**Looping**

---

If we want to perform some task repeat again and again called as loop.
The major purpose of loop is  to optimize the code

**There are two types of loop**

---

**1. Entry control loop:** Entry control loop means first we check condition and after that decide loop will be execute or not called as entry control loop.

**There are two types of entry control loop**
**1. while loop**
**2. for loop**

**2. Exit control loop:** exit control loop means first we execute the loop and after that check condition called as exit control loop.

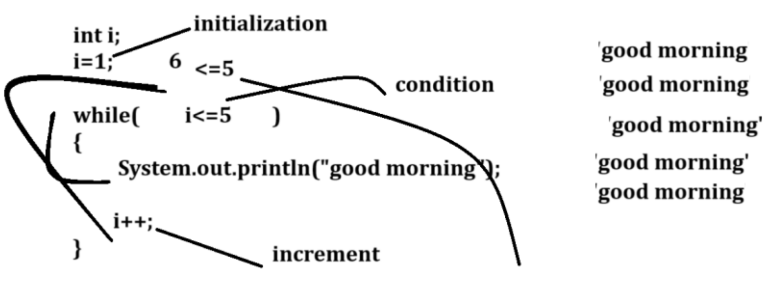
**1. do while loop**

**If we want to work with any loop we should have to know three important points.**
**1. Initialization:** initialization means decide the starting point of loop
**2. Condition:** condition means check how many times loops will be execute or how many times loop will be iterate called as condition.
**3. Increment or decrement:** gap between every iterations**.**

**While loop**

---

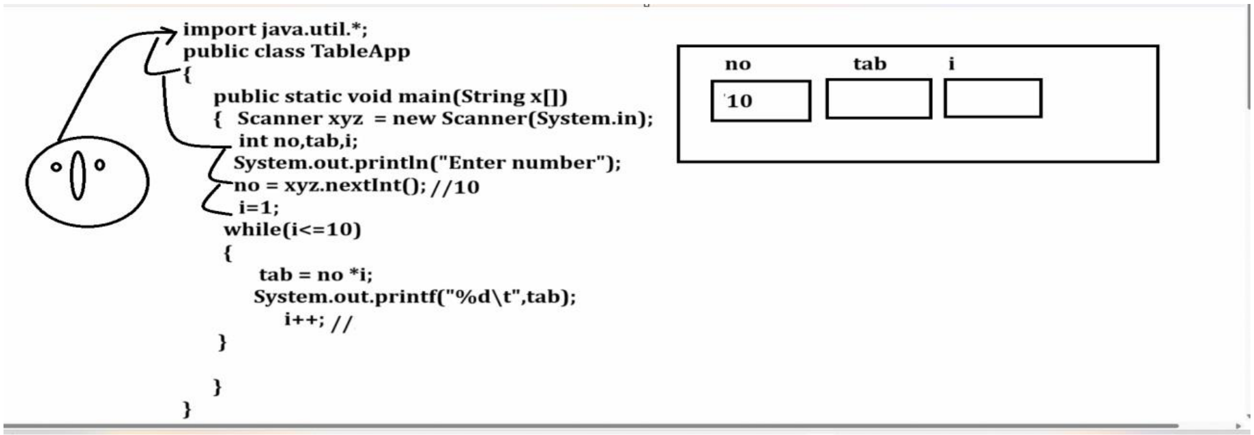
```
initialization;
while(condition)
{   write here your logics
    increment or decrement
}
```



```
int i;
i=1;
while(      i<=5      )
{
   System.out.println("good morning");
   i++;
}
```

```
'good morning
'good morning
 'good morning'
'good morning'
'good morning
```

**Example: WAP input number from keyboard and print its table**
**2**
**2   4   6   8   10   12   14   16   18   20**

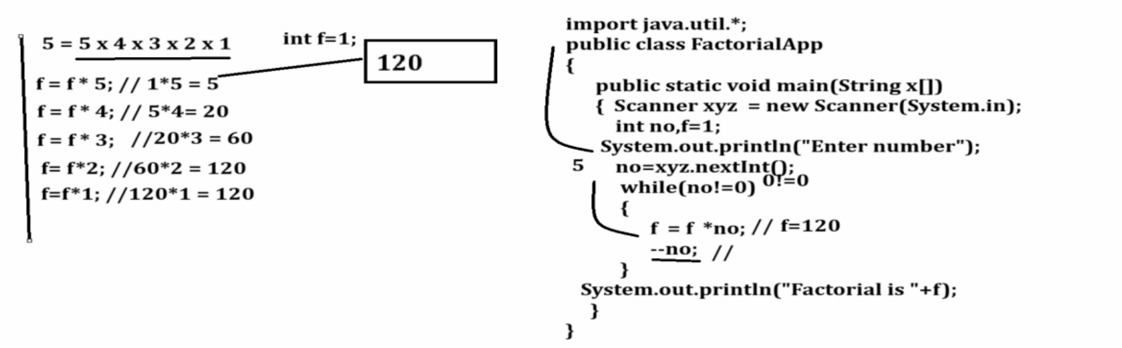


```
import java.util.*;
public class TableApp
{
    public static void main(String x[])
    {  Scanner xyz  = new Scanner(System.in);
       int no,tab,i;
      System.out.println("Enter number");
      no = xyz.nextInt(); //10
       i=1;
     while(i<=10)
     {
          tab = no *i;
         System.out.printf("%d\t",tab);
             i++; //
     }

     }
}
```

**Example: WAP to input number and print its factorial using while loop**

**5**

**Output: 120**



```
5 = 5 x 4 x 3 x 2 x 1        int f=1;
f = f * 5; // 1*5 = 5
f = f * 4; // 5*4= 20
f = f * 3;   //20*3 = 60
 f= f*2; //60*2 = 120
 f=f*1; //120*1 = 120
```

```
import java.util.*;
public class FactorialApp
{
    public static void main(String x[])
    { Scanner xyz  = new Scanner(System.in);
       int no,f=1;
     System.out.println("Enter number");
5      no=xyz.nextInt();
       while(no!=0) 0!=0
       {
           f  = f *no; // f=120
           --no;  //
       }
  System.out.println("Factorial is "+f);
    }
}
```

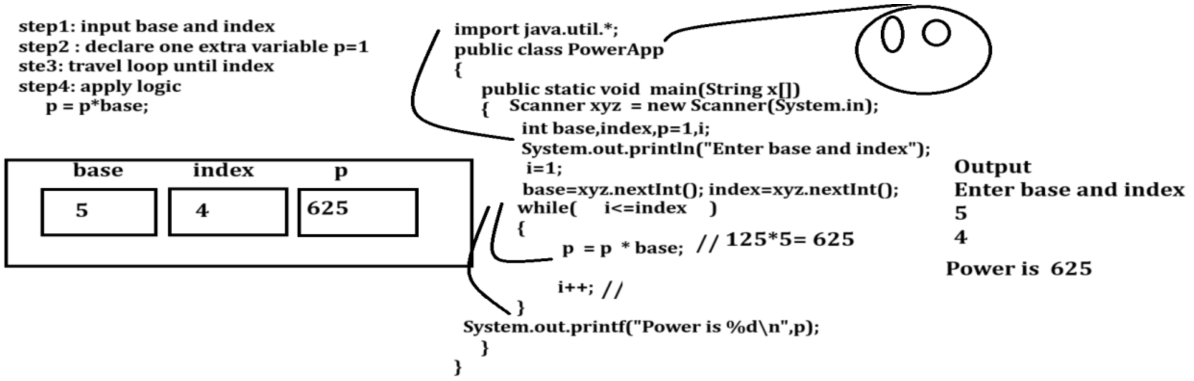
**Example:** WAP to input the two values consider first as base and second as index and calculate its power.

5

4

Output    4

5  = 5 x 5 x 5 x 5  = 625



```
step1: input base and index
step2 : declare one extra variable p=1
ste3: travel loop until index
step4: apply logic
     p = p*base;
```

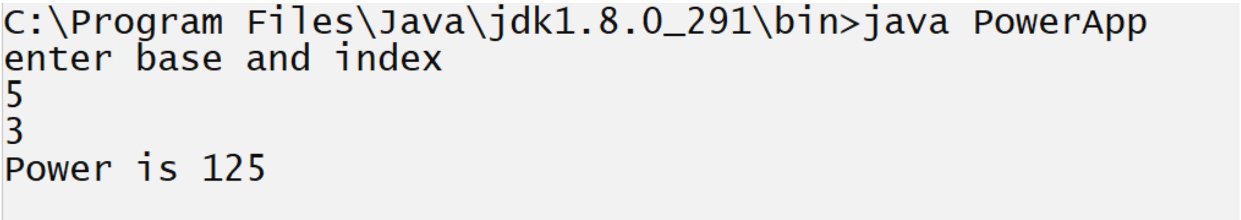
```
import java.util.*;
public class PowerApp
{
    public static void  main(String x[])
    {   Scanner xyz  = new Scanner(System.in);
        int base,index,p=1,i;
        System.out.println("Enter base and index");
         i=1;
        base=xyz.nextInt(); index=xyz.nextInt();
       while(     i<=index    )
       {
             p  = p  * base;  // 125*5= 625

             i++; //
       }
 System.out.printf("Power is %d\n",p);
   }
}
```

```
Output
Enter base and index
5
4
Power is  625
```

**Example with source code**

```
import java.util.*;
public class PowerApp
{
   public static void main(String x[])
           {
                       Scanner xyz  = new Scanner(System.in);
                       int base,index,p=1;
                       System.out.println("enter base and index");
                       base=xyz.nextInt();
                       index=xyz.nextInt();
                       int i=1;
                       while(i<=index)
                       {  p =p  * base;
                         i++;
                       }
                       System.out.printf("Power is %d\n",p);
           }
}
```

**Output**



```
C:\Program Files\Java\jdk1.8.0_291\bin>java PowerApp
enter base and index
5
3
Power is 125
```

Example: WAP to input the number and check number is duck or not

Input:  1024

Output: it is duck number

Input: 1234

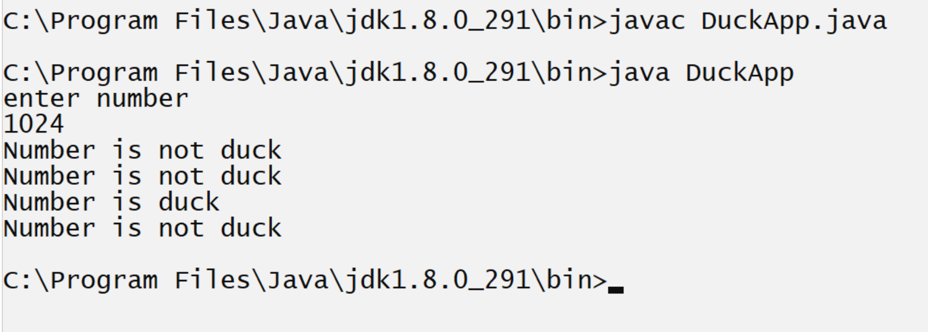
Output: it is not duck.

**Example with source code**

```
import java.util.*;
public class DuckApp
{
   public static void main(String x[])
           {     int no,rem;
                       Scanner xyz  = new Scanner(System.in);
                       System.out.println("enter number");
                       no=xyz.nextInt();
```

```
                  while(no!=0)
                  {
                    rem  = no % 10;
                     no  = no / 10;
                            if(rem ==0)
                            { System.out.println("Number is duck");
                            }
                            else
                            { System.out.println("Number is not duck");
                            }
                  }
         }
}
```

**Output**



```
C:\Program Files\Java\jdk1.8.0_291\bin>javac DuckApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java DuckApp
enter number
1024
Number is not duck
Number is not duck
Number is duck
Number is not duck

C:\Program Files\Java\jdk1.8.0_291\bin>
```

Note: if we think about above code we get output number is not duck three times and number is duck single time.

If we think about above output then above code not generate property output to us

Means we expect program should generate only one time number is duck or number is not duck but we use if else in loop so every time either if or else get executed so this is major reason we not get proper output so if we want to solve this problem we have to use flag concept.

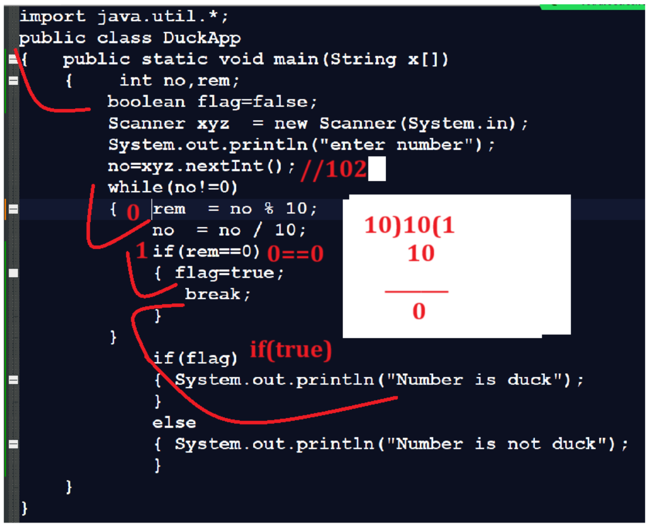
**Q. What is flag in programming?**

flag is boolean variable which either represent true or false

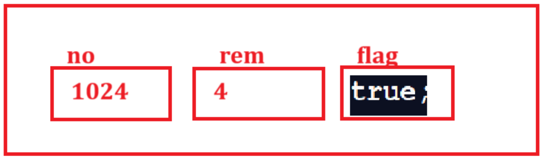
**Q. Why use flag variable in code?**

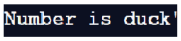
___________________________________________________________________________

If we think about loop and if we if else in loop then we get true and false output every time but if we required either true or false only once in code then we can use single if in code and use boolean variable and if update the boolean variable as per the condition in loop and if we break the loop after meeting condition and use if else outside loop and execute if or else according boolean variable value



```
import java.util.*;
public class DuckApp
{   public static void main(String x[])
    {     int no,rem;
          boolean flag=false;
          Scanner xyz  = new Scanner(System.in);
          System.out.println("enter number");
          no=xyz.nextInt();
          while(no!=0)
          { rem  = no % 10;
            no   = no / 10;
            if(rem==0)
            { flag=true;
              break;
            }
          }
            if(flag)
            { System.out.println("Number is duck");
            }
            else
            { System.out.println("Number is not duck");
            }
    }
}
```





```
Number is duck'
```

Example: WAP to input the number from keyboard and input the search digit from keyboard and check digit present in number or not.

Example: input number: 12345

Input digit for search 3

Output: digit found

Example: input number: 12345

Input digit for search 6

Output: digit not found.

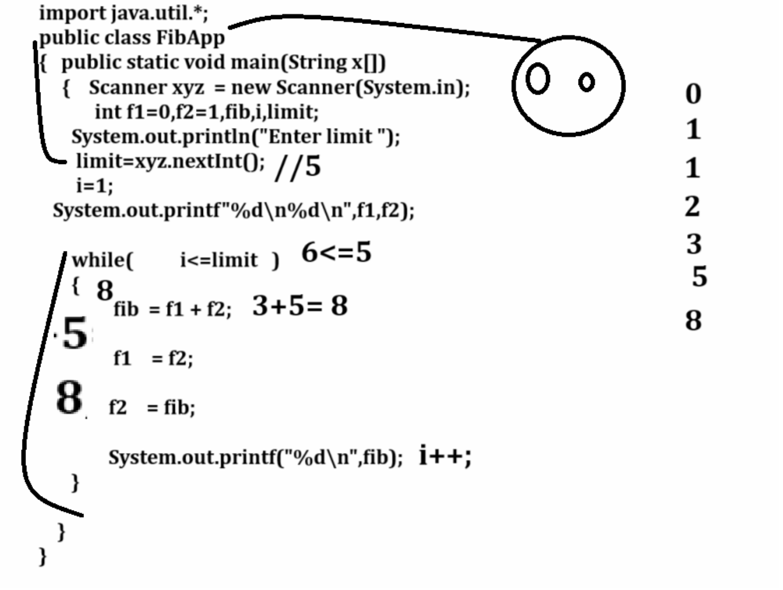
Example with source code

```
import java.util.*;
public class SearchDigitApp
{   public static void main(String x[ ])
    {   Scanner xyz  = new Scanner(System.in);
        int no,rem,skey;
                    boolean flag=false;
        System.out.println("Enter number from keyboard");
        no=xyz.nextInt();
       System.out.println("Enter search key from keyboard");
        skey=xyz.nextInt();
        while( no!=0 )
        {
                rem = no % 10;
                no  = no /10;
                if(rem==skey)
                                    { flag=true;
                              break;
                                    }
       }
                          if(flag)
              {   System.out.println("Search key found");
              }
             else{
               System.out.println("Search key not found");
             }
   }
}
```

**Example:**

**WAP to print the fibonacii series like as**

**0 1 1 2 3 5 8 13 21 34 etc**



```
   import java.util.*;
   public class FibApp
   {   public static void main(String x[])
     {    Scanner xyz  = new Scanner(System.in);
           int f1=0,f2=1,fib,i,limit;
       System.out.println("Enter limit ");
        limit=xyz.nextInt();
        i=1;
     System.out.printf"%d\n%d\n",f1,f2);

       while(       i<=limit  )
       {
           fib  = f1 + f2;

           f1   = f2;

          f2   = fib;

          System.out.printf("%d\n",fib);
       }

     }
   }
```

**Example with source code**

```
import java.util.*;
public class FibApp
{   public static void main(String x[ ])
    {   Scanner xyz  = new Scanner(System.in);
        int f1=0,f2=1,fib,i,limit;
                     System.out.println("Enter number of iterations");
                     limit=xyz.nextInt();
                     System.out.printf("%d\n%d\n",f1,f2);
                     i=1;
       while( i<=limit)
       {
            fib=f1+f2;
                                        f1=f2;
                                        f2=fib;
```

```
                                 System.out.printf("%d\n",fib);
                                 i++;
     }


  }
}
```

**Example:** WAP input the number and check number present in Fibonacci series or not

```
import java.util.*;
public class FibApp
{   public static void main(String x[ ])
    {   Scanner xyz  = new Scanner(System.in);
         int f1=0,f2=1,fib,i,limit,skey;
                 boolean b=false;
                 System.out.println("Enter number of iterations");
                 limit=xyz.nextInt();
                 i=1;
                 System.out.println("Enter search key");
                 skey=xyz.nextInt();
                  System.out.printf("%d\n%d\n",f1,f2);

     while( i<=limit)
     {
         fib=f1+f2;
                                 if(fib==skey)
                                 { b=true;

                                 }
                                 f1=f2;
                                 f2=fib;
                                 System.out.printf("%d\n",fib);
                                 i++;
     }
                 if(b)
                 { System.out.println("Number present in fibonacii series");
                 }
                 else
                 { System.out.println("Number not present in fibonacii series");
                 }
```

```
   }
}
```

Example: WAP to input the number and find the occurrence of every digit in number 1231245

```
Output: 1  --- 2
         2 --- 2
         3 – 1
        4 – 1
        5  - 1
```

**For loop**

---

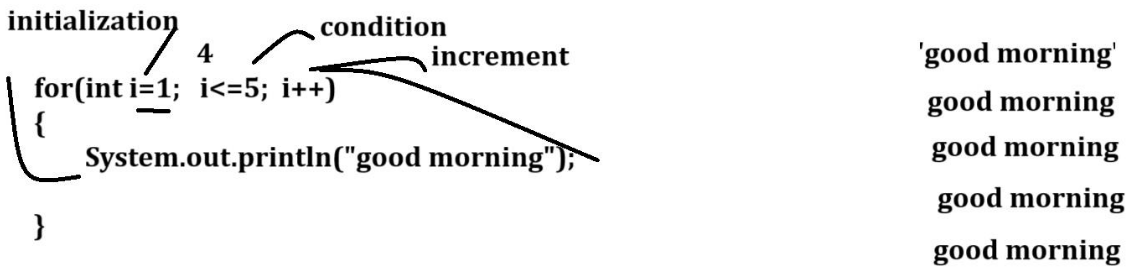
for loop is same like as while loop with some major differences

**Syntax**

---

```
for(initialization;   condition;    increment or decrement)
{   write here your logics
}
```

Example: WAP to print the good morning 5 times.



```
for(int i=1;  i<=5;  i++)
{
    System.out.println("good morning");

}
```

```
'good morning'
good morning
good morning
good morning
good morning
```

**Example with source code**

```
public class ForApp
{
  public static void main(String x[])
  {        for(int i=1; i<=5; i++)
            {
              System.out.println("good morning");
            }
  }
}
```

**You can use for loop in following fashion.**

```
public class ForApp
```

```
{
  public static void main(String x[])
  {    int i=1;
          for(; i<=5; i++)
          {
            System.out.println("good morning");
          }
  }
}
```

or

```
public class ForApp
{
  public static void main(String x[])
  {    int i=1;
          for(; i<=5; )
          {
            System.out.println("good morning");
            i++;
          }
  }
}
```

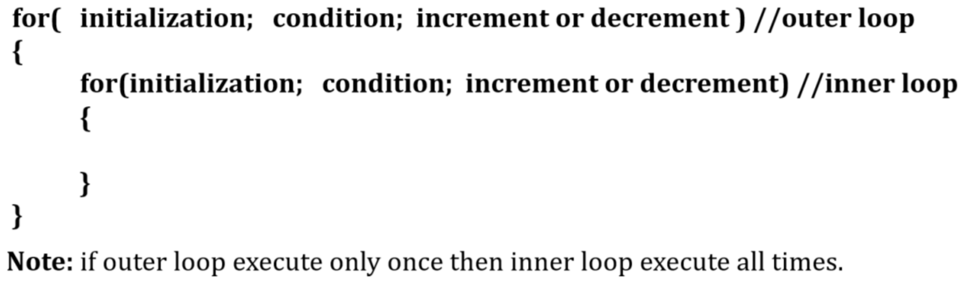
**Example:** WAP to input the two values consider first as base and second index and calculate its power.

```
import java.util.*;
public class PowerApp
{   public static void main(String x[])
   {  Scanner xyz  = new Scanner(System.in);
          int base,index,p=1;
          System.out.println("Enter base and index");
          base=xyz.nextInt();
          index=xyz.nextInt();
          for(int i=1; i<=index;i++)
          {  p = p *base;
          }
          System.out.printf("\nPower is %d\n",p);
   }
}
```

**Nested for loop or Nested loop**

Nested loop means loop within loop called as nested loop
There are two types of loop in nested loop.

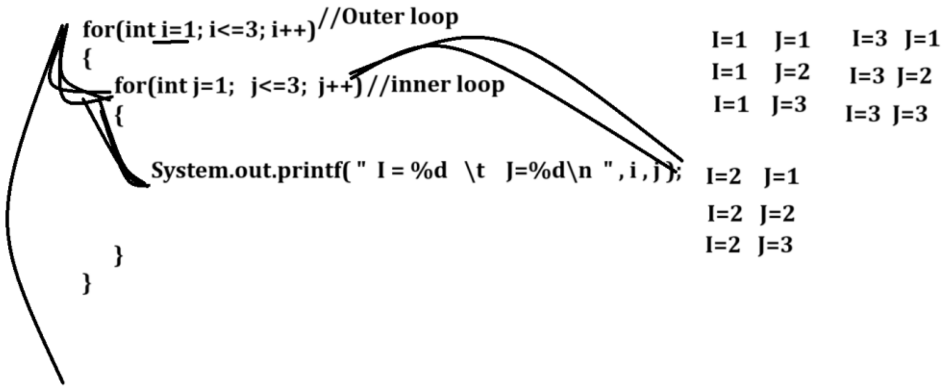
**Syntax of nested loop**



```
for(   initialization;   condition;  increment or decrement ) //outer loop
{
        for(initialization;   condition;  increment or decrement) //inner loop
        {

        }
}
```

**Note:** if outer loop execute only once then inner loop execute all times.

**Example of nested loop**



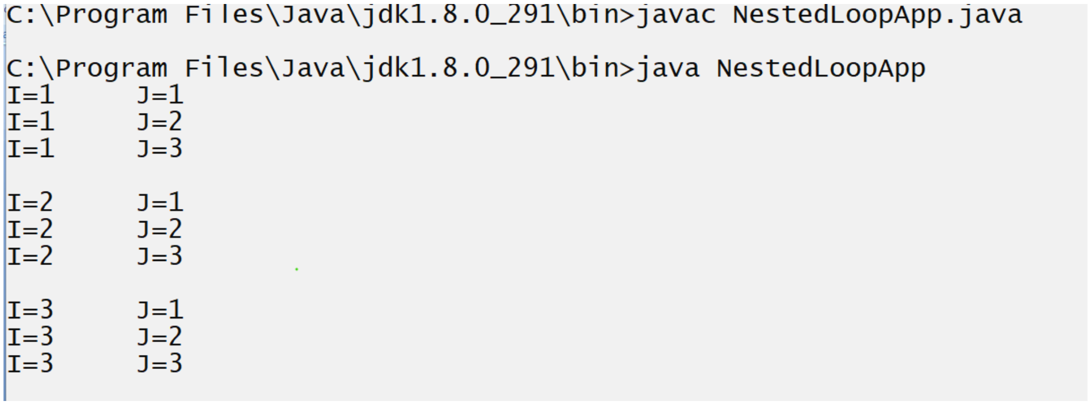
```
for(int i=1; i<=3; i++)//Outer loop
{
  for(int j=1;  j<=3;  j++) //inner loop
  {
      System.out.printf( "  I = %d   \t   J=%d\n  " , i , j );

  }
}
```

```
I=1   J=1     I=3  J=1
I=1   J=2     I=3  J=2
I=1   J=3     I=3  J=3

I=2   J=1
I=2   J=2
I=2   J=3
```

**Example with source code**

```
public class NestedLoopApp
{
  public static void main(String x[])
  {
          for(int i=1; i<=3;i++)
          {
            for(int j=1; j<=3; j++)
            {
              System.out.printf("I=%d\tJ=%d\n",i,j);
            }
          System.out.printf("\n");
          }
  }
}
```

**Output**



```
C:\Program Files\Java\jdk1.8.0_291\bin>javac NestedLoopApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java NestedLoopApp
I=1     J=1
I=1     J=2
I=1     J=3

I=2     J=1
I=2     J=2
I=2     J=3

I=3     J=1
I=3     J=2
I=3     J=3
```

**Q. What is benefit of nested loop?**

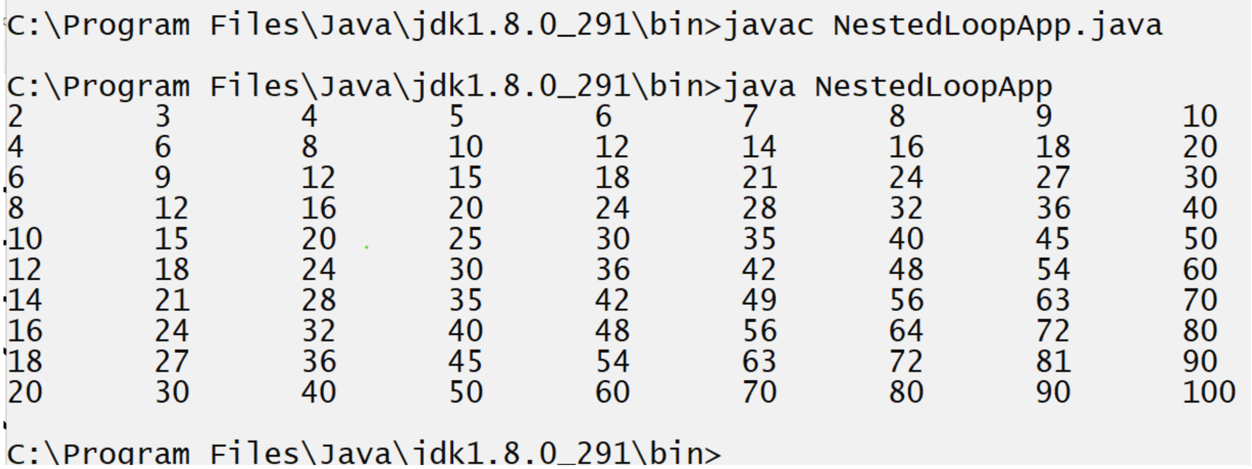
______________________________________________________________________________

Nested loop help us to work with matrix means here outer loop work as row and inner loop work as column.

**Example with source code**

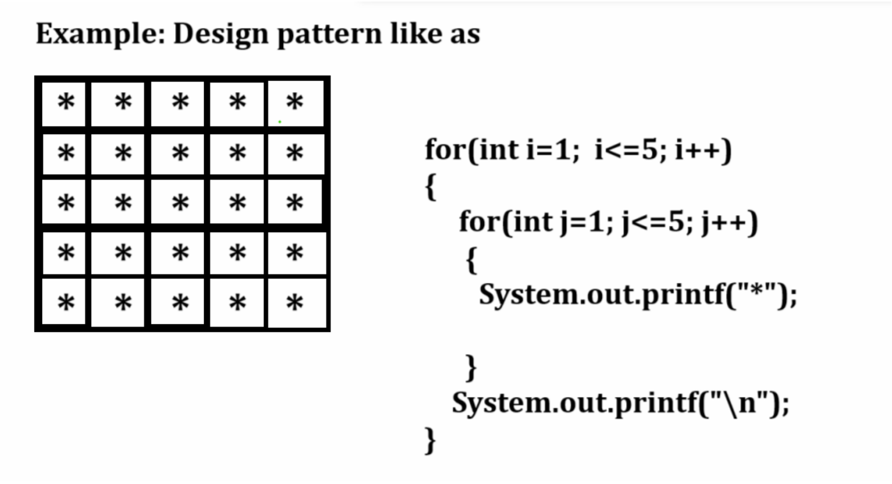
```
public class NestedLoopApp
{   public static void main(String x[])throws Exception
  {    for(int i=1; i<=10;i++)
            {
              for(int j=2; j<=10; j++)
              {
                System.out.printf("%d\t",i*j);
                 Thread.sleep(100);
              }
            System.out.printf("\n");
            }
  }
}
```

**Output**



```
C:\Program Files\Java\jdk1.8.0_291\bin>javac NestedLoopApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java NestedLoopApp
2       3       4       5       6       7       8       9       10
4       6       8       10      12      14      16      18      20
6       9       12      15      18      21      24      27      30
8       12      16      20      24      28      32      36      40
10      15      20      25      30      35      40      45      50
12      18      24      30      36      42      48      54      60
14      21      28      35      42      49      56      63      70
16      24      32      40      48      56      64      72      80
18      27      36      45      54      63      72      81      90
20      30      40      50      60      70      80      90      100

C:\Program Files\Java\jdk1.8.0_291\bin>
```

**Now we want to work with pattern programming or start patterns , number patterns , combination of number and start, alphabetical pattern etc**



**Example: Design pattern like as**

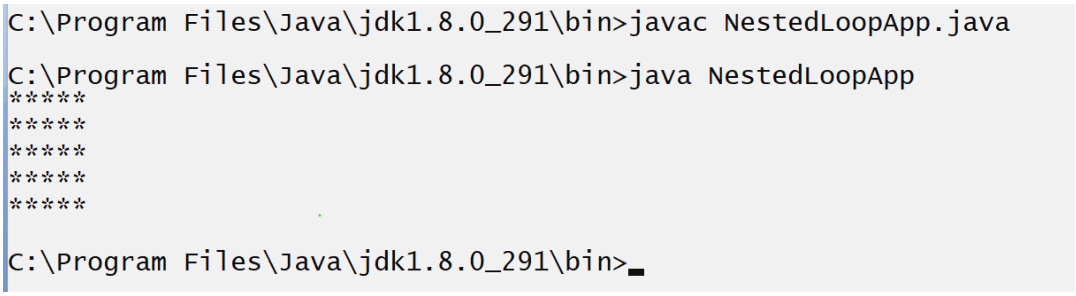
| | | | | |
|---|---|---|---|---|
| * | * | * | * | * |
| * | * | * | * | * |
| * | * | * | * | * |
| * | * | * | * | * |
| * | * | * | * | * |

```
for(int i=1;  i<=5; i++)
{
    for(int j=1; j<=5; j++)
     {
       System.out.printf("*");

     }
   System.out.printf("\n");
}
```

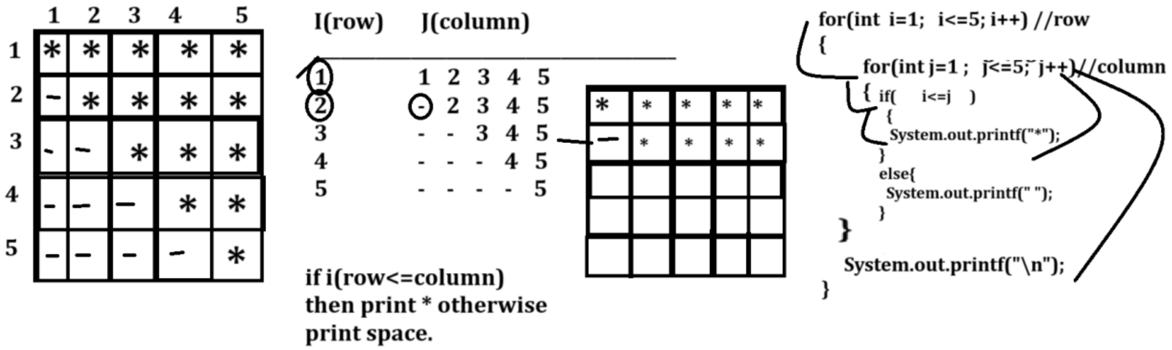
**Example with source code**

```
public class NestedLoopApp
{
  public static void main(String x[])
  {
          for(int i=1; i<=5;i++)
          {
            for(int j=1; j<=5; j++)
            {
              System.out.printf("*");
            }
          System.out.printf("\n");
          }
  }
}
```

**Output**



```
C:\Program Files\Java\jdk1.8.0_291\bin>javac NestedLoopApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java NestedLoopApp
*****
*****
*****
*****
*****

C:\Program Files\Java\jdk1.8.0_291\bin>
```

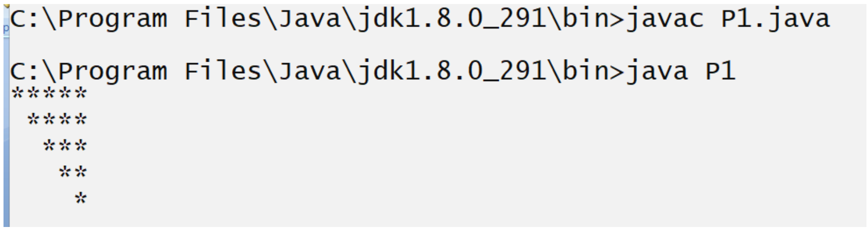


```
for(int i=1; i<=5; i++) //row
{
   for(int j=1 ; j<=5; j++)//column
   { if(    i<=j   )
      {
      System.out.printf("*");
      }
     else{
      System.out.printf(" ");
      }
   }
   System.out.printf("\n");
}
```

if i(row<=column)
then print * otherwise
print space.

**Example with source code**

```
public class P1
{
  public static void main(String x[])
  {
            int i,j;
            for(i=1; i<=5;  i++)
            {
              for(j=1; j<=5; j++)
              {
                        if(i<=j)
                        { System.out.printf("*");
                        }
                        else{
                         System.out.printf(" ");
                   }
          }
            System.out.printf("\n");
            }
  }
}
```

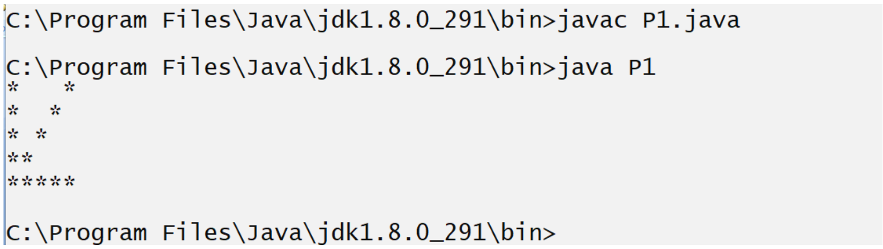


```
C:\Program Files\Java\jdk1.8.0_291\bin>javac P1.java

C:\Program Files\Java\jdk1.8.0_291\bin>java P1
*****
 ****
  ***
   **
    *
```

**Example with source code**

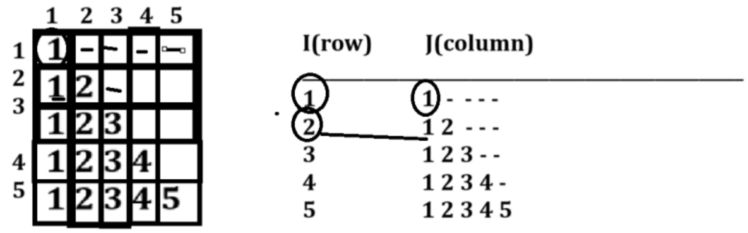
```
public class P1
{  public static void main(String x[])
   {    int i,j;
            for(i=1; i<=5;  i++)
            {
              for(j=1; j<=5; j++)
              {
                        if(j==1 || i==5 || j==6-i)
                        { System.out.printf("*");
                        }
                        else{
                         System.out.printf(" ");
                    }
           }
            System.out.printf("\n");
            }
   }
}
```

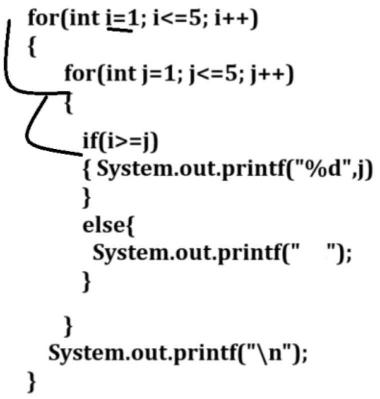
**Output**



```
C:\Program Files\Java\jdk1.8.0_291\bin>javac P1.java

C:\Program Files\Java\jdk1.8.0_291\bin>java P1
*   *
*  *
* *
**
*****

C:\Program Files\Java\jdk1.8.0_291\bin>
```

Example: WAP to print the following patterns?



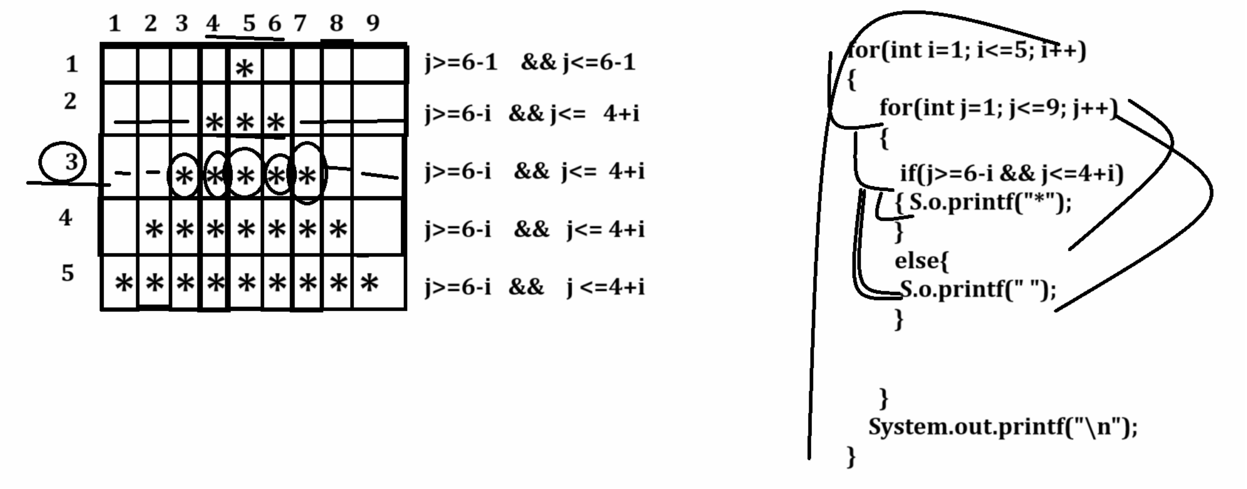


```
for(int i=1; i<=5; i++)
{
    for(int j=1; j<=5; j++)
    {
      if(i>=j)
      { System.out.printf("%d",j)
      }
      else{
        System.out.printf("    ");
      }

    }
  System.out.printf("\n");
}
```

**Example with source code**

```
public class P1
{
  public static void main(String x[])
  {
          int i,j;
          for(i=1; i<=5;  i++)
          {
            for(j=1; j<=5; j++)
            {
                    if(i>=j)
                    { System.out.printf("%d",j);
                    }
                    else{
                     System.out.printf(" ");
                 }
        }
          System.out.printf("\n");
          }
  }
}
```

Example : WAP to print the following pattern?



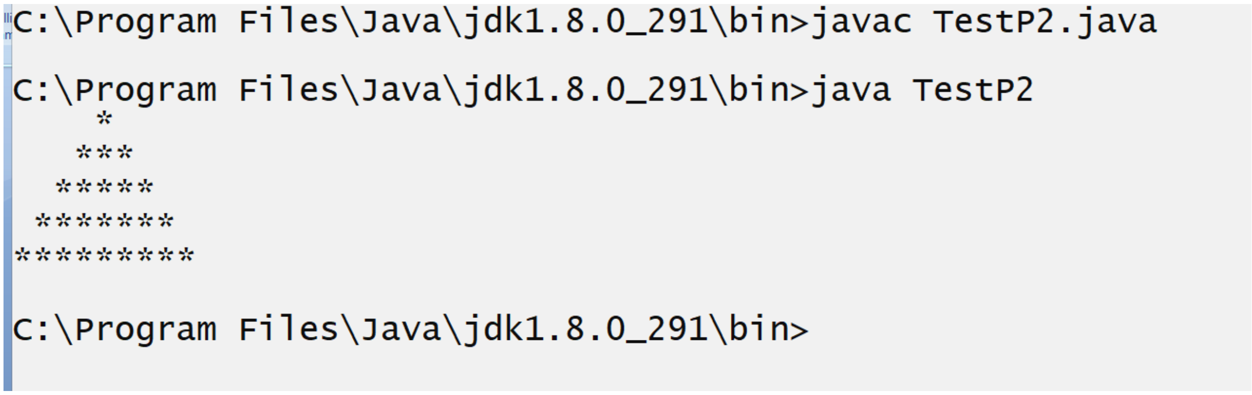
|   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |   |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 |   |   |   |   | * |   |   |   |   | j>=6-1 && j<=6-1 |
| 2 | — | — | — | * | * | * | — | — | — | j>=6-i && j<= 4+i |
| 3 | - | - | * | * | * | * | * | - | — | j>=6-i && j<= 4+i |
| 4 |   | * | * | * | * | * | * | * |   | j>=6-i && j<= 4+i |
| 5 | * | * | * | * | * | * | * | * | * | j>=6-i && j <=4+i |

```
for(int i=1; i<=5; i++)
{
  for(int j=1; j<=9; j++)
   {
    if(j>=6-i && j<=4+i)
    { S.o.printf("*");
    }
    else{
     S.o.printf(" ");
    }


  }
 System.out.printf("\n");
}
```

**Example with source code**

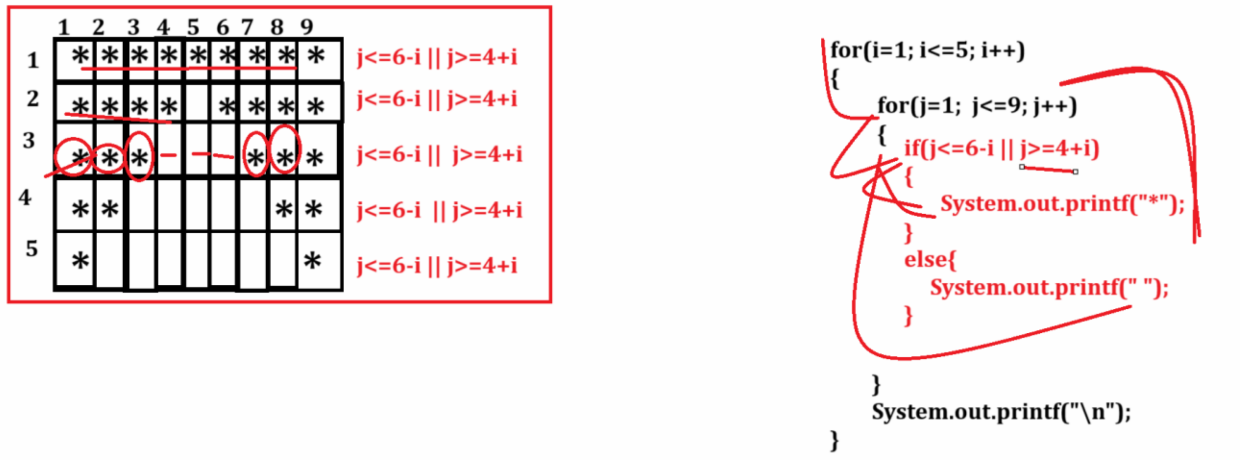
```
public class TestP2
{
  public static void main(String x[])
  {
           int i,j;
          for(i=1; i<=5; i++)
          {
            for(j=1; j<=9;j++)
            {
                 if(j>=6-i && j<=4+i)
                 { System.out.printf("*");
                 }
                else{
                 System.out.printf(" ");
                }
        }
           System.out.printf("\n");
          }
  }
}
```

**Output**



```
C:\Program Files\Java\jdk1.8.0_291\bin>javac TestP2.java

C:\Program Files\Java\jdk1.8.0_291\bin>java TestP2
    *
   ***
  *****
 *******
*********

C:\Program Files\Java\jdk1.8.0_291\bin>
```

**Example: WAP to print the following pattern?**



|   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |   |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | * | * | * | * | * | * | * | * | * | j<=6-i \|\| j>=4+i |
| 2 | * | * | * | * |   | * | * | * | * | j<=6-i \|\| j>=4+i |
| 3 | * | * | * | - | - | - | * | * | * | j<=6-i \|\|  j>=4+i |
| 4 | * | * |   |   |   |   |   | * | * | j<=6-i  \|\| j>=4+i |
| 5 | * |   |   |   |   |   |   |   | * | j<=6-i \|\| j>=4+i |

```
for(i=1; i<=5; i++)
{
    for(j=1;  j<=9; j++)
    {  if(j<=6-i || j>=4+i)
       {
          System.out.printf("*");
       }
       else{
         System.out.printf(" ");
       }

    }
    System.out.printf("\n");
}
```

**Example with source code**

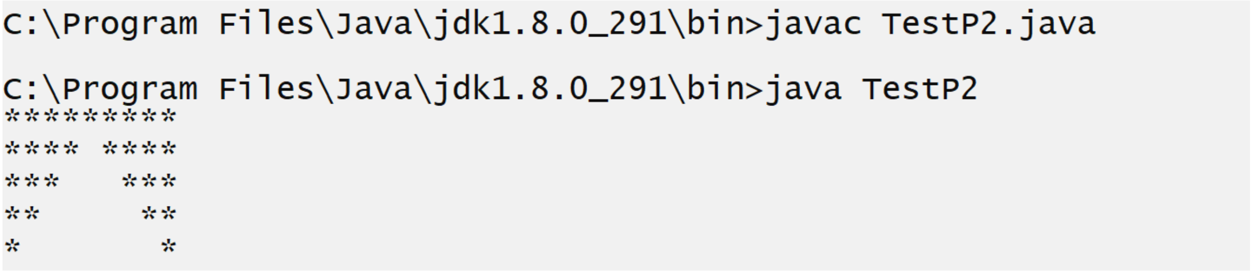
**public class TestP2**

**{**

**public static void main(String x[])**

```
  {
            int i,j;
           for(i=1; i<=5; i++)
           {
             for(j=1; j<=9;j++)
             {
                        if(j<=6-i || j>=4+i)
                        { System.out.printf("*");
                         }
                        else{
                         System.out.printf(" ");
                        }
        }
            System.out.printf("\n");
           }
  }
}
```

**Output**

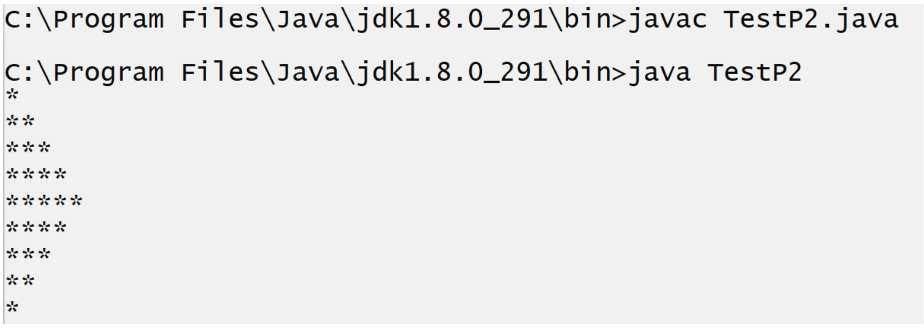


```
C:\Program Files\Java\jdk1.8.0_291\bin>javac TestP2.java

C:\Program Files\Java\jdk1.8.0_291\bin>java TestP2
*********
**** ****
***   ***
**     **
*       *
```

**Example with source code**

```
public class TestP2
{ public static void main(String x[])
  {    int i,j;
           for(i=1; i<=9; i++)
           {  for(j=1; j<=5;j++)
             {    String s=((i>=j && i<=5)||(j<=10-i && i>=6))?"*":" ";
                System.out.print(s);
             }
             System.out.printf("\n");
           }
  }
}
```

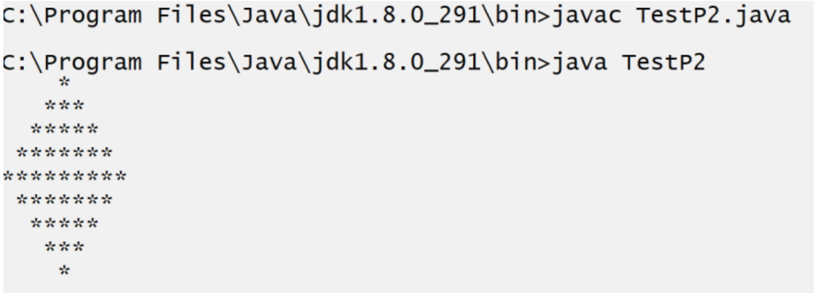
**Output**



```
C:\Program Files\Java\jdk1.8.0_291\bin>javac TestP2.java

C:\Program Files\Java\jdk1.8.0_291\bin>java TestP2
*
**
***
****
*****
****
***
**
*
```

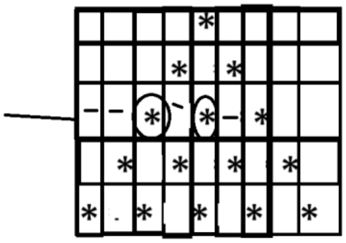
**Example with source code**

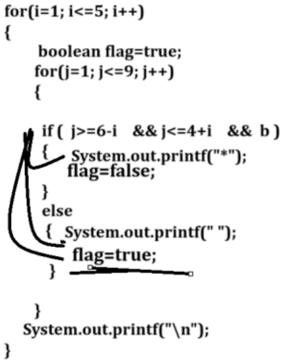
```
public class TestP2
{
  public static void main(String x[])
  {    int i,j;
           for(i=1; i<=9; i++)
           {
         for(j=1; j<=9;j++)
             {
               if((j>=6-i && j<=4+i && i<=5) ||(j>=i-4&&j<=14-i&&i>=6))
               { System.out.printf("*");
               }
             else{
            System.out.printf(" ");
           }


         }
            System.out.printf("\n");
           }
  }
}
```



```
C:\Program Files\Java\jdk1.8.0_291\bin>javac TestP2.java

C:\Program Files\Java\jdk1.8.0_291\bin>java TestP2
    *
   ***
  *****
 *******
*********
 *******
  *****
   ***
    *
```

Example: WAP to print the following pattern?



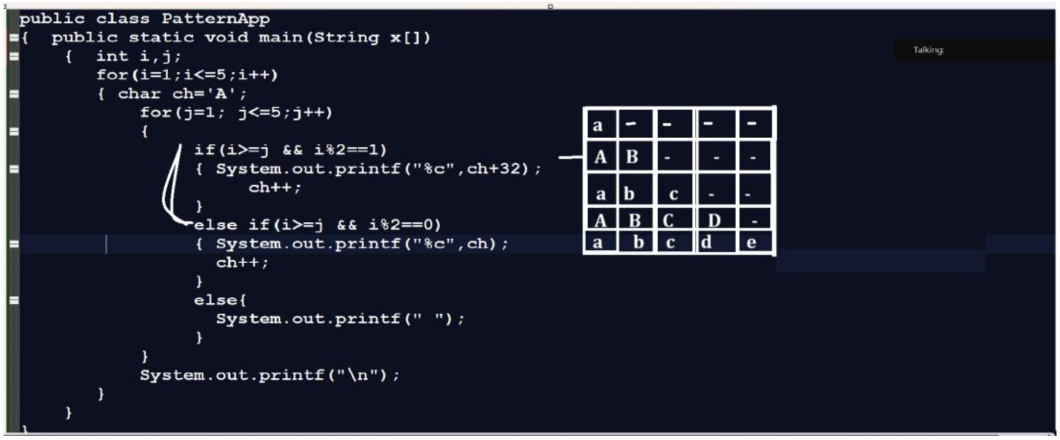


```
for(i=1; i<=5; i++)
{
     boolean flag=true;
    for(j=1; j<=9; j++)
    {

      if (  j>=6-i   && j<=4+i   &&  b )
      {  System.out.printf("*");
         flag=false;
      }
      else
      {  System.out.printf(" ");
         flag=true;
       }

    }
  System.out.printf("\n");
}
```

Example: with source code

```
public class PatternApp
{
   public static void main(String x[])
           {
             int i,j;
             for(i=1;i<=5;i++)
             {boolean b=true;
                for(j=1; j<=9;j++)
                       {
                          if(j>=6-i&&j<=4+i&&b)
                                 { System.out.printf("*");
                                   b=false;
                                 }
                                 else{
                                   System.out.printf(" ");
                                         b=true;
                                 }
                       }
                       System.out.printf("\n");
             }
           }
}
```

**Example: WAP to print the following pattern?**



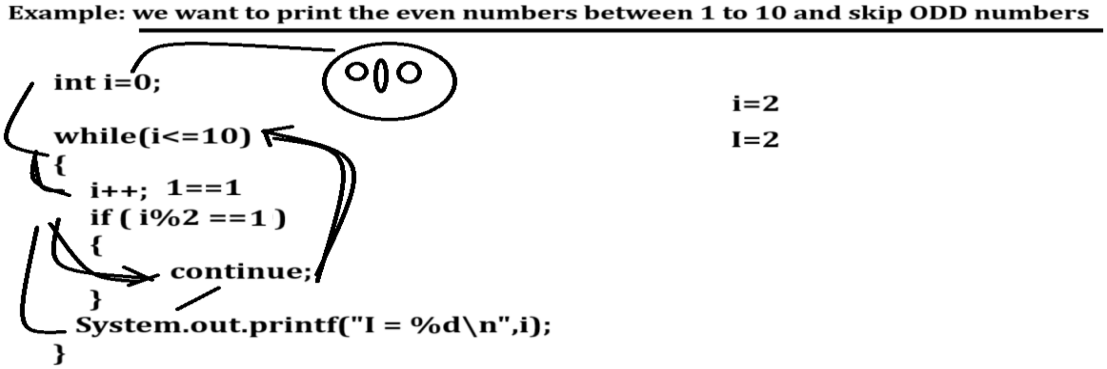
```
public class PatternApp
{   public static void main(String x[])
    {   int i,j;
        for(i=1;i<=5;i++)
        { char ch='A';
              for(j=1; j<=5;j++)
              {
                 if(i>=j && i%2==1)
                 { System.out.printf("%c",ch+32);
                      ch++;
                 }
                 else if(i>=j && i%2==0)
                 { System.out.printf("%c",ch);
                   ch++;
                 }
                 else{
                   System.out.printf(" ");
                 }
              }
              System.out.printf("\n");
        }
    }
}
```

| | | | | |
|---|---|---|---|---|
| a | - | - | - | - |
| A | B | - | - | - |
| a | b | c | - | - |
| A | B | C | D | - |
| a | b | c | d | e |

**Continue keyword**

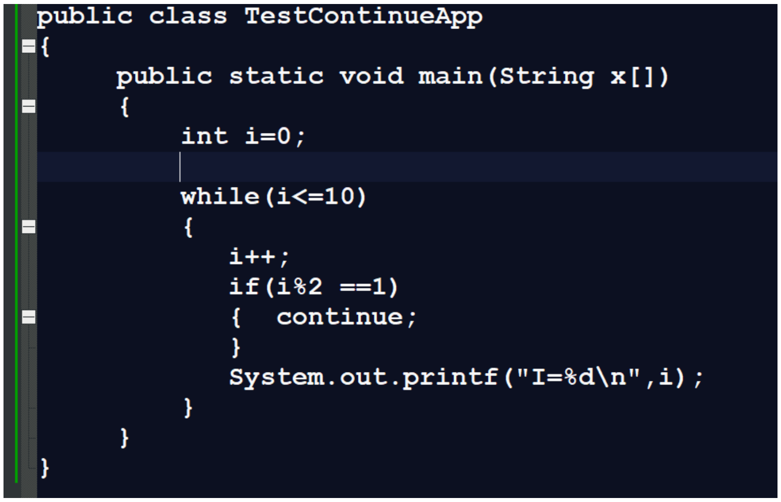
______________________________________________________________________________

Continue keyword is used for skip the some execution steps of loop called as continue keyword.



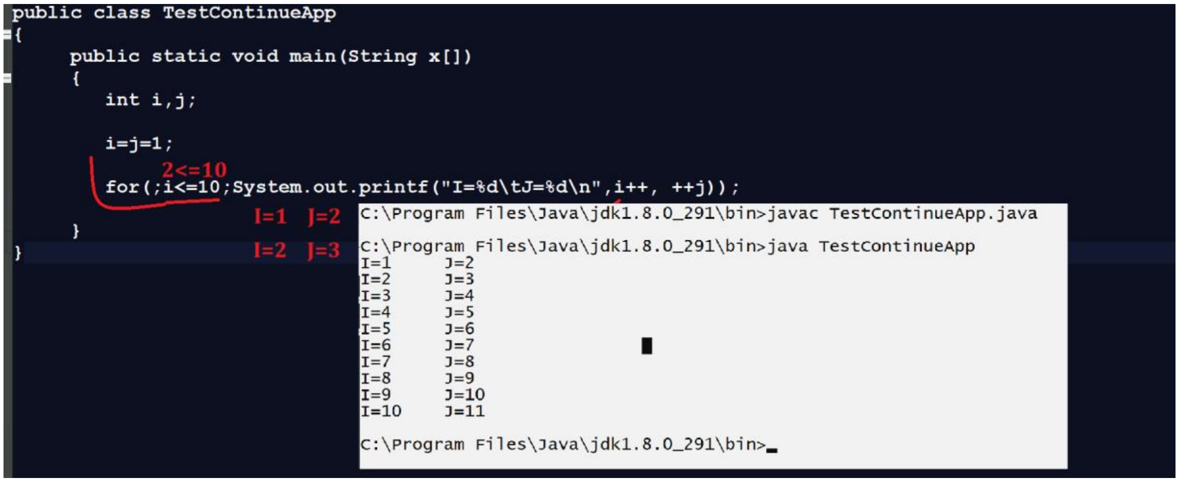
**Example: we want to print the even numbers between 1 to 10 and skip ODD numbers**

```
int i=0;

while(i<=10)
{
   i++;  1==1
   if ( i%2 ==1 )
   {
         continue;
   }
  System.out.printf("I = %d\n",i);
}
```



```
public class TestContinueApp
{
     public static void main(String x[])
     {
         int i=0;

         while(i<=10)
         {
            i++;
            if(i%2 ==1)
            {   continue;
            }
            System.out.printf("I=%d\n",i);
         }
     }
}
```

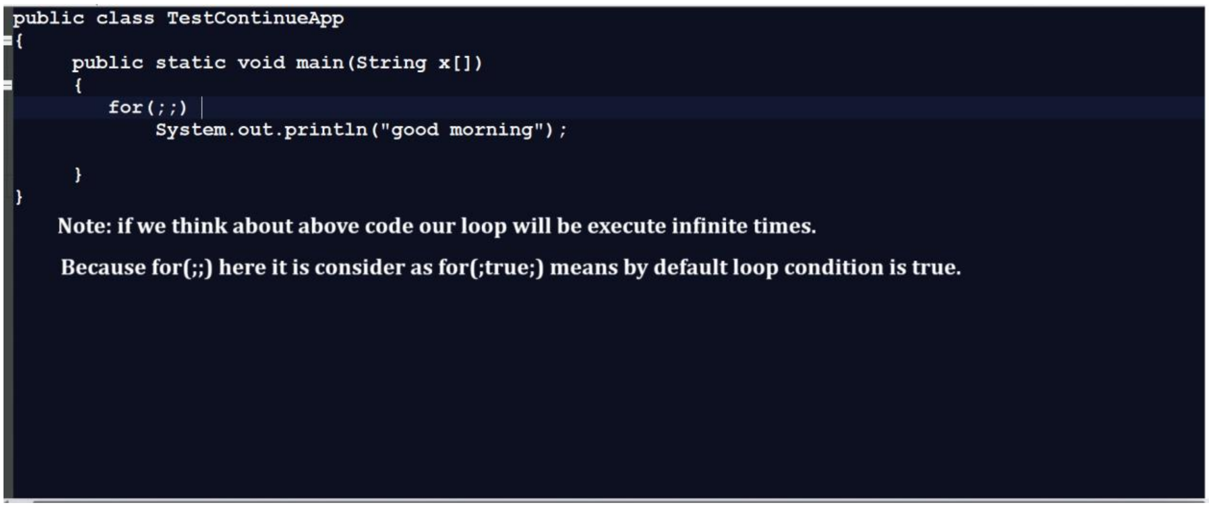
**Q. What will be output of given code?**



```
public class TestContinueApp
{
     public static void main(String x[])
     {
        int i,j;

        i=j=1;
        for(;i<=10;System.out.printf("I=%d\tJ=%d\n",i++, ++j));
     }
}
```

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac TestContinueApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java TestContinueApp
I=1     J=2
I=2     J=3
I=3     J=4
I=4     J=5
I=5     J=6
I=6     J=7
I=7     J=8
I=8     J=9
I=9     J=10
I=10    J=11

C:\Program Files\Java\jdk1.8.0_291\bin>_
```

**Q. What will be output of given code?**

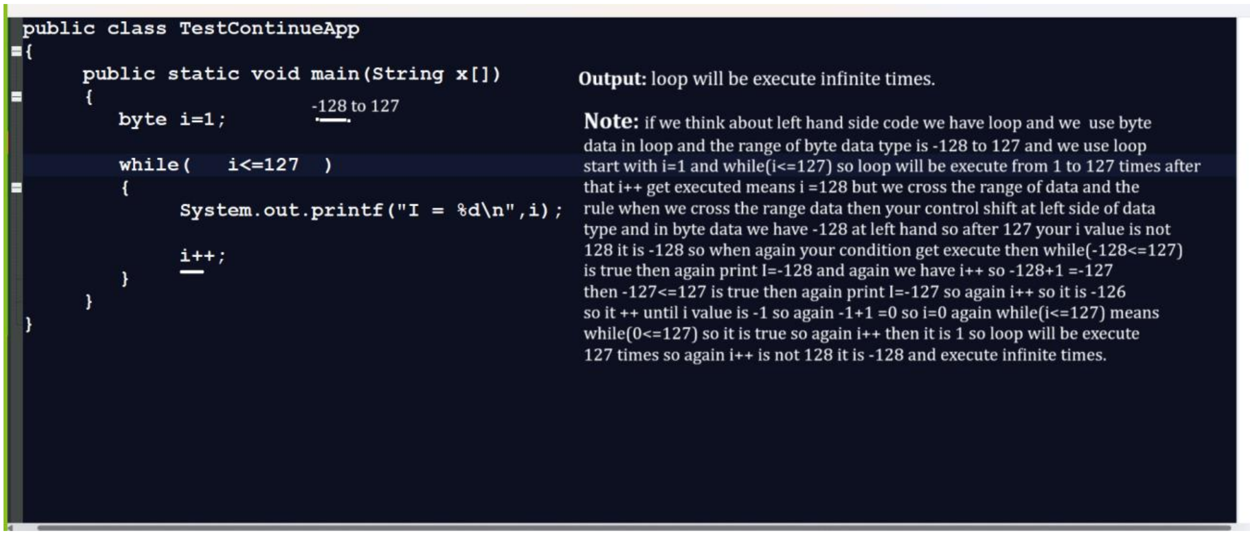


```
public class TestContinueApp
{
     public static void main(String x[])
     {
        for(;;)
            System.out.println("good morning");

     }
}
```

**Note: if we think about above code our loop will be execute infinite times.**

**Because for(;;) here it is consider as for(;true;) means by default loop condition is true.**

Q. What will be output of given code?

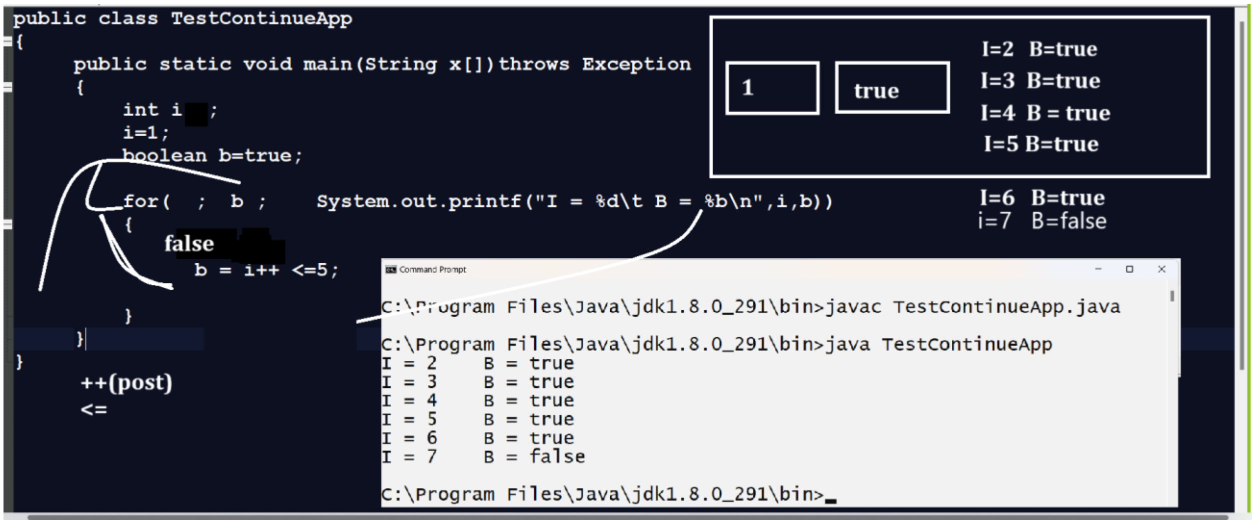


```
public class TestContinueApp
{
     public static void main(String x[])
     {
        byte i=1;

        while(   i<=127  )
        {
             System.out.printf("I = %d\n",i);

             i++;
        }
     }
}
```

**Output:** loop will be execute infinite times.

**Note:** if we think about left hand side code we have loop and we use byte data in loop and the range of byte data type is -128 to 127 and we use loop start with i=1 and while(i<=127) so loop will be execute from 1 to 127 times after that i++ get executed means i =128 but we cross the range of data and the rule when we cross the range data then your control shift at left side of data type and in byte data we have -128 at left hand so after 127 your i value is not 128 it is -128 so when again your condition get execute then while(-128<=127) is true then again print I=-128 and again we have i++ so -128+1 =-127 then -127<=127 is true then again print I=-127 so again i++ so it is -126 so it ++ until i value is -1 so again -1+1 =0 so i=0 again while(i<=127) means while(0<=127) so it is true so again i++ then it is 1 so loop will be execute 127 times so again i++ is not 128 it is -128 and execute infinite times.

**Q. what will be output of given code?**



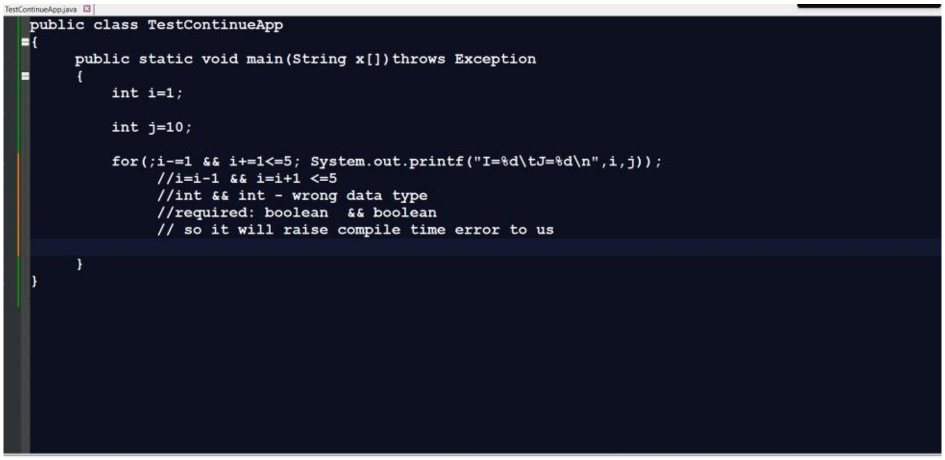
```
public class TestContinueApp
{
      public static void main(String x[])throws Exception
      {
          int i [redacted];
          i=1;
          boolean b=true;

          for(  ;  b ;    System.out.printf("I = %d\t B = %b\n",i,b))
          {
              false
                 b = i++ <=5;

          }
      }
}
```

++(post)
<=

I=2  B=true
I=3  B=true
I=4  B = true
I=5 B=true
I=6  B=true
i=7  B=false

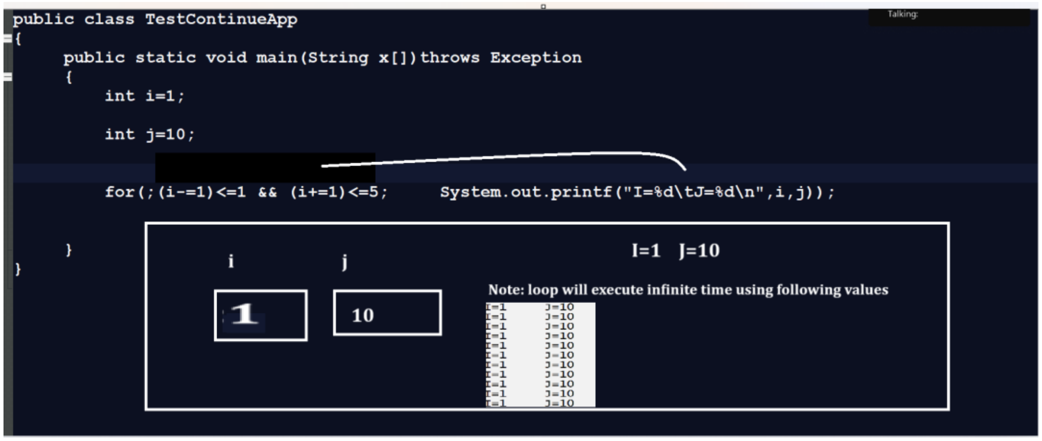
```
C:\Program Files\Java\jdk1.8.0_291\bin>javac TestContinueApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java TestContinueApp
I = 2     B = true
I = 3     B = true
I = 4     B = true
I = 5     B = true
I = 6     B = true
I = 7     B = false

C:\Program Files\Java\jdk1.8.0_291\bin>_
```

**Q. what will be output of given code?**



```
public class TestContinueApp
{
     public static void main(String x[])throws Exception
     {
         int i=1;

         int j=10;

         for(;i-=1 && i+=1<=5; System.out.printf("I=%d\tJ=%d\n",i,j));
             //i=i-1 && i=i+1 <=5
             //int && int - wrong data type
             //required: boolean  && boolean
             // so it will raise compile time error to us

     }
}
```

Q. what will be output of given code?



```
public class TestContinueApp
{
     public static void main(String x[])throws Exception
     {
         int i=1;

         int j=10;

         [redacted]
         for(;(i-=1)<=1 && (i+=1)<=5;     System.out.printf("I=%d\tJ=%d\n",i,j));

     }
}
```

I=1  J=10

**Note: loop will execute infinite time using following values**

```
I=1      J=10
I=1      J=10
I=1      J=10
I=1      J=10
I=1      J=10
I=1      J=10
I=1      J=10
I=1      J=10
I=1      J=10
I=1      J=10
I=1      J=10
```