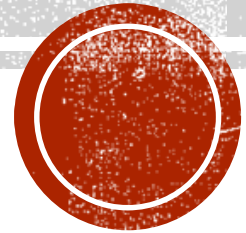
# EXERCISES

Nested Loops



# SESSION

- **Definition of nested loops**
- Samples of programs that require nested loops

# DEFINITION

A **nested loop** is a loop within a loop, an inner loop within the body of an outer loop.

# DEFINITION

A **nested loop** is a loop within a loop, an inner loop within the body of an outer loop.

```
while( expression 1)
{
      …..
      While(expression2)
      {
      …..
      }
}
```

```
while (expression 1)
{
      …..
for(…..;expression2;….)
      {
      …..
      }
}
```

```
while (expression 1)
{
      …..
      do
      {
      …..
      }while(expression2);
}
```

# DEFINITION

A **nested loop** is a loop within a loop, an inner loop within the body of an outer loop.

```
for (….; expression 1;…)
{
      …..
      for(…..;expression2;….)
      {
      …..
      }
}
```

```
for (….; expression 1;…)
{
      …..
      while (expression2)
      {
      …..
      }
}
```

```
for (expression 1)
{
      …..
      do
      {
      …..
      }while(expression2);
}
```

# DEFINITION

A **nested loop** is a loop within a loop, an inner loop within the body of an outer loop.

## How this works?

- At each iteration of the outer loop, the inner loop is triggered to be executed completely.
- This repeats until the outer loop finishes.

# DEFINITION

Example 1:

```
for(i=1;i<=4;i++)
{
    for(j=1;j<=3;j++)
        cout<<“i=”<<i<<“,j=”<<j<<“\n”;
    cout<<“end of loop j\n”;
}
```

- **Outer Loop:** *for* loop with variable i
- **Inner loop**: *for* loop with variable j
- At each iteration of the outer loop, the inner loop is triggered to be executed completely.

```
i=1,j=1
i=1,j=2
i=1,j=3
end of loop j
i=2,j=1
i=2,j=2
i=2,j=3
end of loop j
i=3,j=1
i=3,j=2
i=3,j=3
end of loop j
i=4,j=1
i=4,j=2
i=4,j=3
end of loop j
```

# DEFINITION

```
while (expression 1)
{
    instruction a;

    for(…..;expression2;….)
    {
        instruction 1;
        for(….;expression3;…)
            instruction2;
    }
    instruction b;
}
```

**Outer Loop? Inner loop?**

# DEFINITION

```
while (expression 1)
{
    instruction a;

    for(…..;expression2;….)
    {
        instruction 1;
        for(….;expression3;…)
            instruction2;
    }
    instruction b;
}
```

- At each iteration of while loop we have execution of :
  - instruction a
  - Loop for(…..;expression2;….)
  - instruction b.

➔ Loop for(…..;expression2;….) is an **inner loop** for the outer while loop.

➔ Loop while is an **outer loop** for for(…..;expression2;….) loop.

# DEFINITION

```
while (expression 1)
{
    instruction a;

    for(…..;expression2;….)
    {
        instruction 1;
        for(….;expression3;…)
            instruction2;
    }
    instruction b;
}
```

- At each iteration of for(…..;expression2;….) loop we have execution of :
  - instruction 1
  - Loop for(…..;expression3;….)

➔ Loop for(…..;expression3;….) is an **inner loop** for the outer for(…..;expression2;….) loop.

➔ Loop for(…..;expression2;….) is an **outer loop** for the outer Loop for(…..;expression3;….) loop.

# Definition

**Example**:

```
int main()
{
    int i,j,k;
    for(i=1;i<=3;i++)
    {
        cout<<"i="<<i<<"\n";
        for(j=1;j<=3;j++)
        {
            cout<<"j="<<j <<"\n";
            for(k=j; k<=4;k++)
                cout<<"k="<<k<<"\t";
            cout<<"end of loop k\n";
        }
        cout<<" end of loop j\n";
    }
    cout<<" end of loop i\n";
return 0;
}
```

```
i=1
j=1
k=1     k=2     k=3     k=4     end of loop k
j=2
k=2     k=3     k=4     end of loop k
j=3
k=3     k=4     end of loop k
 end of loop j
i=2
j=1
k=1     k=2     k=3     k=4     end of loop k
j=2
k=2     k=3     k=4     end of loop k
j=3
k=3     k=4     end of loop k
 end of loop j
i=3
j=1
k=1     k=2     k=3     k=4     end of loop k
j=2
k=2     k=3     k=4     end of loop k
j=3
k=3     k=4     end of loop k
 end of loop j
 end of loop i

--------------------------------
Process exited after 0.03853 seconds with return value 0
Press any key to continue . . .
```

# DEFINITION

A **nested loop** is a loop within a loop, an inner loop within the body of an outer loop.

## How this works?

- At each iteration of the outer loop, the inner loop is triggered to be executed completely.
- This repeats until the outer loop finishes.
- Of course, a *break* can be used within either the inner or outer loop. Break would end the execution of the loop where break is belonging to.

# Definition

**Example**:

```
#include<iostream>
using namespace std;
int main()
{
    int i,j,k;
    for(i=1;i<=3;i++)
    {
        cout<<"i="<<i<<"\n";
        for(j=1;j<=3;j++)
        {
            cout<<"j="<<j <<"\n";
            break;
            for(k=j; k<=4;k++)
                cout<<"k="<<k<<"\t";
            cout<<"end of loop k\n";
        }
        cout<<" end of loop j\n";
    }
    cout<<" end of loop i\n";
return 0;
}
```

```
i=1
j=1
 end of loop j
i=2
j=1
 end of loop j
i=3
j=1
 end of loop j
 end of loop i

--------------------------------
Process exited after 1.583 seconds with return value 0
Press any key to continue . . .
```

# SESSION

- Definition of nested loops
- **Samples of programs that require nested loops**

**EXERCISE 1:**

WRITE A PROGRAM THAT READS 10 POSITIVE NUMBERS AND PRINTS THEIR SUM

# EXERCISE 1 -SOLUTION

WRITE A PROGRAM THAT READS 10 POSITIVE NUMBERS AND PRINTS THEIR SUM

```
#include<iostream>
using namespace std;
int main()
{
    int i;
    float x, S;
    for(i=1;i<=10;i++)
    {
        do
        {
            cout<<"\nenter a positive value ";
            cin>>x;
        }while(x<=0);
        cout<<"\nfor i= "<<i << " x= " <<x;
        S=S+x;
    }
    cout<<"\nS= "<<S;
}
```

```
enter a positive value -2

enter a positive value 3

for i= 1 x= 3
enter a positive value -5

enter a positive value -20

enter a positive value 14.3

for i= 2 x= 14.3
enter a positive value 16

for i= 3 x= 16
enter a positive value 19.5

for i= 4 x= 19.5
enter a positive value 20.7

for i= 5 x= 20.7
enter a positive value 39

for i= 6 x= 39
enter a positive value 25

for i= 7 x= 25
enter a positive value 12.6

for i= 8 x= 12.6
enter a positive value 16.4

for i= 9 x= 16.4
enter a positive value 13.4

for i= 10 x= 13.4
S= 179.9
--------------------------------
Process exited after 49.96 seconds with return value 0
Press any key to continue . . .
```

**EXERCISE 2:**

WRITE A PROGRAM THAT READS N (POSITIVE INTEGER NUMBER) AND PRINTS THE SUM OF:

$$1! + 2! + 3! + \cdots + n!$$

# EXERCISE 2-SOLUTION

WRITE A PROGRAM THAT READS N (POSITIVE INTEGER NUMBER) AND PRINTS THE SUM OF:

$$1! + 2! + 3! + \cdots + n!$$

```
int main()
{
int i,j,n;
double f,S;
do
{
    cout<< "enter the value of n>0 ";
    cin>>n;
}while(n<=0);

for (S=0,j=1;j<=n;j++)
{
    for (i=1, f=1; i<=j; i++)
        f=f*i;
    cout<<j<<"!="<<f<<"\n";
    S=S+f;
}
cout<<"S= "<<S;
return 0;
}
```

```
enter the value of n>0 5
1!=1
2!=2
3!=6
4!=24
5!=120
S= 153
--------------------------------
Process exited after 3.081 seconds with return value 0
Press any key to continue . . .
```

# EXERCISE 2- REMARK

```
enter the value of n>0 50
1!=1
2!=2
3!=6
4!=24
5!=120
6!=720
7!=5040
8!=40320
9!=362880
10!=3.6288e+006
11!=3.99168e+007
12!=4.79002e+008
13!=6.22702e+009
14!=8.71783e+010
15!=1.30767e+012
16!=2.09228e+013
17!=3.55687e+014
18!=6.40237e+015
19!=1.21645e+017
20!=2.4329e+018
21!=5.10909e+019
22!=1.124e+021
23!=2.5852e+022
24!=6.20448e+023
25!=1.55112e+025
26!=4.03291e+026
27!=1.08889e+028
28!=3.04888e+029
29!=8.84176e+030
30!=2.65253e+032
31!=8.22284e+033
32!=2.63131e+035
33!=8.68332e+036
34!=2.95233e+038
35!=1.03331e+040
36!=3.71993e+041
37!=1.37638e+043
38!=5.23023e+044
39!=2.03979e+046
40!=8.15915e+047
41!=3.34525e+049
42!=1.40501e+051
43!=6.04153e+052
44!=2.65827e+054
45!=1.19622e+056
46!=5.50262e+057
47!=2.58623e+059
48!=1.24139e+061
49!=6.08282e+062
50!=3.04141e+064
S= 3.10351e+064
--------------------------------
Process exited after 2.931 seconds with return value 0
Press any key to continue . . .
```

Case of :

```
double f,S;
```

Case of :

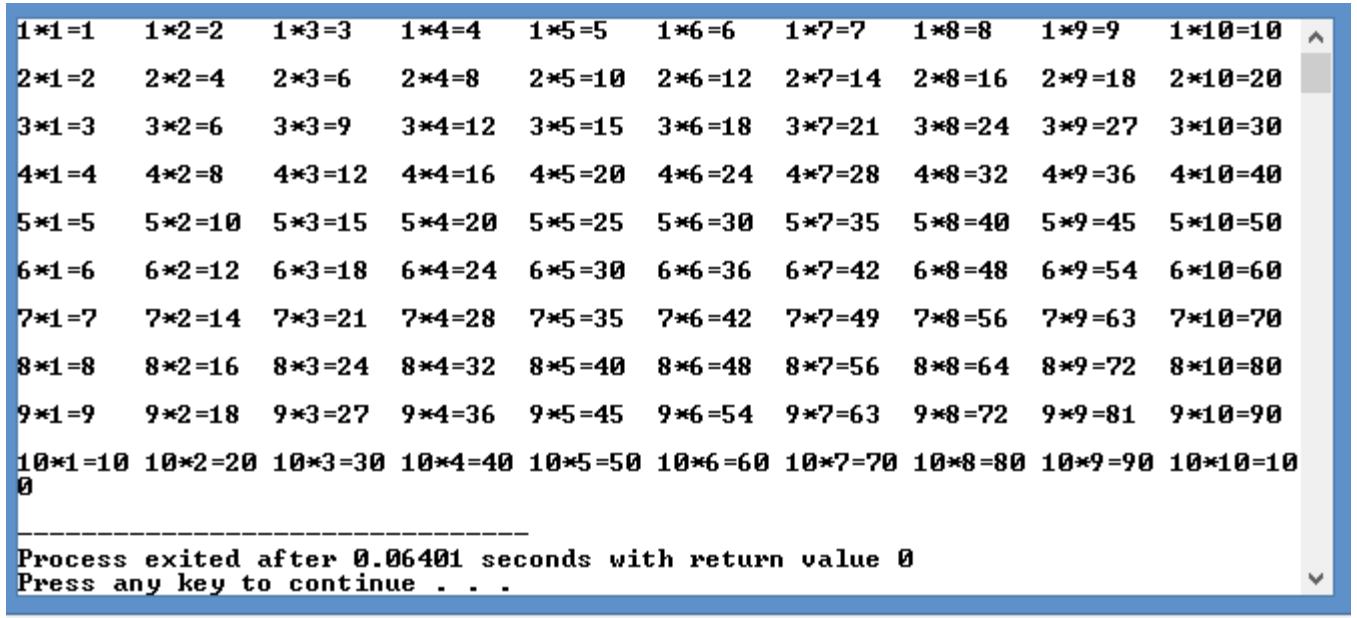
```
int f,S;
```

Incorrect results

```
enter the value of n>0 50
1!=1
2!=2
3!=6
4!=24
5!=120
6!=720
7!=5040
8!=40320
9!=362880
10!=3628800
11!=39916800
12!=479001600
13!=1932053504
14!=1278945280
15!=2004310016
16!=2004189184
17!=-288522240
18!=-898433024
19!=109641728
20!=-2102132736
21!=-1195114496
22!=-522715136
23!=862453760
24!=-775946240
25!=2076180480
26!=-1853882368
27!=1484783616
28!=-1375731712
29!=-1241513984
30!=1409286144
31!=738197504
32!=-2147483648
33!=-2147483648
34!=0
35!=0
36!=0
37!=0
38!=0
39!=0
40!=0
41!=0
42!=0
43!=0
44!=0
45!=0
46!=0
47!=0
48!=0
49!=0
50!=0
S= -125961703
--------------------------------
Process exited after 2.17 seconds with return value 0
Press any key to continue . . .
```

# EXERCISE 3:

WRITE A PROGRAM THAT DISPLAYS THE PRODUCT TABLE FOR N VARYING FROM 1 TO 10



# EXERCISE 3-SOLUTION
# WRITE A PROGRAM THAT DISPLAYS THE PRODUCT TABLE FOR N VARYING FROM 1 TO 10

```
#include<iostream>
using namespace std;
int main()
{
int i,j;
for (i=1;i<=10;i++)
{
    for (j=1; j<=10; j++)
        cout<<i<<"*"<<j<<"="<<i*j<<"\t";
    cout<<endl;
}
return 0;
}
```

```
1*1=1   1*2=2   1*3=3   1*4=4   1*5=5   1*6=6   1*7=7   1*8=8   1*9=9   1*10=10
2*1=2   2*2=4   2*3=6   2*4=8   2*5=10  2*6=12  2*7=14  2*8=16  2*9=18  2*10=20
3*1=3   3*2=6   3*3=9   3*4=12  3*5=15  3*6=18  3*7=21  3*8=24  3*9=27  3*10=30
4*1=4   4*2=8   4*3=12  4*4=16  4*5=20  4*6=24  4*7=28  4*8=32  4*9=36  4*10=40
5*1=5   5*2=10  5*3=15  5*4=20  5*5=25  5*6=30  5*7=35  5*8=40  5*9=45  5*10=50
6*1=6   6*2=12  6*3=18  6*4=24  6*5=30  6*6=36  6*7=42  6*8=48  6*9=54  6*10=60
7*1=7   7*2=14  7*3=21  7*4=28  7*5=35  7*6=42  7*7=49  7*8=56  7*9=63  7*10=70
8*1=8   8*2=16  8*3=24  8*4=32  8*5=40  8*6=48  8*7=56  8*8=64  8*9=72  8*10=80
9*1=9   9*2=18  9*3=27  9*4=36  9*5=45  9*6=54  9*7=63  9*8=72  9*9=81  9*10=90
10*1=10 10*2=20 10*3=30 10*4=40 10*5=50 10*6=60 10*7=70 10*8=80 10*9=90 10*10=10
0

--------------------------------
Process exited after 0.06401 seconds with return value 0
Press any key to continue . . .
```

**EXERCISE 4**

WRITE A PROGRAM THAT READS A POSITIVE INTEGER N AND DISPLAYS:

1
22
333
4444
. . .
NNNN.. . .N

**EXERCISE 4**-SOLUTION

WRITE A PROGRAM THAT READS A POSITIVE INTEGER N AND DISPLAYS:

1

22

333

4444

. . .

NNNN.. . .N

```cpp
#include<iostream>
using namespace std;
int main()
{
    int i,j,N;
    do
    {
        cout<<"enter N>0 :";
        cin>>N;
    }while(N<=0);

    for(i=1;i<=N;i++)
    {
        for(j=1;j<=i;j++)
            cout<<i;
        cout<<"\n"; //cout<<endl;
    }
```

```
enter N>0 :9
1
22
333
4444
55555
666666
7777777
88888888
999999999

--------------------------------
Process exited after 1.806 seconds with return value 0
Press any key to continue . . .
```

```
enter N>0 :20
1
22
333
4444
55555
666666
7777777
88888888
999999999
10101010101010101010
11111111111111111111
12121212121212121212
13131313131313131313
14141414141414141414
15151515151515151515
16161616161616161616
17171717171717171717
18181818181818181818
19191919191919191919
20202020202020202020
```

# EXERCISE 5-1:

WRITE A PROGRAM THAT READS A POSITIVE INTEGER N AND DISPLAYS:

```
enter N>0 :10
*
 *
  *
   *
    *
     *
      *
       *
        *
         *
```

# EXERCISE 5-1: SOLUTION

## WRITE A PROGRAM THAT READS A POSITIVE INTEGER N AND DISPLAYS:

```
enter N>0 :10
*
 *
  *
   *
    *
     *
      *
       *
        *
         *
```

```
#include<iostream>
using namespace std;
int main()
{
    int i,j,N;
    do
    {
        cout<<"enter N>0 :";
        cin>>N;
    }while(N<=0);
    for(i=1;i<=N;i++)
    {
        for(j=1;j<=i-1;j++)
            cout<<" ";
        cout<<"*\n";
    }
return 0;
}
```

```
enter N>0 :10
*
 *
  *
   *
    *
     *
      *
       *
        *
         *

--------------------------------
Process exited after 1.576 seconds with return value 0
Press any key to continue . . .
```

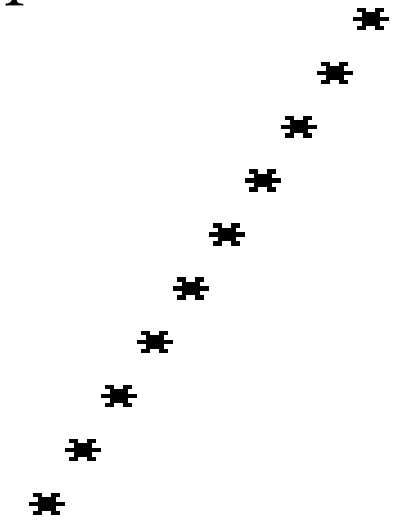
```
enter N>0 :8
*
 *
  *
   *
    *
     *
      *

--------------------------------
Process exited after 3.601 seconds with return value 0
Press any key to continue . . .
```

# EXERCISE 5-2:

WRITE A PROGRAM THAT READS A POSITIVE INTEGER N

Æntер N>0 :10YS:



# EXERCISE 5-2: SOLUTION 1

## WRITE A PROGRAM THAT READS A POSITIVE INTEGER N

```
enter N>0 :10
          *
         *
        *
       *
      *
     *
    *
   *
  *
 *
```

```
#include<iostream>
using namespace std;
int main()
{
    int i,j,N;
    do
    {
        cout<<"enter N>0 :";
        cin>>N;
    }while(N<=0);
    for(i=1;i<=N;i++)
    {
        for(j=1;j<=N-i;j++)
            cout<<" ";
        cout<<"*\n";
    }
return 0;
}
```

# EXERCISE 5-2: SOLUTION 2

## WRITE A PROGRAM THAT READS A POSITIVE INTEGER N

```
enter N>0 :10
*
*
*
*
*
*
*
*
*
*
```

```cpp
#include<iostream>
using namespace std;
int main()
{
    int i,j,N;
    do
    {
        cout<<"enter N>0 :";
        cin>>N;
    }while(N<=0);

    for(i=N;i>=1;i--)
    {
        for(j=1;j<=i-1;j++)
            cout<<" ";
        cout<<"*\n";
    }
return 0;
}
```

# EXERCISE 6

DISPLAY AN ISOSCELES TRIANGLE FORMED OF STARS OF N LINES (N IS PROVIDED ON THE KEYBOARD):

```
enter N>0 :10
---------*
--------***
-------*****
------*******
-----*********
----***********
---*************
--***************
-*****************
*******************
```

# EXERCISE 6: SOLUTION

## DISPLAY AN ISOSCELES TRIANGLE FORMED OF STARS OF N LINES (N IS PROVIDED ON THE KEYBOARD):

```
enter N>0 :10
---------*
--------***
-------*****
------*******
-----*********
----***********
---*************
--***************
-*****************
*******************
```

```cpp
#include<iostream>
using namespace std;
int main()
{
    int i,j,N;
    do
    {
        cout<<"enter N>0 :";
        cin>>N;
    }while(N<=0);
    for(i=1;i<=N;i++)
    {
        for(j=1;j<=N-i;j++)
            cout<<"-";
        for (j=1;j<=2*i-1;j++)
            cout<<"*";
        cout<<endl;
    }
return 0;
}
```

```
enter N>0 :10
---------*
--------***
-------*****
------*******
-----*********
----***********
---*************
--***************
-*****************
*******************

--------------------------------
Process exited after 2.527 seconds with return value 0
Press any key to continue . . .
```

```
enter N>0 :20
-------------------*
------------------***
-----------------*****
----------------*******
---------------*********
--------------***********
-------------*************
------------***************
-----------*****************
----------*******************
---------*********************
--------***********************
-------*************************
------***************************
-----*****************************
----*******************************
---*********************************
--***********************************
-*************************************
***************************************

--------------------------------
Process exited after 4.612 seconds with return value 0
```