Control Statements & Decision-Making In C

Course Title :- Structured Programming Language Sessional

Course Code :- CSE-122 [SECTION-B]

Level Term: 1-II-A(G1) & 1-II-B(G3,G4)

Khandaker Jannatul Ritu, Lecturer(CSE), BAIUST

Loop

What is Loop?

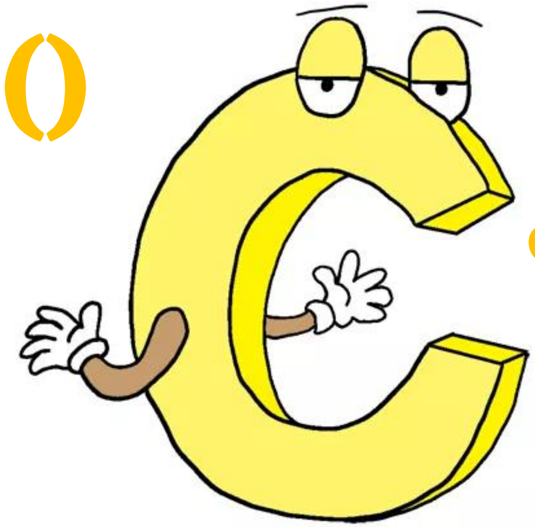
Loop

while()

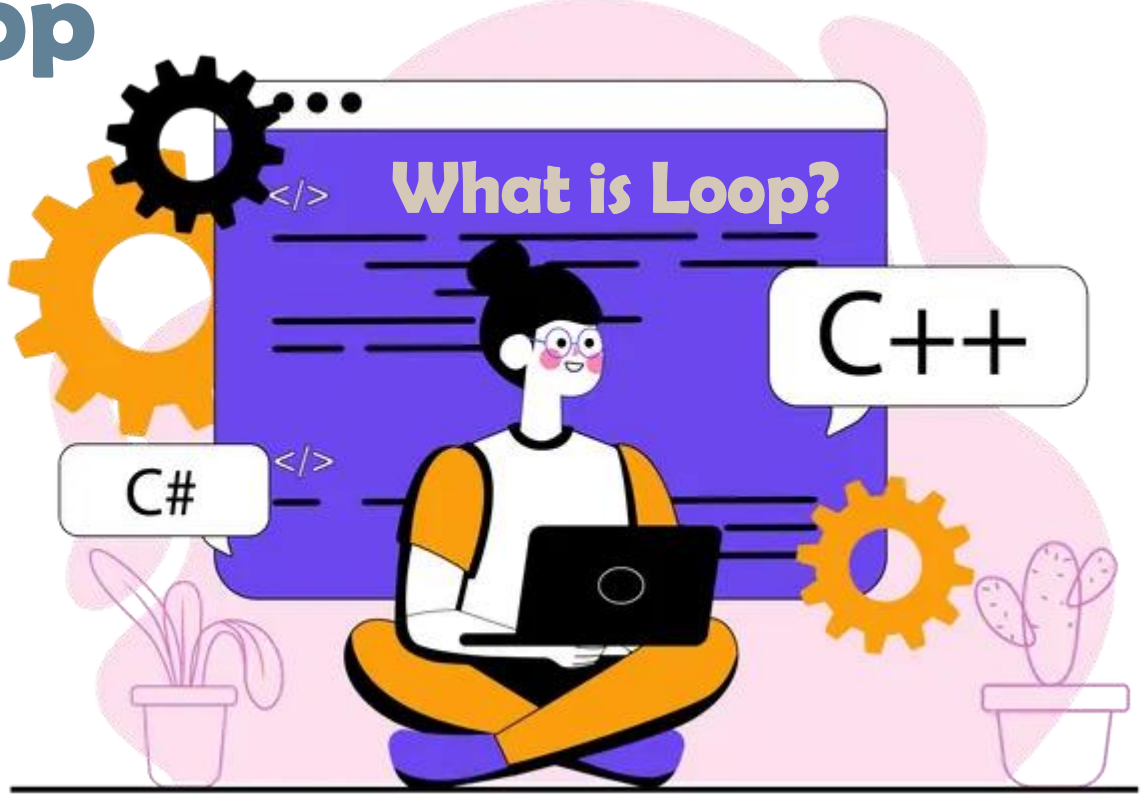
for()

Loop

Oh! I



do while()



What is Loop?

while()

Loop

Loop

do while()

Loop

Loop

Loop

for()

Why C – Loops?

Loops in programming are used to repeat a block of code until the specified condition is met. A loop statement allows programmers to execute a statement or group of statements multiple times without repetition of code.

```
#include <stdio.h>
int main()
{
      printf( "Hello World\n");
      printf( "Hello World\n");
      printf( "Hello World\n");
      printf( "Hello World\n");
      printf( "Hello World\n");
      printf( "Hello World\n");
      printf( "Hello World\n");
      printf( "Hello World\n");
      printf( "Hello World\n");
      printf( "Hello World\n");
}
```

Output

```
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
```

Loop

```
for(start ; condition ; increment/decrement)
for(i = 0; i <= 5; i++)
i = 0 ; 0 <= 5  → printf(“%d”, i);  i = 0+1=1 ;
        1 <= 5  → printf(“%d”, i);  i = 1+1=2 ;
        2 <= 5  → printf(“%d”, i);  i = 2+1=3 ;
        3 <= 5  → printf(“%d”, i);  i = 3+1=4 ;
        4 <= 5  → printf(“%d”, i);  i = 4+1=5 ;
        5 <= 5  → printf(“%d”, i);  i = 5+1=6 ;
        6 <= 5  → condition not true, so exit
```

❑ for Loop

Syntax:

```
for (initialize expression; test expression; update expression)
{
  // body of for loop
}
```

Example:

```
for(int i = 0; i < n; ++i)
{
   printf("Body of for loop which will execute till n");
}
```

```
#include <stdio.h>
int main(){
        int i;
        // for loop without curly braces
        for (i = 1; i <= 10; i++)
                printf("%d ", i);
        printf("\nThis statement executes after for loop end!!!!");
}
```

Output

```
1 2 3 4 5 6 7 8 9 10
This statement executes after for loop end!!!!
```

```
#include <stdio.h>
// Driver code
int main(){
        int i = 0;
        for (i = 1; i <= 10; i++){
                printf( "Hello World\n");
        }
}
```

Output

```
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
```

❑ While loop: Syntax:

```
initialization_expression;
while (test_expression)
{
   // body of the while loop
   update_expression;
}
```

```
#include <stdio.h>
int main(){
        int i = 2;
        while(i < 10){
                printf( "Hello World\n");
                i++;
        }
}
```

Output

```
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
```

❑ Traverse a while() loop

```
i=0;
while(i <= 5)
{
   printf("%d\n", i);
   i++;
}
```

```
i = 0   while(0 <= 5)   printf("%d\n", i);   i = i+1 = 0+1=1
             while(1 <= 5)   printf("%d\n", i);   i = i+1 = 1+1=2
             while(2 <= 5)   printf("%d\n", i);   i = i+1 = 2+1=3
             while(3 <= 5)   printf("%d\n", i);   i = i+1 = 3+1=4
             while(4 <= 5)   printf("%d\n", i);   i = i+1 = 4+1=5
             while(5 <= 5)   printf("%d\n", i);   i = i+1 = 5+1=6
             while(6 <= 5)  → condition overflow, so exit
```

❑ do-while Loop:

Syntax:

```
initialization_expression;
do
{
   // body of do-while loop
   update_expression;

} while (test_expression);
```

Traverse a do while() loop

```
Example-1:
i = 0;
   do{
      printf("%d\n", i);
      i++;
   }while(i <= 5);
```

```
i = 0    printf("%d\n", i);   i = i + 1 = 0 + 1 = 1     while (1 <= 5)
         printf("%d\n", i);   i = i + 1 = 1 + 1 = 2     while (2 <= 5)
         printf("%d\n", i);   i = i + 1 = 2 + 1 = 3     while (3 <= 5)
         printf("%d\n", i);   i = i + 1 = 3 + 1 = 4     while (4 <= 5)
         printf("%d\n", i);   i = i + 1 = 4 + 1 = 5     while (5 <= 5)
         printf("%d\n", i);   i = i + 1 = 5 + 1 = 6     while (6 <= 5)
         → now condition overflow, so exit the loop
```

Ex-2:

```
#include <stdio.h>
int main()
{
        // Initialization expression
        int i = 2;
        do{
                // loop body
                printf( "Hello World\n");

                // Update expression
                i++;

                // Test expression
        } while (i < 1);
}
```

Output

```
Hello World
```

Ex-3:

```
#include <stdio.h>
int main(){
        int i = 0;
        // do while loop
        do {
                printf("Geeks\n");
                i++;
        } while (i < 3);
}
```

Output

```
Geeks
Geeks
Geeks
```

❑ Infinite Loop

```
#include <stdio.h>
int main (){
    int i;
    for ( ; ; ){
        printf("This loop will run forever.\n");
    }
}
```

Output

```
This loop will run forever.
This loop will run forever.
This loop will run forever.
…
```

```
#include <stdio.h>
int main(){
    do{
        printf("This loop will run forever.\n");
    } while (1);
}
```

Output

```
This loop will run forever.
This loop will run forever.
This loop will run forever.
…
```

```
#include <stdio.h>
int main() {
    while (1)
        printf("This loop will run forever.\n");
}
```

Output

```
This loop will run forever.
This loop will run forever.
This loop will run forever.
…
```

❑Nested loops in C

```
while()
{
   for()
   {

   }
}
```

```
for()
{
   for()
   {

   }
}
```

```
do{
   do{

   }while();
}while();
```

```
for()
{
   while()
   {

   }
}
```

```
while()
{
   while()
   {

   }
}
```

```
do{
   for(){
      while(){

      }
   }
}while();
```

Nested for loop:

```
#include <stdio.h>
int main() {
        int i, j;
        for (i = 1; i <= 2; ++i)
        {
                printf("Outer: %d\n", i);
                for (j = 1; j <= 3; ++j)
                {
                     printf(" Inner: %d\n", j);
                }
        }
}
```

Output:

```
Outer: 1
 Inner: 1
 Inner: 2
 Inner: 3
Outer: 2
 Inner: 1
 Inner: 2
 Inner: 3
```

❑ **for** Versus **while**

```
#include <stdio.h>
int main(){
    int sum=0, i;
    for(i=1;i<=5;i++){
        sum=sum+i;
    }
    printf("SUM = %d" , sum);
}
```

Output

```
SUM = 15
```

```
#include<stdio.h>
int main(){
    int no=1, sum=0;
    while(no<=5){
        sum=sum+no;
        no++;
    }
    printf("SUM = %d" , sum);
}
```

Output

```
SUM = 15
```

Part-1 : loop Practise problems

1. a. Traverse a for() loop → print numbers 1 to n & print numbers n to 1
   b. Traverse a while() loop → print numbers 1 to n & print numbers n to 1
   c. Traverse a do while() loop → print numbers 1 to n & print numbers n to 1

2. Print all even/odd numbers for 1 to n
3. Sum of all numbers 1 to n

4. a. C Program to Display Characters from a to z Using Loop
   b. C Program to Display Characters from A to Z Using Loop

5. Multiplication Table Up to 10
6. C Program to Find Factorial of a Number
7. C Program to Check Whether a Number is Prime or Not

1 a. Traverse a for() loop → print numbers 1 to n & print numbers n to 1

///traverse a for loop: 0 to n

```
#include<stdio.h>
int main()
{
   int n,i;
   printf("Enter n : ");
   scanf("%d", &n);

   printf("-----print 0 to n----\n");
   /// print 0 to n
   for(i = 0 ; i<=n; i++)
   {
      printf("hello %d\n", i);
   }
}
```

///traverse a for loop: n to 0

```
#include<stdio.h>
int main()
{
   int n,i;
   printf("Enter n : ");
   scanf("%d", &n);

printf("-----print n to 0-----\n");
   for(i = n ; i>= 0 ; i--)
   {
      printf("hello %d\n", i);
   }

}
```

1 b. Traverse a while() loop → print numbers 1 to n & print numbers n to 1

```
///traverse a while loop: 0 to n
#include<stdio.h>
int main(){
   int n,i;
   printf("Enter n : ");
   scanf("%d", &n);
   printf("-----print 0 to n----\n");
   /// print 0 to n
   i=0;
   while(i <= n)
   {
      printf("%d\n", i);
      i++;
   }
}
```

```
///traverse a while loop: n to 0
#include<stdio.h>
int main(){
   int n,i;
   printf("Enter n : ");
   scanf("%d", &n);
   /// print n to 0
   printf("-----print n to 0-----\n");
   i = n;
   while( i >= 0)
   {
      printf("%d\n", i);
      i--;
   }
}
```

1 c. Traverse a do while() loop → print numbers 1 to n & print numbers n to 1

///traverse a do while() loop: 0 to n

```
#include<stdio.h>
int main(){
    int n,i;
    printf("Enter n : ");
    scanf("%d", &n);

    printf("-----print 0 to n----\n");
    i = 0;
    do{
        printf("%d\n", i);
        i++;
    }while(i <= n);
}
```

///traverse a do while() loop: n to 0

```
#include<stdio.h>
int main(){
    int n,i;
    printf("Enter n : ");
    scanf("%d", &n);

    printf("-----print n to 0-----\n");
    i = n;
    do{
        printf("%d\n", i);
        i--;
    }while(i>=0);
}
```

2. Print all even/odd numbers for 1 to n

```
#include<stdio.h>
int main(){
   int number,i;

   printf("Enter a number: ");
   scanf("%d", &number);

   for(i = 0; i <= number; i++){
      if( i % 2 == 0){
         printf("%d : Even\n", i);
      }
      else{
         printf("%d : Odd\n", i);
      }
   }
}
```

3. sum of all numbers for 1 to n

```
#include<stdio.h>
int main()
{
   int number,i, sum;
   printf("Enter a number: ");
   scanf("%d", &number);

   sum = 0;
   for(i = 1; i <= number; i++)
   {
      sum = sum + i;
   }
   printf("sum [1 to n] : %d\n", sum);
}
```

4 a. Display Characters from a to z Using Loop

```
#include<stdio.h>
int main()
{
    char i;
    printf("---Print small letters [a to z] ---\n");
    for(i = 'a'; i <= 'z'; i++)
    {
        printf("%c ", i);
    }
}
```

4 b. Display Characters from A to Z Using Loop

```
#include<stdio.h>
int main()
{
    char i;
    printf("\n---Print capital letters [A to Z]---\n");
    for(i = 'A'; i <= 'Z'; i++)
    {
        printf("%c ", i);
    }
}
```

5. Multiplication Table Up to 10

```
#include<stdio.h>
int main(){
        int n,i;
        printf("Enter a number : ");
        scanf("%d", &n);
        for(i = 1 ; i <= 10 ; i++){
          printf("%d * %d = %d\n",i, n, (i*n));
        }
}
```

```
Enter a number : 10
1 * 10 = 10
2 * 10 = 20
3 * 10 = 30
4 * 10 = 40
5 * 10 = 50
6 * 10 = 60
7 * 10 = 70
8 * 10 = 80
9 * 10 = 90
10 * 10 = 100
```

6. C Program to Find Factorial of a Number

```
#include<stdio.h>
int main(){
  int factorial, result, i;
  printf("Enter a number : ");
  scanf("%d", &factorial);
  result = 1;
  for(i = 2; i<=factorial; i++){
    result = result * i ;
  }
  printf("Factorial : %d\n", result);
}
```

output:

```
Enter a number: 5
Factorial : 120
```

Analysis:

5! = 1 x 2 x 3 x 4 x 5

1 * 2 = 2

2 * 3 = 6

6 * 4 = 24

24 * 5 = 120

7. C Program to check whether a number is prime or not.

```
#include<stdio.h>
#include<stdbool.h>
int main(){
   int n;
   scanf("%d", &n);
  if(n<=1){
      printf("Not prime");
   }
   else if(n == 2){
      printf("Prime!!");
   }
   else if(n!=2 && n % 2 == 0){
      printf("Not prime");
   }
```

```
else{
      bool check = true;
      for(int i = 2 ; i<= n - 1 ; i++){
         if(n % i == 0){
            printf("Not prime");
            check = false;
            break;
         }
      }
    if(check == true){
         printf("Prime!!");
      }
   }
   printf("\n\n\n");
}
```

Part-2 : Pattern printing - nested loop examples

Pattern Type - 1				
N = 5 1 0 0 1 1 1 0 0 0 0 1 1 1 1 1	N = 5 1 1 0 1 0 1 1 0 1 0 1 0 1 0 1	N = 5 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5	N = 5 1 1 2 1 2 3 1 2 3 4 1 2 3 4 5	N = 5 * * * * * * * * * * * * * * *
N = 5 a b b c c c d d d d e e e e e	N = 5 a a b a b c a b c d a b c d e	N = 5 A B B C C C D D D D E E E E E	N = 5 A A B A B C A B C D A B C D E	N = 5 # # # # # # # # # # # # # # #

```c
#include<stdio.h>
int main(){
   int n, row, col;
   printf("Enter n : ");
   scanf("%d", &n);

   for(row = 1; row<= n; row++)
   {
      for(col = 1; col <= row; col++)
      {
         printf("* ");
      }
      printf("\n");
   }
}
```

```c
/*
printf("%d ", row % 2);
printf("%d ", col % 2);
printf("%d ", row);
printf("%d ", col);
printf("* ");
printf("%c ", 96+row);
printf("%c ", 96+col);
printf("%c ", 64+row);
printf("%c ", 64+col);
printf("# ");
*/
```

Output:

```
Enter n : 7
*
* *
* * *
* * * *
* * * * *
* * * * * *
* * * * * * *
```

Pattern Type - 2

N = 5 1 1 1 1 1 0 0 0 0 1 1 1 0 0 1	N = 5 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1	N = 5 5 5 5 5 5 4 4 4 4 3 3 3 2 2 1	N = 5 1 2 3 4 5 1 2 3 4 1 2 3 1 2 1	N = 5 * * * * * * * * * * * * * * *
N = 5 e e e e e d d d d c c c b b a	N = 5 a b c d e a b c d a b c a b a	N = 5 E E E E E D D D D C C C B B A	N = 5 A B C D E A B C D A B C A B A	N = 5 # # # # # # # # # # # # # # #

```
#include<stdio.h>
int main()
{
    int n, row, col;
    printf("Enter n : ");
    scanf("%d", &n);

    for(row = n; row >= 1; row--)
    {
        for(col = 1; col<= row; col++)
        {
            printf("%c ", 64+col);
        }
        printf("\n");
    }
}
```

```
/*
printf("%d ", row % 2);
printf("%d ", col % 2);
printf("%d ", row);
printf("%d ", col);
printf("* ");
printf("%c ", 96+row);
printf("%c ", 96+col);
printf("%c ", 64+row);
printf("%c ", 64+col);
printf("# ");
*/
```

Output:

```
A B C D E
A B C D
A B C
A B
A
```

Pattern Type - 3

N = 5 1 0 0 1 1 1 0 0 0 0 1 1 1 1 1 0 0 0 0 1 1 1 0 0 1	N = 5 1 1 0 1 0 1 1 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1	N = 5 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5 4 4 4 4 3 3 3 2 2 1	N = 5 1 1 2 1 2 3 1 2 3 4 1 2 3 4 5 1 2 3 4 1 2 3 1 2 1	N = 5 *
N = 5 a b b c c c d d d d e e e e e d d d d c c c b b a	N = 5 a a b a b c a b c d a b c d e a b c d a b c a b a	N = 5 A B B C C C D D D D E E E E E D D D D C C C B B A	N = 5 A A B A B C A B C D A B C D E A B C D A B C A B A	N = 5 #

```c
#include<stdio.h>
int main(){
    int n, row, col;
    printf("Enter n : ");
    scanf("%d", &n);
    for(row = 1; row<= n; row++){
        for(col = 1; col <= row; col++) {
            printf("# ");
        }
        printf("\n");
    }
    for(row = n-1; row >= 1; row--){
        for(col = 1; col<= row; col++){
            printf("# ");
        }
        printf("\n");
    }
}
```

```c
/*
printf("%d ", row % 2);
printf("%d ", col % 2);
printf("%d ", row);
printf("%d ", col);
printf("* ");
printf("%c ", 96+row);
printf("%c ", 96+col);
printf("%c ", 64+row);
printf("%c ", 64+col);
printf("# ");
*/
```

Output:

```
#
# #
# # #
# # # #
# # # # #
# # # #
# # #
# #
#
```

Pattern Type - 4				
N = 5 1 00 111 0000 11111	N = 5 1 10 101 1010 10101	N = 5 1 22 333 4444 55555	N = 5 1 12 123 1234 12345	N = 5 * ** *** **** *****
N = 5 a bb ccc dddd eeeee	N = 5 a ab abc abcd abcde	N = 5 A BB CCC DDDD EEEEE	N = 5 A AB ABC ABCD ABCDE	N = 5 # ## ### #### #####

```
#include<stdio.h>
int main(){
   int n, row, col;
   printf("Enter n : ");
   scanf("%d", &n);

   for(row = 1; row<= n; row++){
      for(col = 1; col <= n-row; col++){
         printf(" ");
      }
      for(col = 1; col<= row; col++){
         printf("#");
      }
      printf("\n");
   }
}
```

```
/*
printf("%d", row % 2);
printf("%d", col % 2);
printf("%d", row);
printf("%d", col);
printf("*");
printf("%c", 96+row);
printf("%c", 96+col);
printf("%c", 64+row);
printf("%c", 64+col);
printf("#");
*/
```

Output:

```
    #
   ##
  ###
 ####
#####
```

Pattern Type - 5				

N = 5
11111
 0000
  111
   00
    1
	N = 5
	10101
	 1010
	  101
	   10
	    1
		N = 5
		55555
		 4444
		  333
		   22
		    1
			N = 5
			12345
			 1234
			  123
			   12
			    1
				N = 5
				*****
				 ****
				  ***
				   **
				    *

N = 5
eeeee
 dddd
  ccc
   bb
    a
	N = 5
	abcde
	 abcd
	  abc
	   ab
	    a
		N = 5
		EEEEE
		 DDDD
		  CCC
		   BB
		    A
			N = 5
			ABCDE
			 ABCD
			  ABC
			   AB
			    A
				N = 5
				#####
				 ####
				  ###
				   ##
				    #

```
#include<stdio.h>
int main(){
    int n, row, col;
    printf("Enter n : ");
    scanf("%d", &n);

    for(row = n; row >= 1; row--){
        for(col = 1; col <= n-row; col++) {
            printf(" ");
        }
        for(col = 1; col<= row; col++) {
            printf("%c", 64+col);
        }
        printf("\n");
    }
}
```

```
/*
printf("%d", row % 2);
printf("%d", col % 2);
printf("%d", row);
printf("%d", col);
printf("*");
printf("%c", 96+row);
printf("%c", 96+col);
printf("%c", 64+row);
printf("%c", 64+col);
printf("#");
*/
```

Output:

```
ABCDE
 ABCD
  ABC
   AB
    A
```

Pattern Type - 6

```
N = 5
    1
   0 0
  1 1 1
 0 0 0 0
1 1 1 1 1
 0 0 0 0
  1 1 1
   0 0
    1
```

```
N = 5
    1
   1 0
  1 0 1
 1 0 1 0
1 0 1 0 1
 1 0 1 0
  1 0 1
   1 0
    1
```

```
N = 5
    1
   2  2
  3 3 3
 4 4 4 4
5 5 5 5 5
 4 4 4 4
  3 3 3
   2 2
    1
```

```
N = 5
    1
   1 2
  1 2 3
 1 2 3 4
1 2 3 4 5
 1 2 3 4
  1 2 3
   1  2
    1
```

```
N = 5
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

```
N = 5
    a
   b b
  c c c
 d d d d
e e e e e
 d d d d
  c c c
   b b
    a
```

```
N = 5
    a
   a b
  a b c
 a b c d
a b c d e
 a b c d
  a b c
   a b
    a
```

```
N = 5
    A
   B B
  C C C
 D D D D
E E E E E
 D D D D
  C C C
   B B
    A
```

```
N = 5
    A
   A B
  A B C
 A B C D
A B C D E
 A B C D
  A B C
   A B
    A
```

```
N = 5
        #
      # #
    # # #
  # # # #
# # # # #
  # # # #
    # # #
      # #
        #
```

```c
#include<stdio.h>
int main(){
   int n, row, col;
   printf("Enter n : ");
   scanf("%d", &n);
   for(row = 1; row<= n; row++){
      for(col = 1; col <= n-row; col++){
         printf(" ");
      }
      for(col = 1; col<= row; col++){
         printf("%d", row % 2);
      }
      printf("\n");
   }
   for(row = n-1; row >= 1; row--){
      for(col = 1; col <= n-row; col++){
         printf(" ");
      }
      for(col = 1; col<= row; col++){
         printf("%d", row % 2);
      }
      printf("\n");
   }
}
```

```c
/*
printf("%d", row % 2);
printf("%d", col % 2);
printf("%d", row);
printf("%d", col);
printf("*");
printf("%c", 96+row);
printf("%c", 96+col);
printf("%c", 64+row);
printf("%c", 64+col);
printf("#");
*/
```

Output:

```
    1
   2  2
  3 3 3
 4 4 4 4
5 5 5 5 5
 4 4 4 4
  3 3 3
   2 2
    1
```

Pattern Type - 7				
N = 5 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1	N = 5 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1	N = 5 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4 5 5 5 5 5	N = 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5	N = 5 *
N = 5 a a a a a b b b b b c c c c c d d d d d e e e e e	N = 5 a b c d e a b c d e a b c d e a b c d e a b c d e	N = 5 A A A A A B B B B B C C C C C D D D D D E E E E E	N = 5 A B C D E A B C D E A B C D E A B C D E A B C D E	N = 5 #

```c
#include<stdio.h>
int main(){
   int n, row, col;
   printf("Enter n : ");
   scanf("%d", &n);

   for(row = 1; row<= n; row++) {
      for(col = 1; col<= n; col++){
         printf("# ");
      }
      printf("\n");
   }
}
```

```c
/*
printf("%d ", row % 2);
printf("%d ", col % 2);
printf("%d ", row);
printf("%d ", col);
printf("* ");
printf("%c ", 96+row);
printf("%c ", 96+col);
printf("%c ", 64+row);
printf("%c ", 64+col);
printf("# ");
*/
```

```
# # # # #
# # # # #
# # # # #
# # # # #
# # # # #
```

Pattern Type - 8

N = 5 1 0 0 1 1 1 0 0 0 0 1 1 1 1 1	N = 5 1 1 0 1 0 1 1 0 1 0 1 0 1 0 1	N = 5 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5	N = 5 1 1 2 1 2 3 1 2 3 4 1 2 3 4 5	N = 5 * * * * * * * * * * * * * * *
N = 5 a b b c c c d d d d e e e e e	N = 5 a a b a b c a b c d a b c d e	N = 5 A B B C C C D D D D E E E E E	N = 5 A A B A B C A B C D A B C D E	N = 5 # # # # # # # # # # # # # # #

```
#include<stdio.h>
int main(){
   int n, row, col;
   printf("Enter n : ");
   scanf("%d", &n);

   for(row = 1; row<= n; row++) {
      for(col = 1; col <= n-row; col++){
         printf(" ");
      }
      for(col = 1; col<= row; col++){
         printf("# ");
      }
      printf("\n");
   }

}
```

```
/*
printf("%d ", row % 2);
printf("%d ", col % 2);
printf("%d ", row);
printf("%d ", col);
printf("* ");
printf("%c ", 96+row);
printf("%c ", 96+col);
printf("%c ", 64+row);
printf("%c ", 64+col);
printf("# ");
*/
```

```
    *
   * *
  * * *
 * * * *
* * * * *
```

Pattern Type - 9

N = 5 1 1 1 1 1 0 0 0 0 1 1 1 0 0 1	N = 5 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1	N = 5 5 5 5 5 5 4 4 4 4 3 3 3 2 2 1	N = 5 1 2 3 4 5 1 2 3 4 1 2 3 1 2 1	N = 5 * * * * * * * * * * * * * * *
N = 5 e e e e e d d d d c c c b b a	N = 5 a b c d e a b c d a b c a b a	N = 5 E E E E E D D D D C C C B B A	N = 5 A B C D E A B C D A B C A B A	N = 5 # # # # # # # # # # # # # # #

```c
#include<stdio.h>
int main(){
   int n, row, col;
   printf("Enter n : ");
   scanf("%d", &n);

   for(row = n; row >= 1; row--){
      for(col = 1; col <= n-row; col++) {
         printf(" ");
      }
      for(col = 1; col<= row; col++) {
         printf("%c ", 64+col);
      }
      printf("\n");
   }
}
```

```c
/*
printf("%d ", row % 2);
printf("%d ", col % 2);
printf("%d ", row);
printf("%d ", col);
printf("* ");
printf("%c ", 96+row);
printf("%c ", 96+col);
printf("%c ", 64+row);
printf("%c ", 64+col);
printf("# ");
*/
```

```
N = 5
e e e e e
 d d d d
  c c c
   b b
    a
```

```
N = 5
a b c d e
 a b c d
  a b c
   a b
    a
```

Pattern Type - 10

N = 5 1 0 0 1 1 1 0 0 0 0 1 1 1 1 1 0 0 0 0 1 1 1 0 0 1	N = 5 1 1 0 1 0 1 1 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1	N = 5 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5 4 4 4 4 3 3 3 2 2 1	N = 5 1 1 2 1 2 3 1 2 3 4 1 2 3 4 5 1 2 3 4 1 2 3 1 2 1	N = 5 *
N = 5 a b b c c c d d d d e e e e e d d d d c c c b b a	N = 5 a a b a b c a b c d a b c d e a b c d a b c a b a	N = 5 A B B C C C D D D D E E E E E D D D D C C C B B A	N = 5 A A B A B C A B C D A B C D E A B C D A B C A B A	N = 5 #

```c
#include<stdio.h>
int main(){
   int n, row, col;
   printf("Enter n : ");
   scanf("%d", &n);
   for(row = 1; row<= n; row++){
      for(col = 1; col <= n-row; col++){
         printf(" ");
      }
      for(col = 1; col<= row; col++){
         printf("%c ", 64+col);
      }
      printf("\n");
   }
   for(row = n-1; row >= 1; row--){
      for(col = 1; col <= n-row; col++){
         printf(" ");
      }
      for(col = 1; col<= row; col++){
         printf("%c ", 64+col);
      }
      printf("\n");
   }
}
```

```c
/*
printf("%d ", row % 2);
printf("%d ", col % 2);
printf("%d ", row);
printf("%d ", col);
printf("* ");
printf("%c ", 96+row);
printf("%c ", 96+col);
printf("%c ", 64+row);
printf("%c ", 64+col);
printf("# ");
*/
```

```
N = 5
    1
   1 0
  1 0 1
 1 0 1 0
1 0 1 0 1
 1 0 1 0
  1 0 1
   1 0
    1
```

```
N = 5
    1
   2 2
  3 3 3
 4 4 4 4
5 5 5 5 5
 4 4 4 4
  3 3 3
   2 2
    1
```

Floyd's triangle

	C=1	C=2	C=3	C=4	C=5
R=1	1				
R=2	2	3			
R=3	4	5	6		
R=4	7	8	9	10	
R=5	11	12	13	14	15

```
#include<stdio.h>
int main(){
    int n, row, col;
    printf("Enter n : ");
    scanf("%d", &n);

    int count = 1;
    for(row = 1; row<= n; row++){
        for(col = 1; col<= row; col++){
            printf("%d ", count);
            count++;
        }
        printf("\n");
    }
}
```

❑ Jump Statements in C

A) break

B) continue

C) goto

Break in C switch case

Syntax of break in switch case

```
switch(expression)
{
case value1:
   statement_1;
   break;

case value2:
   statement_2;
   break;
.....
.....

case value_n:
   statement_n;
   break;

default:
   default statement;
}
```

// C Program to demonstrate infinite loop without using break statement

```
#include <stdio.h>
int main(){
        int i = 0;
        // while loop which will always be true
        while (1) {
                printf("%d ", i);
                i++;
                if (i == 5) {
                        break;
                }
        }
}
Output:  0 1 2 3 4
```

B) continue

```
#include <stdio.h>
int main(){
        int i = 0;
        while (i < 8) {
                i++;
                if (i == 4) {
                        continue;
                }
                printf("%d ", i);
        }
}
Output
1 2 3 5 6 7 8
```

```
#include <stdio.h>
int main() {
        // loop from 1 to 10
        for (int i = 1; i <= 10; i++) {
                if (i == 6)
                        continue;
                else
                        printf("%d ", i);
        }
}
Output
1 2 3 4 5 7 8 9 10
```

<u>Example:</u> C Program to demonstrate the difference between the working of break and continue statement in C.

```
#include <stdio.h>
int main(){
        printf("The loop with break produces output as: \n");
        for (int i = 1; i <= 7; i++) {
                if (i == 3)
                        break;
                else
                        printf("%d ", i);
        }
        printf("\nThe loop with continue produces output as: \n");
        for (int i = 1; i <= 7; i++) {
                if (i == 3)
                        continue;
                printf("%d ", i);
        }
}
```

```
printf("\nbreak in while loop\n");
  int i = 1;
  while (i < 20) {
    if (i == 3)
      break;
    else
      printf("%d ", i);
    i++;
  }
```

```
#include <stdio.h>
int main(){
  for (int i = 1; i <= 6; ++i) {
    for (int j = 1; j <= i; ++j) {
      if (i <= 4) {
        printf("%d ", j);
      }
      else {
      break;
      }
    }
    printf("\n");
  }
}
```

```
#include <stdio.h>
int main() {
  int i;
  double number, sum = 0.0;
  for (i = 1; i <= 10; ++i) {
    printf("Enter n%d: ", i);
    scanf("%lf", &number);
    if (number < 0.0) {
      break;
    }
    sum += number;
   }
  printf("Sum = %.2lf", sum);
}
```

Output:

```
Enter n1: 2.4
Enter n2: 4.5
Enter n3: 3.4
Enter n4: -3
Sum = 10.30
```

Online Judge

BeeCrowd

https://judge.beecrowd.com/en/problems/view/1095
https://judge.beecrowd.com/en/problems/view/1096
https://judge.beecrowd.com/en/problems/view/1097
https://judge.beecrowd.com/en/problems/view/1098
https://judge.beecrowd.com/en/problems/view/1074
https://judge.beecrowd.com/en/problems/view/1078
https://judge.beecrowd.com/en/problems/view/1165

CodeForces:

https://codeforces.com/group/MWSDmqGsZm/contest/219432/problem/A
https://codeforces.com/group/MWSDmqGsZm/contest/219432/problem/B
https://codeforces.com/group/MWSDmqGsZm/contest/219432/problem/C
https://codeforces.com/group/MWSDmqGsZm/contest/219432/problem/D
https://codeforces.com/group/MWSDmqGsZm/contest/219432/problem/E
https://codeforces.com/group/MWSDmqGsZm/contest/219432/problem/F
https://codeforces.com/group/MWSDmqGsZm/contest/219432/problem/G
https://codeforces.com/group/MWSDmqGsZm/contest/219432/problem/H
https://codeforces.com/group/MWSDmqGsZm/contest/219432/problem/I
https://codeforces.com/group/MWSDmqGsZm/contest/219432/problem/J
https://codeforces.com/group/MWSDmqGsZm/contest/219432/problem/O
https://codeforces.com/group/MWSDmqGsZm/contest/219432/problem/P
https://codeforces.com/group/MWSDmqGsZm/contest/219432/problem/T
https://codeforces.com/group/MWSDmqGsZm/contest/219432/problem/W

Solve this exercise

```
int i,j,k,p,q=0;
  for(i=1; i<=n; ++i){
     p=0;
     for(j=n; j>1; j=j/2){
        ++p;
     }
     for(k=1; k<p; k=k*2){
        ++q;
     }
  }
  printf("%d\n", q);
```

```
int count = 0;
  for(i=1; i<=3;i++){
     for(j=1;j<=3;j+=i){
        for(k=j; k<=3; k+=i){
           printf("%d %d %d\n",i,j,k);
           count++;
        }
     }
  }
  printf("Count = %d", count);
```

```
int i,j,k=0;
  j=2 * 3 / 4 + 2.0 / 5 + 8 / 5;
  k -=j;
  for(i=0; i<5; i++){
     switch(i+k){
     case 1:
     case 2: printf("\n %d", i+k);
     case 3: printf("\n %d", i+k);
     default: printf("\n %d", i+k);
     }
  }
```

```
float sum=0.0, j=1.0, i=2.0;
  while(i/j > 0.0625){
     j=j+j;
     sum = sum+ i/j;
     printf("%f\n", sum);
  }
```

```
  while(r >= y){
     r = r - y;
     q = q + 1;
  }
  printf("%d %d", r, q);
```

```
for(i=1; i<=n; i++){
     for(j=1; j<n; j+=i){
        printf("%d %d",i,j);
     }
  }
```

```
int res=1, a=10, b=34;
  while(b != 0){
     if(b %2 ==0){
        a = a*a;
        b = b/2;
     }
     else{
        res = res*a;
        b=b-1;
     }
  }
```

```
main() {
   int x, y, m, n;
   scanf ("%d %d", &x, &y);

   m = x; n = y;
   while (m != n) {
     if(m>n)
        m = m - n;
     else
        n = n - m;
   }
   printf("%d", n);
}
```

```
int x=3, y=7 ;
for (int i=0; i<y; i++)
{
         x=x+x+y;
}
printf(“%d”, x);
```

```
int a = 6;
 int b = 0;
 while(a < 10)
 {
          a = a / 12 + 1;
          a += b;
 }
 printf(”%d”, a);
```

```
int i, j = 0, sum = 0;
for (i = n; i > 1; i = i/2) j++;
for ( ; j > 1; j = j/2) sum++;

printf(“%d”, sum);
```

```
int i,j=5;
i=1;
while(i+j<25)
{       if(j+i%4==11)break;
        if(i%2==0){
                j /=2;
                i +=3;
                continue;
        }
        i--;
        j *=3;
}
printf("%d %d\n",i,j);
```

```
x=1,y=5;
for(i=1;i<5;i +=2)
{
        x +=i;
        y *=(i++);
}
printf(“x = %d y = %d\n”,x,y);
```

```
x=3,y=3;
if(x<y)
        x +=2;
if(x>y)
        y +=2;
printf(“x = %d y = %d\n”,x,y);
```

```
x=1,y=2;
while(x+y<12)
{
        x++;
        y *=2;
}
printf(“x = %d y = %d\n”,x,y);
```

```
 x=1,y=2;
while(x+y<12)
{
        x++;
        y++;
}
printf(“x = %d y = %d\n”,x,y);
```