Programming Using C

T7-Jump Statements, Nested Loops

Prepared By: REC Faculty 4.0 Team

TOPICS

- ✔ JUMP STATEMENTS
- ✔ BREAK
- ✔ CONTINUE
- ✔ GOTO
- ✔ EXIT
- ✔ NESTED LOOP STATEMENT SYNTAX & EXAMPLES
- ✔ MCQ

JUMP STATEMENTS

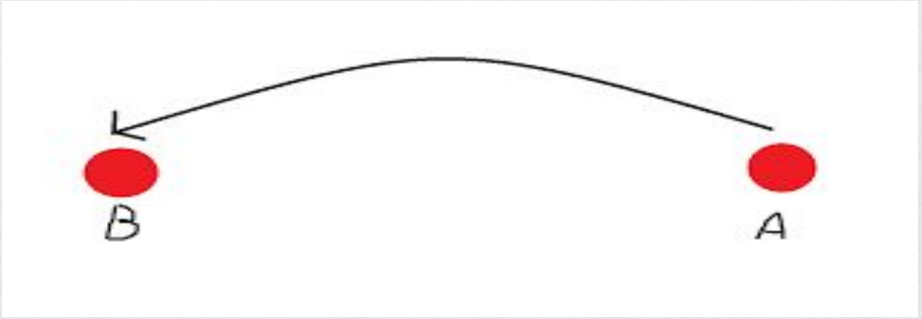
- C programming language allows jumping from one statement to another.

- It also supports break, continue, return and go to jump statements.

JUMP STATEMENTS

Jump statements?

Transfer control from one point to another point within the program

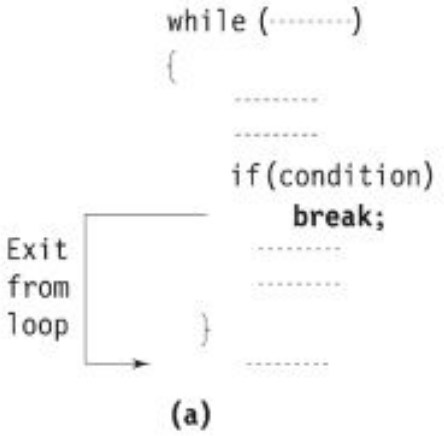


break

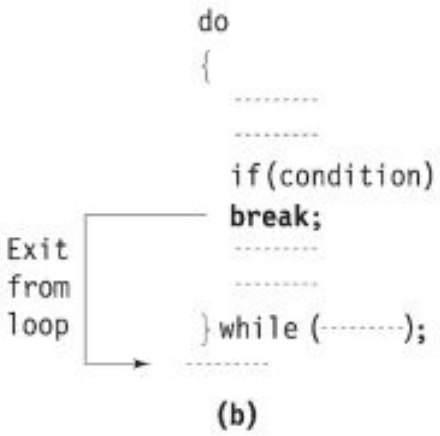
- The keyword break allows the programmer to terminate the loop.
- The break skips from the loop or block in which it is defined.
- The control then automatically goes to the first statement after the loop or block.
- The break can be associated with all conditional statements.

break

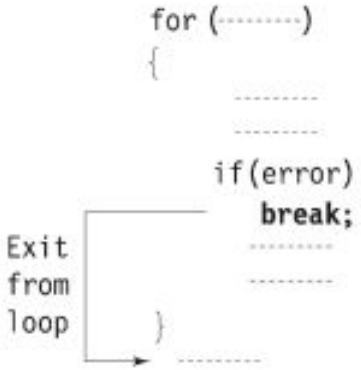
- We can also use break statements in the nested loops.
- If we use break statement in the innermost loop, then the control of the program is terminated only from the innermost loop.
- *break statement can be used in an if statement only when the if statement is written in a loop.*
- Syntax: **break;**


while (.....)
{
.....
.....
if(condition)
break;
.....
.....
}
.....
Exit
from
loop

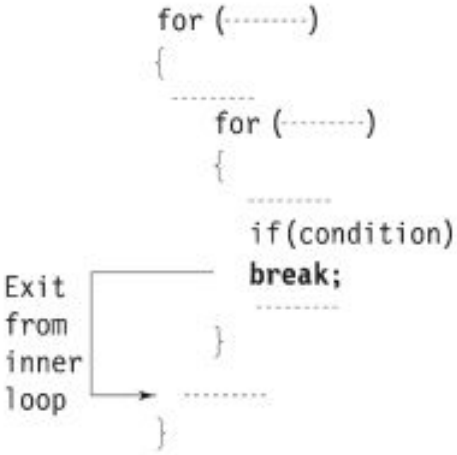
(a)


do
{
.....
.....
if(condition)
break;
.....
.....
} while (.....);
.....
Exit
from
loop

(b)


for (.....)
{
.....
.....
if(error)
break;
.....
.....
}
.....
Exit
from
loop

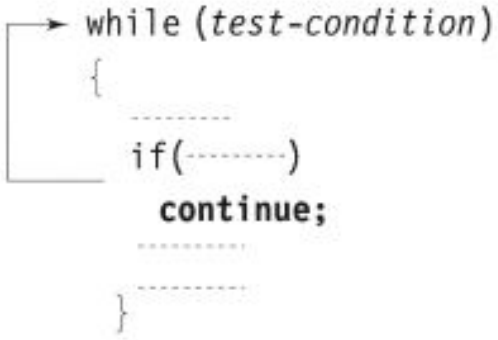
(c)


for (.....)
{
.....
for (.....)
{
.....
if(condition)
break;
.....
}
.....
}
Exit
from
inner
loop

(d)

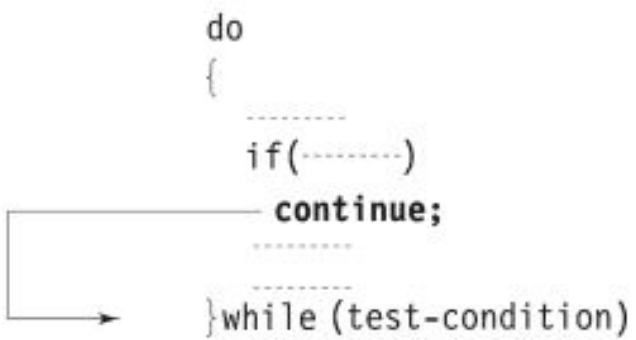
continue

- The continue statement is exactly opposite to break.
- The continue statement is used for continuing next iteration of loop statement.
- When it occurs in the loop, it does not terminate, but it skips the statements after this statement.
- It is useful when we want to continue the program without executing any part of the program.


while (test-condition)
{
----
if(---------)
continue;

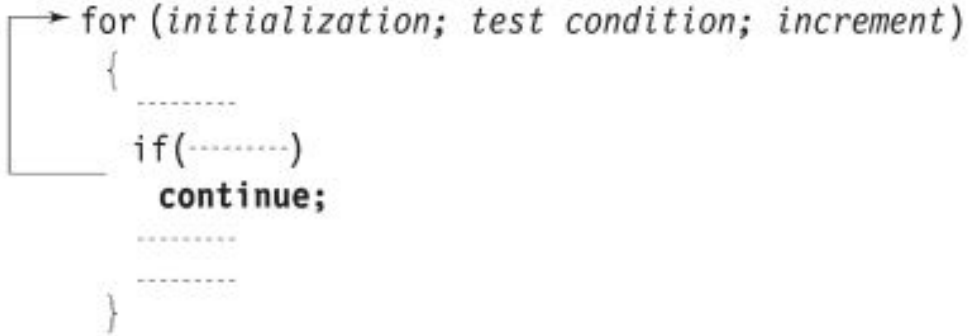
}

(a)


do
{
----
if(---------)
continue;

}while (test-condition)

(b)


for (initialization; test condition; increment)
{
----
if(---------)
continue;

}

(c)

exit

- The function ‘exit’ is used to quit the program.
- It terminates the program and returns the status code to the operating system.
- This function is defined in ‘stdlib.h’ header file.

goto

- A goto statement in C programming provides an unconditional jump from the 'goto' to a labeled statement in the same function.
- The syntax for a **goto** statement in C is as follows:

```
goto label;
..
.
label: statement;
```

```
#include <stdio.h>
int main () {
   /* local variable definition */
   int a = 10;
   /* do loop execution */
   LOOP:do {
      if( a == 15) {
         /* skip the iteration */
         a = a + 1;
         goto LOOP;
      }
      printf("value of a: %d\n", a);
      a++;
   }while( a < 20 );
   return 0;
}
```

```
value of a: 10
value of a: 11
value of a: 12
value of a: 13
value of a: 14
value of a: 16
value of a: 17
value of a: 18
value of a: 19
```

exit

- *exit(status);*

- The status code zero indicates that the program completed successfully.
- If there is a failure any other code has to be returned.

MCQS

What is the output of the following C program?

```c
#include <stdio.h>
int main()
{
	int a = 0, i = 0, b;
	for (i = 0; i < 5; i += 0.5)
	{
		a++;
		continue;
	}
	printf("%d", a);
	return 0;
}
```

A. 5
B. 10
C. No output (infinite loop)
D. Compilation error

Answer : C

What will be the output?

```
#include <stdio.h>
int main()
{
      int x = 1;
      do
      {
            continue;
            printf("%d", x);
            x++;
            break;
      } while (x <= 10);
      printf("After loop x=%d", x);
      return 0;
}
```

Answer : C

What will be the output of the following code?

```
#include <stdio.h>
int main()
{
      int s = 0;
      while (s++ < 10)
      {
            if (s < 4 && s < 9)
                  continue;
            printf("%d ", s);
      }
      return 0;
}
```

A. 1 2 3 4 5 6 7 8 9
B. 1 2 3 10
C. 4 5 6 7 8 9 10
D. 4 5 6 7 8 9
E. None of these

Answer : C

The keyword 'break' cannot be simply used within _____.

A. do-while

B. if-else

C. for

D. while

Answer : B

Point out the error, if any in the while loop.

```
#include <stdio.h>
int main()
{
      int i = 1;
      while ()
      {
            printf("%d ", i++);
            if (i > 10)
                  break;
      }
      return 0;
}
```

A. There should be a condition in the while loop
B. There should be at least a semicolon in the while
C. The while loop should be replaced with for loop.
D. No error

Answer : A

What will be the output of the program?

```
#include <stdio.h>
int main()
{
      int i;
      for (i = -1; i <= 10; i++)
      {
            if (i < 5)
                  continue;
            else
                  break;
            printf("Gets printed only once!!");

      }
      return 0;
}
```

A. No output
B. Gets printed only once!!
C. Compile time error
D. Run time error

Answer : A

What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
      int i = 0;
      if (i == 0)
      {
            printf("Hello");
            continue;
      }
      return 0;
}
```

A. Hello is printed infinite times
B. Hello
C. Varies
D. Compile time error

Answer : D

Nested Loops

- Similar to nested if statements, loops can be nested as well
- That is, the body of a loop can contain another loop
- For each iteration of the outer loop, the inner loop iterates completely

Nested Loops

- How many times will the string "Here" be printed?

```
count1 = 1;
while (count1 <= 10)
{
    count2 = 1;
    while (count2 <= 20)
    {
        printf ("Here");
        count2++;
    }
    count1++;
}
```

10 * 20 = 200

Looping- nested for statements

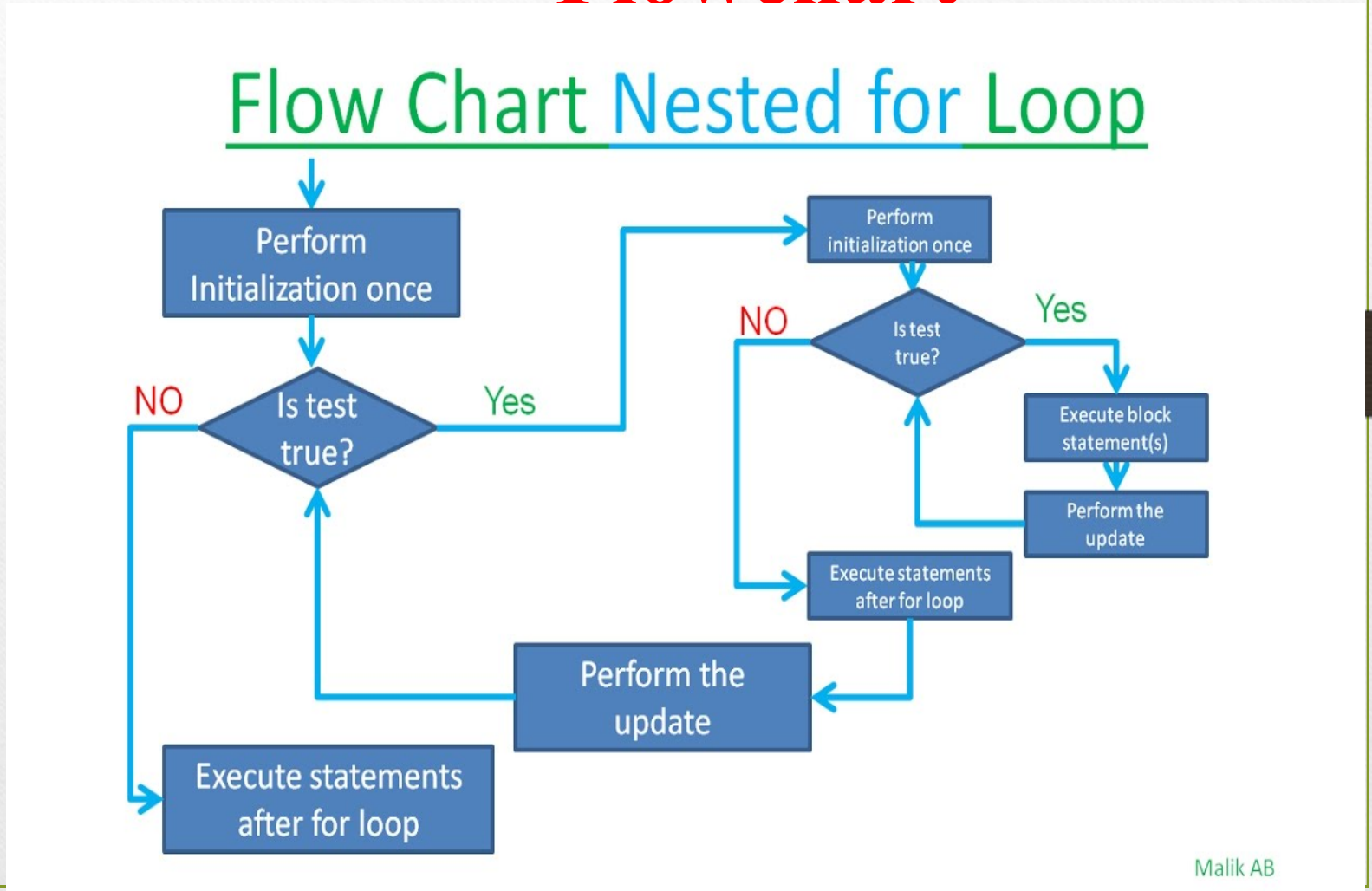
What are nested for loops?

Loop nested within loops

For example:

```
for(initialization; condition test ; updation)
{
        //outer loop statement;
        for(initialization; condition test ; updation)
        {
                //inner loop statements;
        }
        //outer loop statements;
}
```

Flowchart


Flow Chart Nested for Loop
Perform Initialization once
Is test true?
NO
Yes
Execute statements after for loop
Perform the update
Perform initialization once
Is test true?
NO
Yes
Execute block statement(s)
Perform the update
Execute statements after for loop
Malik AB

Looping- nested for statement working

```
for(int i=1; i<=2;i++)
{
    int j;
    for( j=1 ;i<=3 ; j++)
    {
        printf("%d %d",i,j);
    }
    printf("\n");
    }
```

OUTER FOR

INNER FOR

How many statements inside outer for?

How many statements inside inner for?

Looping- nested for statement working

i=1 2(for true)

```
for(int i=1; i<=2;i++)
{
    int j;
    for( j=1 ;i<=3 ; j++)
    {
        printf("%d %d",i,j);
    }
    printf("\n");
    }
```

j= 1 2 3 (for true)

i	j	printf
1	1	1 1
1	2	1 2
1	3	1 3
2	1	2 1
2	2	2 2
2	3	2 3

How many times i and j are printed?

Looping- nested for statement working

```
for(int i=1; i<=2;i++)
{
    int j;
    for( j=1 ;i<=3 ; j++)
    {
        printf("%d %d",i,j);
    }
    printf("\n");
    }
```

How many times outer for loop got executed ?

2

How many times inner for loop got executed ?

6

For each execution of the outer loop the inner loop is executed 3 times, hence

2*3=6

MCQS

How many times Hello will be printed in the below C code?

```c
#include <stdio.h>
int main()
{
      int k, j;
      for (k = 0; k <= 10; k += 2)
      {
            for (j = 1; j != k; j = j + 1)
            {
                  printf("Hello \n");
                  break;
            }
      }
      return 0;
}
```

A. 10
B. 5
C. 6
D. Infinite

Answer : C

What will be the output of the following C code?

```c
#include <stdio.h>
int main()
{
      int i = 0;
      int j = 0;
      for (i = 0; i < 5; i++)
      {
            for (j = 0; j < 4; j++)
            {
                  if (i > 1)
                        continue;
                  printf("Hi\n");
            }
      }
      return 0;
}
```

A. Hi is printed 9 times
B. Hi is printed 8 times
C. Hi is printed 7 times
D. Hi is printed 6 times

Answer : B

Programs

Programs

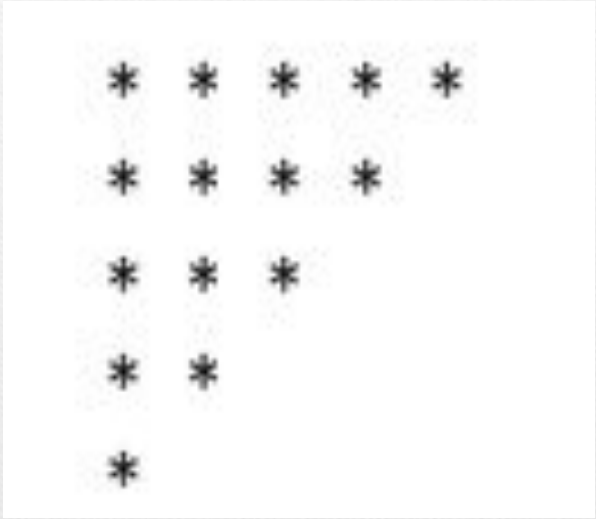
1. Write a program in C to display the triangle using “*” in the following pattern.

What is the output of the following program?

```
#include <stdio.h>
int main()
{
        int i, j, rows;
        rows = 5;
        for (i = 1; i <= rows; ++i)
        {
                for (j = 1; j <= i; ++j)
                {
                        printf("* ");
                }
                printf("\n");
        }
        return 0;
}
```

Programs

1. Write a program in C to display the triangle using “*” in the following pattern.



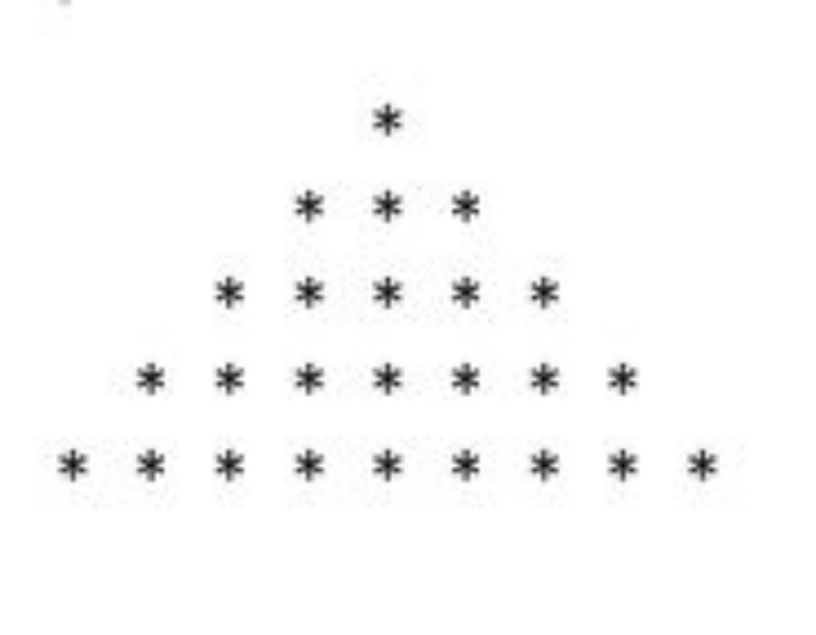
Programs

1. Write a program in C to display the triangle using "*" in the following pattern.

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

Programs

1. Write a program in C to display a pyramid with “*”.



Programs

1. Write a program in C to display a pyramid with “*”.

```
#include <stdio.h>
int main() {
    int i, space, rows, k = 0;
    printf("Enter the number of rows: ");
    scanf("%d", &rows);
    for (i = 1; i <= rows; ++i, k = 0) {
        for (space = 1; space <= rows - i; ++space) {
            printf("  ");
        }
        while (k != 2 * i - 1) {
            printf("* ");
            ++k;
        }
        printf("\n");
    }
    return 0;
}
```

Programs

1. Write a program in C to display a pyramid with “*”.

```c
#include <stdio.h>
int main()
{
	int rows, i, j, space;
	rows = 5;
	for (i = rows; i >= 1; --i)
	{
		for (space = 0; space < rows - i; ++space)
			printf("  ");
		for (j = i; j <= 2 * i - 1; ++j)
			printf("* ");
		for (j = 0; j < i - 1; ++j)
			printf("* ");
		printf("\n");
	}
	return 0;
}
```

Programs

1. Write a program in C to display the right angle triangle using numbers.

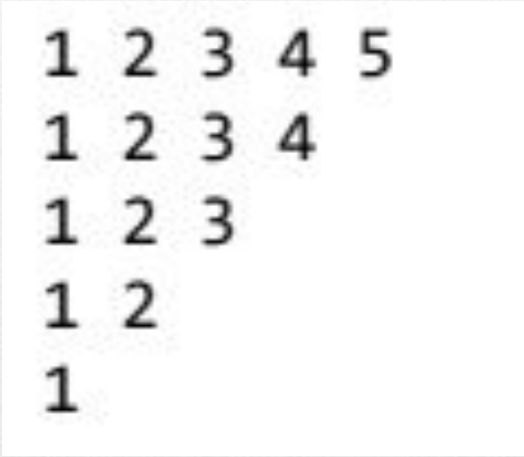
Programs

1. Write a program in C to display the right angle triangle using numbers.

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

Programs

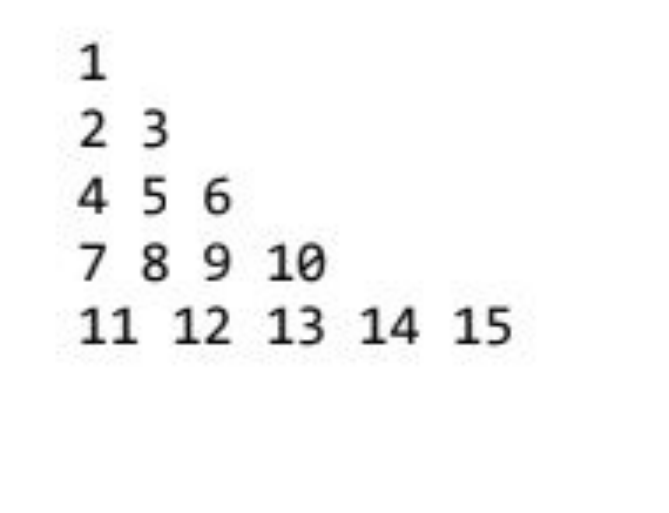
1. Write a program in C to display the right angle triangle using numbers.



```c
#include <stdio.h>
int main()
{
	int i, j, rows;
	rows = 5;
	for (i = rows; i >= 1; --i)
	{
		for (j = 1; j <= i; ++j)
		{
			printf("%d ", j);
		}
		printf("\n");
	}
	return 0;
}
```

Programs

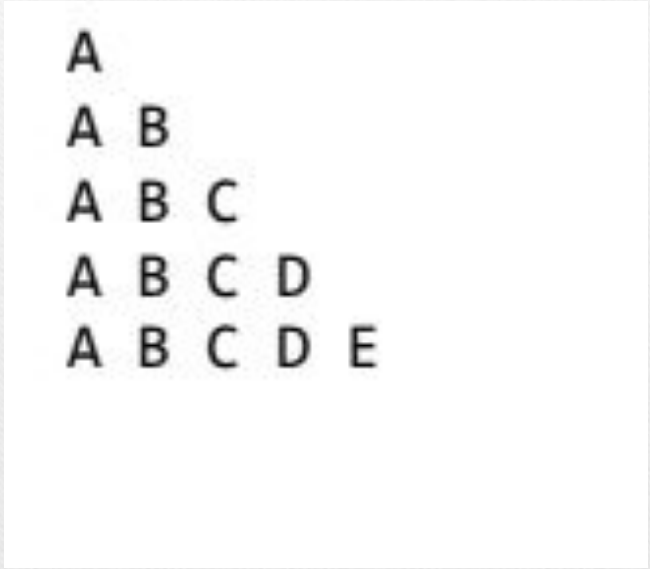
1. Write a program in C to display the right angle triangle using numbers.



```c
#include <stdio.h>
int main()
{
      int rows, i, j, number = 1;
      rows = 5;
      for (i = 1; i <= rows; i++)
      {
            for (j = 1; j <= i; ++j)
            {
                  printf("%d ", number);
                  ++number;
            }
            printf("\n");
      }
      return 0;
}
```

Programs

1. Write a program in C to display the right angle triangle using numbers.



```c
#include <stdio.h>
int main()
{
	int i, j;
	char ch;
	for (i = 1; i <= 5; i++)
	{
		ch = 'A';
		for (j = 1; j <= i; j++)
		{
			printf("%c ", ch++);
		}
		printf("\n");
	}
	return 0;
}
```

Simple Chessboard

Write a program that prints a simple chessboard.

Input format:

The first line contains the number of inputs T.
The lines after that contain a different values for size of the chessboard

Output format:

Print a chessboard of dimensions size * size. Print a Print W for white spaces and B for black spaces.

Input:

2
3
5

Output:

WBW
BWB
WBW
WBWBW
BWBWB
WBWBW
BWBWB
WBWBW

```c
#include <stdio.h>

int main()
{
	int n, size, i, j, count;
	scanf("%d", &n);
	while (n--)
	{
		scanf("%d", &size);
		count = 0;
		for (i = 0; i < size; i++)
		{
			for (j = 0; j < size; j++)
			{
				if (++count % 2 == 1)
					printf("W");
				else
					printf("B");
			}
			if (size % 2 == 0)
				count++;
			printf("\n");
		}
	}
}
```

Reverse and Add Until Get a Palindrome Number

Take a number, reverse it and add it to the original number until the obtained number is a palindrome.

Constraints

1<=num<=99999999

Sample Input 1

32

Sample Output 1

55

Sample Input 2

789

Sample Output 2

66066

```c
#include <stdio.h>
int main()
{
	long long int num, sum, revnum, tempnum, tempsum;
	scanf("%lld", &num);
	while (1)
	{
		revnum = 0;
		tempnum = num;
		while (num)
		{
			revnum = revnum * 10 + (num % 10);
			num = num / 10;
		}
		sum = tempnum + revnum;
		tempsum = sum;
		revnum = 0;
		while (sum)
		{
			revnum = revnum * 10 + (sum % 10);
			sum = sum / 10;
		}
		if (tempsum == revnum)
			break;
		num = tempsum;
	}
	printf("%lld", tempsum);
	return 0;
}
```

Lucky Number

A number is considered lucky if it contains either 3 or 4 or 3 and 4 both in it. Write a program to print the nth lucky number. Example, 1st lucky number is 3, and 2nd lucky number is 4 and 3rd lucky number is 33 and 4th lucky number is 34 and so on. Note that 13, 40 etc., are not lucky as they have other numbers in it.

The program should accept a number 'n' as input and display the nth lucky number as output.

Sample Input 1:

3

Sample Output 1:

33

Explanation:

Here the lucky numbers are 3, 4, 33, 34., and the 3rd lucky number is 33.

Sample Input 2:

34

Sample Output 2:

33344

```c
#include <stdio.h>
int main()
{
	long int i, j;
	int rem, n, count = 0, flag;
	scanf("%d", &n);
	for (i = 1; count <= n; i++)
	{
		flag = 0;
		j = i;
		while (j > 0)
		{
			rem = j % 10;
			if (rem == 3 || rem == 4)
				j = j / 10;
			else
			{
				flag = 1;
				break;
			}
		}
		if (flag == 0)
		{
			count++;
			if (count == n)
				break;
		}
	}
	printf("%ld", i);
	return 0;
}
```

Programs

1. Write a program in C to display a square matrix with alternative character in the cell(consider only 2 characters).Eg:

1	2	1
2	1	2
1	2	1

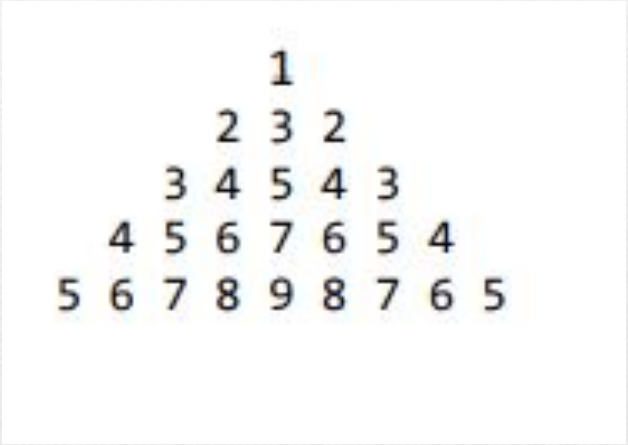
Programs

1. Write a program in C to display a square matrix with alternative character in the cell(consider only 2 characters).Eg:

```
#include<stdio.h>
int main()
{
    int n;
    int a=1,b=2;
    scanf("%d",&n);
    for(int i=1;i<=n;i++)
    {
        for(int j=1;j<=n;j++)
        {
            if((i%2==1 && j%2==1)||(i%2==0 && j%2==0))
                printf("%d ",a);
            else
                printf("%d ",b);
        }
        printf("\n");
    }
}
```

Programs

1. Write a program in C to display a following output.

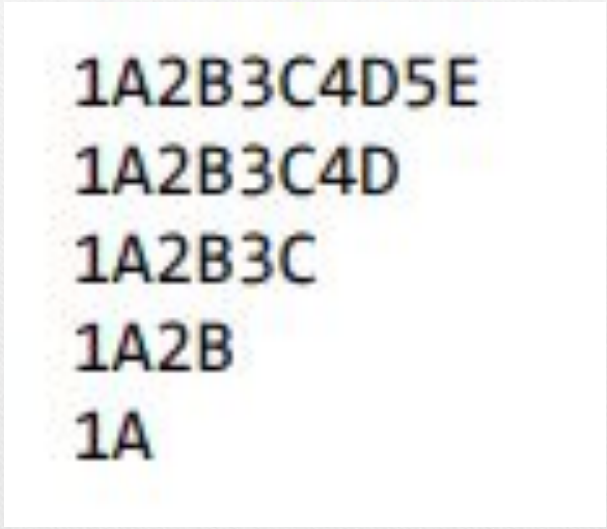


What is the output of the following program?

```
#include <stdio.h>
int main()
{
      int i, space, rows, k = 0, count = 0, count1 = 0;
      rows = 5;
      for (i = 1; i <= rows; ++i) {
            for (space = 1; space <= rows - i; ++space)
            {
                  printf("  ");
                  ++count;
            }
            while (k != 2 * i - 1)
            {
                  if (count <= rows - 1)
                  {
                        printf("%d ", i + k);
                        ++count;
                  }
                  else
                  {
                        ++count1;
                        printf("%d ", (i + k - 2 * count1));
                  }
                  ++k;
            }
            count1 = count = k = 0;
            printf("\n");
      }
      return 0;
}
```

Programs

1. Write a program in C to display a following output.



What is the output of the following program?

```
#include <stdio.h>
int main()
{
      int i, j, k;
      /*Run parent loop*/
      for (i = 5; i >= 1; i--)
      {
            for (j = 1, k = 'A'; j <= i; j++, k++)
            {
                  printf("%d%c", j, k);
            }
            printf("\n");
      }
      return 0;
}
```

Programs

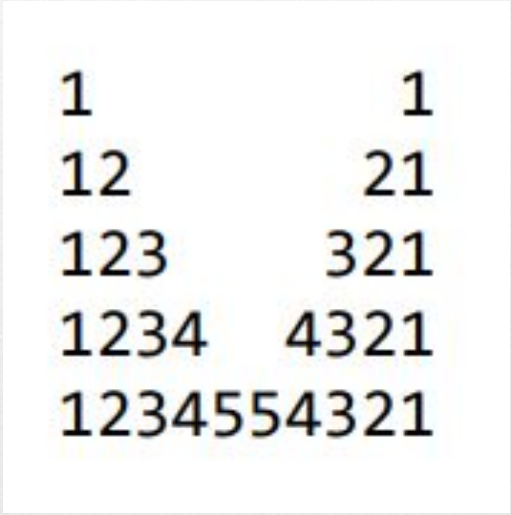
1. Write a program in C to display a following output.

What is the output of the following program?

```
#include<stdio.h>
#define MAX     5
int main()
{
      int i, j;
      int space = 0;
      /*run loop (parent loop) till number of rows*/
      for (i = MAX; i > 0; i--)
      {
            /*print first set of stars*/
            for (j = 0; j < i; j++)
            {
                  printf("*");
            }
            for (j = 0; j < space; j++)
            {
                  printf(" ");
            }
            /*print second set of stars*/
            for (j = 0; j < i; j++)
            {
                  printf("*");
            }
            printf("\n");
            space += 2;
      }
      return 0;
}
```

Programs

1. Write a program in C to display a following output.



What is the output of the following program?

```
#include<stdio.h>
int main()
{
      int i, j, k, l, m = 8, n = 1;
      for (i = 1; i <= 5; i++) {
            for (j = 1; j <= i; j++)
            {
                  printf("%d", j);
            }
            for (k = m; k >= 1; k--)
            {
                  printf(" ");
            }
            m = m - 2;
            for (l = n; l >= 1; l--)
            {
                  printf("%d", l);
            }
            n = n + 1;
            printf("\n");
      }
      return 0;
}
```