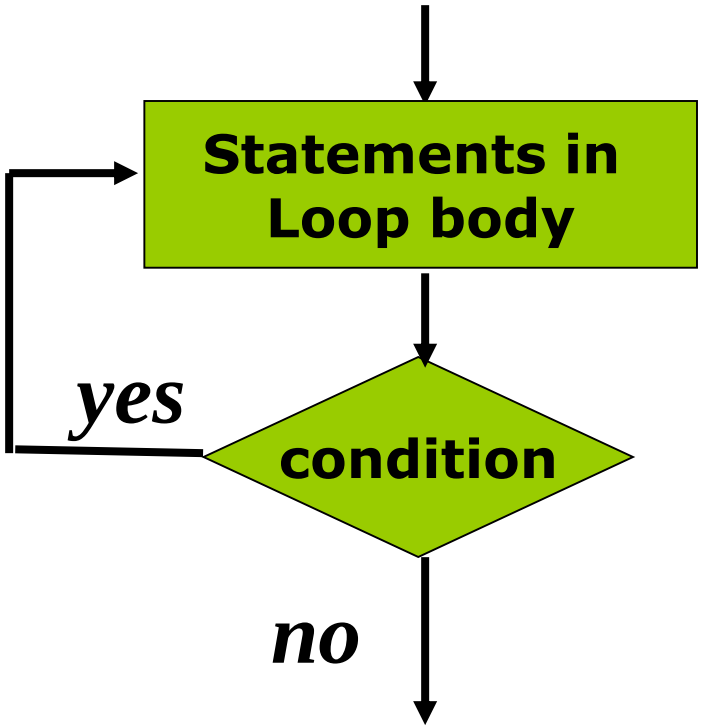
و مجموع اعداد ۱ تا n را چاپ کند

```
cout << "Enter a number:";
unsigned int n;
int sum = 0;
cin >> n;
while(n>0)
{
  sum += n;
  n--;
}
cout<<sum ;
```

```
cout << "Enter a number:";
unsigned int n;
int sum = 0;
cin >> n;
while(n>0)
  sum += n--;
cout<<sum ;
```

حلقه do-while

```
do
    statement to repeat;
while (boolean expression)
```



- شرط حلقه، در انتهای عملیات بررسی می‌شود

```
do
{
        statement1;
        statement2;
        …
} while (boolean expression)
```

- نکته: بدنه حلقه while ممکن است هرگز اجرا نشود
  - اگر شرط while از ابتدا برقرار نباشد
- اما بدنه do-while حداقل یک بار اجرا می‌شود

مثال برای حلقه do-while

```
int counter=0;
do{
     counter++;
     cout << counter ;
}while(counter<10);
```

- خروجی این برنامه؟
- چاپ اعداد ۱ تا ۱۰

مثال

- قطعه‌برنامه‌ای بنویسید که مدام یک عدد از کاربر بگیرد و مجذور آن را چاپ کند
- این کار را تا زمانی ادامه دهد که عدد ورودی صفر باشد
- (شرط خاتمه : ورودی صفر باشد)

```
int n;
do
{
   cin >>n;
   cout<<n*n<<endl;
}
while(n!=0);
```

مثال

- برنامه‌ای که از صفر تا یک را یک‌دهم یک‌دهم اضافه کند و چاپ نماید
- 0 0.1 0.2 0.3 ... 0.9 1.0

```
double n = 0;
do{
    cout<<n<<endl;
    n+=0.1;
}
while(n!=1);
```

- اشکال برنامه مقابل چیست؟
  - اعداد اعشاری به صورت تقریبی ذخیره می‌شوند
  - مقایسه مستقیم تساوی اعداد اعشاری صحیح نیست
  - راه حل:

```
double n = 0;
do{
 cout<<n<<endl;
 n+=0.1;
}
while(n<1.05);
```

مثال: فیبوناچی

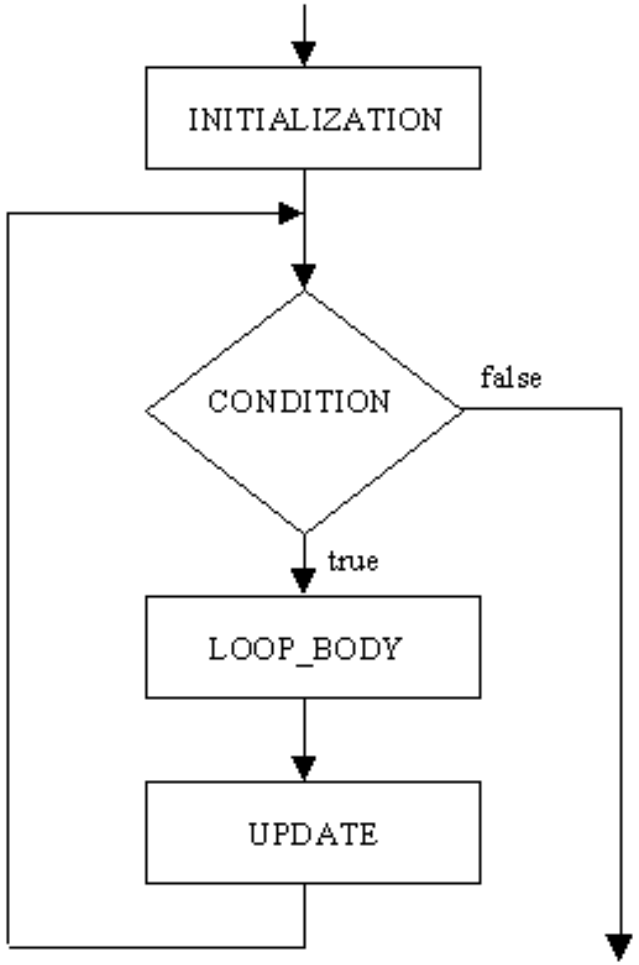
- قطعه‌برنامه‌ای بنویسید که دنباله فیبوناچی را برای اعداد کوچکتر از ۱۰۰ چاپ کند

```
int a = 0, b = 0, c = 1;
do {
   cout << c << endl;
   a = b;
   b = c;
   c = a + b;
} while (c < 100);
```

```
1
1
2
3
5
8
13
21
34
55
89
```

حلقه for

for (یک عملیات ; یک شرط ; یک عملیات) یک دستور



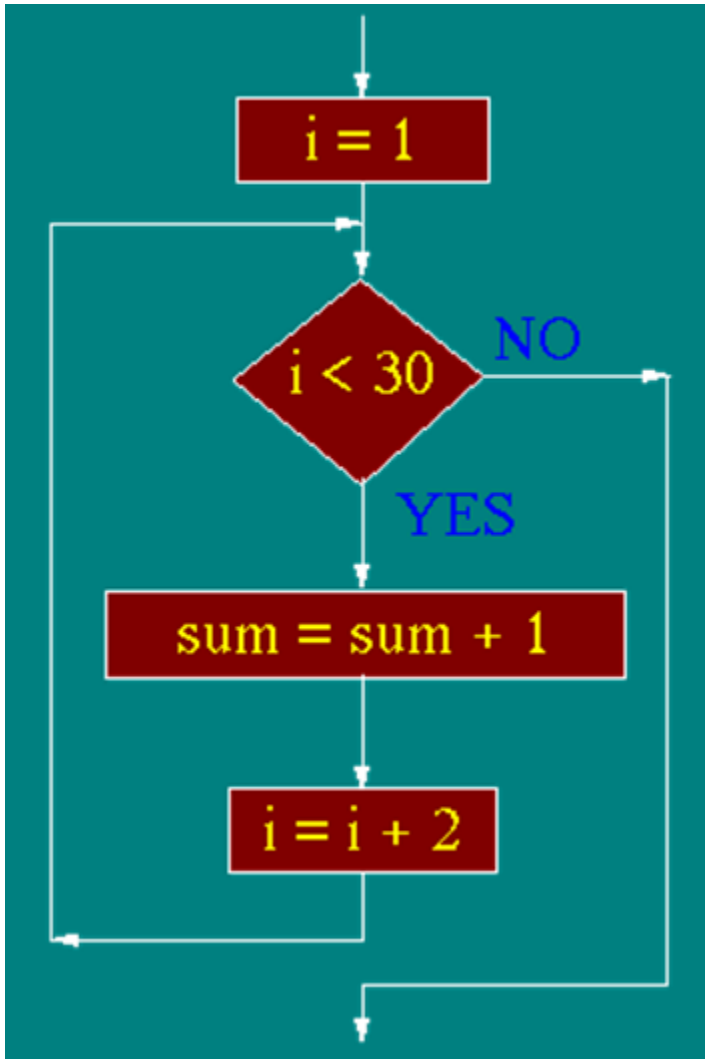
- پراستفاده‌ترین حلقه

```
for (INITIALIZATION; CONDITION; UPDATE)
    LOOP_BODY
```

مثال

```
int i;
int sum = 0;
for(i=1;i<30;i+=2)
   sum++;
```

نتیجه: sum=15



مثال برای حلقه for

- مثال:

```
for (int i = 1; i <= 10; i++) {
  cout << i ;
}
```

- خروجی این برنامه؟
- چاپ اعداد ۱ تا ۱۰

بازنویسی for با کمک while

```
for (X; Y; Z) {
  body();
}
```

این‌ها شبه‌کد هستند

```
X;
while(Y){
  body();
  Z;
}
```

مثال:

```
for (int i = 1; i <= 10; i++) {
  cout << i ;
}
```

```
int i=1;
while(i<=10){
      cout << i ;
      i++;
}
```

مثال: دنباله فیبوناچی را با کمک حلقه for چاپ کنید

```
1 1 2 3 5 8 13 21 34 55 89
```

```
for (int a = 0, b = 0, c = 1; c < 100; a = b, b = c, c = a + b)
    cout << c <<endl;
```

```
for (int a = 0, b = 1; b < 100; b+=a, a=b-a)
  cout << b <<endl;
```

- نکته: در هر بخش از حلقه for می‌توانیم از چند دستور استفاده کنیم
  - دستورات مختلف را به کاما جدا کنید

کدام حلقه بهتر است؟

- از بین سه حلقه while ، do-wile و for کدام بهتر است؟
- هیچ یک بهتر نیست
- هر کدام کاربرد و شرایط استفاده خود را دارند
- پیش‌آزمونی و پس‌آزمونی (این که شرط حلقه را ابتدا یا انتها چک می‌کند)
  - حلقه while : پیش‌آزمونی
  - حلقه do while : پس‌آزمونی
  - حلقه for : پیش‌آزمونی

حلقه‌های تودرتو (Nested Loops)

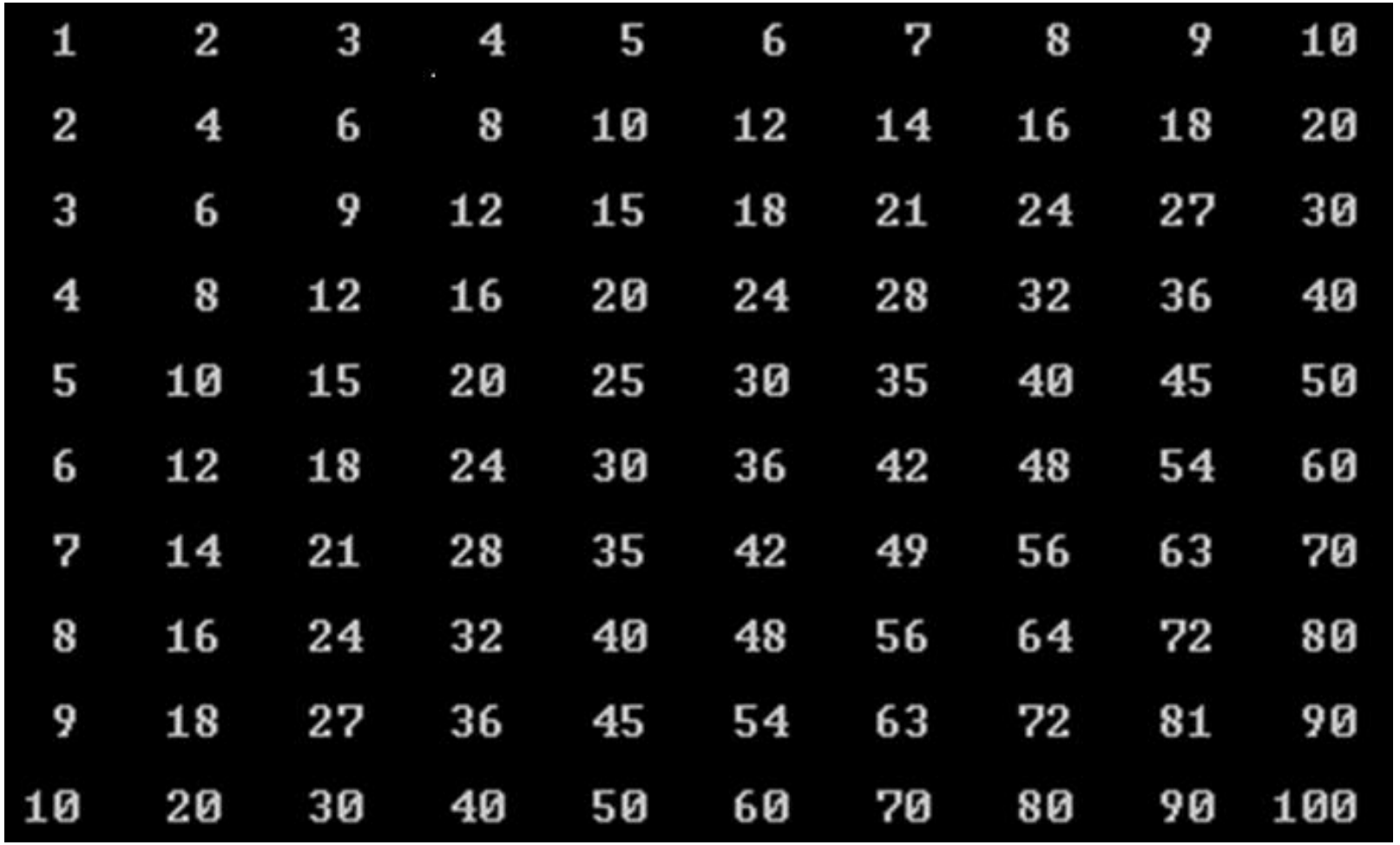
حلقه‌های تودرتو (Nested Loop)

```
for(int i=0 ;i <10; i++)
{
   ...
   while(a>b)
   {
      ...
   }
   ...
}
```

- حلقه‌ای داخل حلقه دیگر
- مثال: یک while داخل یک for
- حلقه بیرونی (outer loop)
- حلقه درونی (inner loop)

تمرین

- برنامه‌ای بنویسید که جدول ضرب زیر را چاپ کند



1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

چاپ جدول ضرب

```
for(int i = 1 ; i<=10; i++)
{
  for(int j = 1 ; j<=10; j++)
     printf("%5d", i*j);

  printf("\n");
}
```

حلقه بیرونی

حلقه درونی

- در برنامه فوق،

چرا حلقه بیرونی دارای بلوک (با آکولاد) و حلقه درونی فاقد بلوک است؟

مثال

- برنامه‌ای بنویسید که مادامی که کاربر عدد منفی وارد نکرده، از کاربر یک عدد بگیرد و هربار مجموع اعداد کوچک‌تر از آن را چاپ کند

```
int number;
int sum;
cin >> number;
while (number >= 0)
{
  sum=0;
  for(int i=0;i<number;i++)
     sum+=i;
  cout << sum << '\n';
  cin >> number;
}
```

نمونه خروجی :

```
0
0
1
0
2
1
3
3
4
6
-1
```

مثال

- برنامه مترجم: تا زمانی که کاربر صفر را وارد نکرده، یک منو از کلمات را نشان دهد و انتخاب کاربر را ترجمه کند.

```
int select;
do{
  cout << "Please select: \n";
  cout << "1)Hi 2)Bye 0)EXIT \n";

  cin>>select;
  switch(select){
  case 1: cout<<"Salam\n"; break;
  case 2: cout<<"Khodahafez\n"; break;
  case 0: break;
  default: cout<<"Chi?\n";
  }
}while(select!=0);
```

```
Please select:
1)Hi 2)Bye 0)EXIT
1
Salam
Please select:
1)Hi 2)Bye 0)EXIT
1
Salam
Please select:
1)Hi 2)Bye 0)EXIT
2
Khodahafez
Please select:
1)Hi 2)Bye 0)EXIT
3
Chi?
Please select:
1)Hi 2)Bye 0)EXIT
0
```

دانستنی: تولید عدد تصادفی

- در بسیاری از برنامه‌ها نیاز به اعداد تصادفی (random) داریم
- دستور rand() یک عدد تصادفی تولید می‌کند
- مثال: برنامه زیر صد عدد تصادفی یک رقمی چاپ می‌کند

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main() {
  srand(time(NULL));
  for (int i = 0; i < 100; i++) {
    int random = rand() % 10 ;
    printf("%d ", random);
  }
  return 0;
}
```

تمرین: بازی حدس اعداد

- کامپیوتر یک عدد بین ۱۰ تا ۲۰ انتخاب کند (10<=n<=20)
- کاربر یک حدس بزند
- کامپیوتر راهنمایی کند: کوچکتر یا بزرگتر
- تا زمانی که کاربر موفق شود
- وقتی موفق شد، کامپیوتر بپرسد: باز هم می‌خواهی بازی کنی؟

برنامه حدس اعداد

```
#include <stdlib.h>
#include <time.h>
#include <iostream>
using namespace std;
int main() {
    srand(time(NULL));
    int select;
    do {
        int number;
        int random = (rand() % 11) + 10;
        do {
            cin >> number;
            if (number > random)
                cout << "Koochiktar Entekhab Kon!\n";
            else if (number < random)
                cout << "Bozorgtar Entekhab Kon! \n";
        } while (number != random);
            cout << "Doroste!\n 1)Edame 2)Payan ";
        cin >> select;
    } while (select == 1);
    return 0;
}
```

سایر دستورات کنترل جریان برنامه

دستور break

- اجرای حلقه قطع می‌شود
  - این دستور، حلقه را متوقف می‌کند
- ادامه برنامه (از بعد از حلقه) اجرا می‌شود

```
while (true) {
   int number;
   cin >> number;
   if (number < 0)
      break;
   cout <<number*number<<endl;
}
```

```
2
4
5
25
-1
```

```
for (int i = 1; i < 10; i++)
{
   cout << i << '\n';
   if (i == 4)
      break;
}
```

```
1
2
3
4
```

مثال

- break در یک حلقه تودرتو

```
#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;
   }
}
```

```
   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
```

دستور continue

- اجرای ادامه بدنه‌ی حلقه را متوقف می‌کند و به ابتدای بدنه حلقه برمی‌گردد
- همچنین شرط حلقه را نیز بررسی می‌کند (اگر شرط برقرار نبود، حلقه را ترک می‌کند)
- مثال:

```
int number;
do {
  cin >> number;
  if (number < 0)
    continue;
  cout<<number*number<<endl;
} while (number != 0);
```

تفاوت continue در دستور for :

بخش update هم اجرا می‌شود

مثلاً i++ در قطعه‌برنامه زیر:

```
for(int i=0;i<10;i++){
  if(i%4==0)
    continue;
  cout<<i;
}
```

حلقه بینهایت

Infinite loop
Endless loop

- حلقه‌ای که هرگز تمام نشود
- (معمولاً) یک خطای منطقی است
- باعث خطای کامپایل یا خطا در زمان اجرا نمی‌شود

```
for(int i=0;;i++)
  cout<<i<<endl;
```

```
int i=0;
while(true)
  cout<<i++<<endl;
```

```
int i=0;
while (true) {
  cout << i++ << endl;
  if(i>10)
    break;
}
```

حلقه بینهایت نیست

دستور exit

- کل برنامه تمام می‌شود
- مانند return در تابع main عمل می‌کند
- تفاوت exit با return در تابع main چیست؟
  - اگر در تابع main به کار رود، عملاً تفاوتی ندارد
  - ولی ممکن است در هر جای دیگر برنامه به کار رود
- مثال:

```
if(i<0)
  exit(0);
```

دستور goto

- با این دستور، اجرای برنامه به خط دیگری منتقل می‌شود
  - پرش به بخش دیگری از برنامه
- مثال:

```
int i = 0;
LABEL:
cout<<i<<endl;
i++;
if(i<10)
  goto LABEL;
```

```
0
1
2
3
4
5
6
7
8
9
```

نکته درباره break و continue

- اگر این دو دستور در یک حلقه داخلی فراخوانی شوند، بر روی حلقه داخلی اعمال می‌شوند
- دستورات break و continue بر روی داخلی‌ترین حلقه صورت می‌گیرند
- اگر بخواهیم از یک حلقه خارجی break یا continue کنیم چه؟
- راه اول: استفاده از break و شرط (flag)
- راه دوم: استفاده از goto
- نکته هر دو راه حل معایبی دارند
  - زبان‌های جدیدی مثل جاوا راه‌های بهتری برای این مسأله فراهم کرده‌اند

```
for (int i = 0; i < 4; i++) {
  bool breakOuter = false, continueOuter=false;
  for (int j = 0; j < 4; j++)
  {
    if (i ==3 ) {
       breakOuter = true;
       break;
    }else if (i == j) {
       continueOuter = true;
       break;
    }else
       cout<<"i="<<i<<",j="<<j<<endl;
  }
  if(continueOuter)
    continue;
  else if(breakOuter)
    break;
  cout<<"Round "<<i<<endl;
}
```

```
i=1,j=0
i=2,j=0
i=2,j=1
```

break و continue از حلقه بیرونی

راه اول: استفاده از break و شرط (flag)

```
for (int i = 0; i < 4; i++) {
  outer_loop:
  for (int j = 0; j < 4; j++) {
    if (i == 3) {
      goto outside;
    } else if (i == j) {
      i++;
      goto outer_loop;
    } else
    cout << "i=" << i << ",j=" << j << endl;
  }
  cout << "Round " << i << endl;
}
outside:
cout << "Finish" << endl;
```

```
i=1,j=0
i=2,j=0
i=2,j=1
Finish
```

break و continue از حلقه بیرونی

راه دوم: استفاده از goto

معایب goto

Considered harmful

- برنامه‌نویسی ساخت‌یافته (ساختارمند)
- مشکل برنامه اسپاگتی (spaghetti code)
- استفاده از goto برنامه را کثیف و غیرساختارمند می‌کند
  - و فهمیدن و نگهداری برنامه را سخت می‌کند
- نیازی به استفاده از این دستور نیست
  - حلقه‌ها و شرط‌ها مکانیزم بهتری ارائه می‌کنند

جمع‌بندی

جمع‌بندی

- حلقه‌ها: while، do while و for
- حلقه‌های تودرتو
- سایر دستورات کنترل جریان اجرای برنامه
  - break
  - continue
  - exit
  - goto (چیز خوبی نیست، جهنم و سایر ماجراها)

تمرین بیشتر

- برنامه‌ای بنویسید که n را از کاربر بگیرد و مجموع زیر را محاسبه و چاپ کند
- $1^1 + 2^2 + 3^3 + \ldots + n^n$
- برنامه‌ای بنویسید که n را از کاربر بگیرد و مجموع زیر را محاسبه و چاپ کند
- $2 - 4 + 6 - 8 + \ldots n$
- خروجی این قطعه‌برنامه چیست؟

```
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;
```

تمرین بیشتر (ادامه)

- برنامه‌ای که یک عدد صحیح (int) مثبت غیرصفر بگیرد
  و تعیین کند که آیا این عدد توانی از ۲ هست یا خیر
  مثلاً ۸ توانی از دو هست ($۲^۳$) ، ولی ۹ نیست

- دو راه حل:

```
int Input;
int NumberOfOneBits=0;
for(cin >> Input; Input!=0;Input>>=1)
  NumberOfOneBits+=Input&1;
cout<< (NumberOfOneBits==1 ? "Yes" : "No");
```

```
int num;
cin>>num;
cout<< (num & (num-1) ? "No" : "Yes");
```

مطالعه

- C How to Program (Deitel & Deitel), 7th edition (Chapter 3 and 4)

- و یا فصل‌های متناظر در کتاب C++

جستجوی بیشتر

- مسأله سه‌در (Monty Hall Problem) چیست؟
  - جواب صحیح را با کمک شبیه‌سازی (نوشتن یک برنامه کامپیوتری) پیدا کنید.
- دستور srand (که برای تولید اعداد تصادفی استفاده کردیم) دقیقاً چه کاری انجام می‌دهد؟
- عددی که توسط دستور return (در تابع main) و یا exit مشخص می‌شود چیست؟ چه کاربردی دارد؟ در یک کاربرد، از آن استفاده کنید.
- دستور goto در زبان‌های جدید (مثل جاوا و C#) چگونه است؟
  - جاوا ندارد، C# دارد.
- زبان جاوا – که goto ندارد- چه راه‌حلی برای انجام break و continue از حلقه‌های بیرونی فراهم کرده است؟

پایان