<https://engineeringinterviewquestions.com/c-programming-interview-questions-answers-pdf/>

**How will you print “Hello World” without semicolon?**  
**Ans:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| #include <stdio.h>  int main(void)  {      if (printf("Hello World")) {      }  }  Write a simple example of a structure in C Language  Ans: Structure is defined as a user-defined data type that is designed to store multiple data members of the different data types as a single unit. A structure will consume the memory equal to the summation of all the data members.  struct employee  {  char name[10];  int age;  }e1;  int main()  {  printf("Enter the name");  scanf("%s",e1.name);  printf("n");  printf("Enter the age");  scanf("%d",&e1.age);  printf("n");  printf("Name and age of the employee: %s,%d",e1.name,e1.age);  return 0;  }  //Example of Call by Value method  #include<stdio.h>  void change(int,int);  int main()  {  int a=25,b=50;  change(a,b);  printf("The value assigned to a is: %d",a);  printf("n");  printf("The value assigned to of b is: %d",b);  return 0;  }  void change(int x,int y)  {  x=100;  y=200;  }  //Output  The value assigned to of a is: 25  The value assigned to of b is: 50    //Example of Call by Reference  #include<stdio.h>  void change(int\*,int\*);  int main()  {  int a=25,b=50;  change(&a,&b);  printf("The value assigned to a is: %d",a);  printf("n");  printf("The value assigned to b is: %d",b);  return 0;  }  void change(int \*x,int \*y)  {  \*x=100;  \*y=200;  }  //Output  The value assigned to a is: 100  The value assigned to b is: 200  **Differentiate between call by value and call by reference.**  **Ans:**   |  |  |  | | --- | --- | --- | | **Factor** | **Call by Value** | **Call by Reference** | | **Safety** | Actual arguments cannot be changed and remain safe | Operations are performed on actual arguments, hence not safe | | **Memory Location** | Separate memory locations are created for actual and formal arguments | Actual and Formal arguments share the same memory space. | | **Arguments** | Copy of actual arguments are sent | Actual arguments are passed | |

**Example 1: Half Pyramid of \***

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

**C Program**

#include <stdio.h>

int main() {

int i, j, rows;

printf("Enter the number of rows: ");

scanf("%d", &rows);

for (i = 1; i <= rows; ++i) {

for (j = 1; j <= i; ++j) {

printf("\* ");

}

printf("\n");

}

return 0;

}

**Example 2: Half Pyramid of Numbers**

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

**C Program**

#include <stdio.h>

int main() {

int i, j, rows;

printf("Enter the number of rows: ");

scanf("%d", &rows);

for (i = 1; i <= rows; ++i) {

for (j = 1; j <= i; ++j) {

printf("%d ", j);

}

printf("\n");

}

return 0;

}

**Example 3: Half Pyramid of Alphabets**

A

B B

C C C

D D D D

E E E E E

**C Program**

#include <stdio.h>

int main() {

int i, j;

char input, alphabet = 'A';

printf("Enter an uppercase character you want to print in the last row: ");

scanf("%c", &input);

for (i = 1; i <= (input - 'A' + 1); ++i) {

for (j = 1; j <= i; ++j) {

printf("%c ", alphabet);

}

++alphabet;

printf("\n");

}

return 0;

}

**Example 4: Inverted half pyramid of \***

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

**C Program**

#include <stdio.h>

int main() {

int i, j, rows;

printf("Enter the number of rows: ");

scanf("%d", &rows);

for (i = rows; i >= 1; --i) {

for (j = 1; j <= i; ++j) {

printf("\* ");

}

printf("\n");

}

return 0;

}

**Example 5: Inverted half pyramid of numbers**

1 2 3 4 5

1 2 3 4

1 2 3

1 2

1

**C Program**

#include <stdio.h>

int main() {

int i, j, rows;

printf("Enter the number of rows: ");

scanf("%d", &rows);

for (i = rows; i >= 1; --i) {

for (j = 1; j <= i; ++j) {

printf("%d ", j);

}

printf("\n");

}

return 0;

}

**Example 6: Full Pyramid of \***

\*

\* \* \*

\* \* \* \* \*

\* \* \* \* \* \* \*

\* \* \* \* \* \* \* \* \*

**C Program**

#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;

}

**Example 7: Full Pyramid of Numbers**

1

2 3 2

3 4 5 4 3

4 5 6 7 6 5 4

5 6 7 8 9 8 7 6 5

**C Program**

#include <stdio.h>

int main() {

int i, space, rows, k = 0, count = 0, count1 = 0;

printf("Enter the number of rows: ");

scanf("%d", &rows);

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;

}

**Example 8: Inverted full pyramid of \***

\* \* \* \* \* \* \* \* \*

\* \* \* \* \* \* \*

\* \* \* \* \*

\* \* \*

\*

**C Program**

#include <stdio.h>

int main() {

int rows, i, j, space;

printf("Enter the number of rows: ");

scanf("%d", &rows);

for (i = rows; i >= 1; --i) {

for (space = 0; space < rows - i; ++space)

printf(" ");

}

while (k != 2 \* i - 1) {

printf("\* ");

++k;

}

printf("\n");

}

return 0;

}

**Example 9: Pascal's Triangle**

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

**C Program**

#include <stdio.h>

int main() {

int rows, coef = 1, space, i, j;

printf("Enter the number of rows: ");

scanf("%d", &rows);

for (i = 1; i <=rows; i++) {

for (space = 1; space <= rows - i; space++)

printf(" ");

for (j = 0; j <= i; j++) {

if (j == 0 || i == 0)

coef = 1;

else

coef = coef \* (i - j + 1) / j;

printf("%4d", coef);

}

printf("\n");

}

return 0;

}

Row Formula's Related To Pascal's Triangle

row 0 = 1

row 1 = (0+1), (1+0) = 1, 1

row 2 = (0+1), (1+1), (1+0) = 1, 2, 1

row 3 = (0+1), (1+2), (2+1), (1+0) = 1, 3, 3, 1

row 4 = (0+1), (1+3), (3+3), (3+1), (1+0) = 1, 4, 6, 4, 1

row 5 = (0+1), (1+4), (4+6), (6+4), (4+1),(1+0) = 1, 5, 10, 10, 5, 1

row 6 = (0+1), (1+5), (5+10), (10+10), (10+5), (5+1), (1+0) = 1, 6, 15, 20, 15, 6, 1

**Example 10: Floyd's Triangle.**

1

2 3

4 5 6

7 8 9 10

**C Program**

#include <stdio.h>

int main() {

int rows, i, j, number = 1;

printf("Enter the number of rows: ");

scanf("%d", &rows);

for (i = 1; i <= rows; i++) {

for (j = 1; j <= i; ++j) {

printf("%d ", number);

++number;

}

printf("\n");

}

return 0;

}

**Sample Input/ Output:-**

Enter the number of rows: 3  
\* \* \*  
\* \* \*  
\* \* \*

Enter the number of rows: 5  
\* \* \* \* \*  
\* \* \* \* \*  
\* \* \* \* \*  
\* \* \* \* \*  
\* \* \* \* \*

#include<stdio.h>

int main()

{

int n;

printf("Enter the number of rows: ");

scanf("%d",&n);

for(int r=1; r<=n; r++)

{

for(int c=1; c<=n; c++)

{

printf("\* ");

}

printf("\n");

}

return 0;

}

### Pattern Program 2

In the below pattern columns are started from 1 to N, and the number of columns is equal to the number of rows.

**Sample Input/ Output:-**

Enter the number of rows: 3  
1 2 3  
1 2 3  
1 2 3

Enter number of rows: 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

#include<stdio.h>

int main()

{

int n;

printf("Enter the number of rows: ");

scanf("%d",&n);

for(int r=1; r<=n; r++)

{

for(int c=1; c<=n; c++)

{

printf("%3d",c);

}

printf("\n");

}

return 0;

}

### Pattern Program 3

The below pattern program is similar to the previous one but in this pattern, each Nth row contains the same N value. It is also a square matrix where the number of rows is equal to the number of columns.

**Sample Input/ Output:-**

Enter the number of rows: 3  
1 1 1  
2 2 2  
3 3 3

Enter number of rows: 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

#include<stdio.h>

int main()

{

int n;

printf("Enter the number of rows: ");

scanf("%d",&n);

for(int r=1; r<=n; r++)

{

for(int c=1; c<=n; c++)

{

printf("%3d",r);

}

printf("\n");

}

return 0;

}

### Pattern Program 4

**Sample Input/ Output:-**

Enter the number of rows: 3  
a a a  
b b b  
c c c

Enter the number of rows: 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

#include<stdio.h>

int main()

{

int n;

printf("Enter number of rows: ");

scanf("%d",&n);

for(int r=1; r<=n; r++)

{

for(int c=1; c<=n; c++)

{

printf("%3c",r+96);

}

printf("\n");

}

return 0;

}

### Pattern Program 5

The below pattern is similar to the previous pattern program but instead of the lowercase letter, it has all uppercase letters in the pattern. The [ASCII value of the character](https://www.knowprogram.com/c-programming/ascii-value-of-character-and-in-range/) ‘A’ is 65.

**Sample Input/ Output:-**

Enter the number of rows: 3  
A A A  
B B B  
C C C

Enter the number of rows: 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

#include<stdio.h>

int main()

{

int n;

printf("Enter number of rows: ");

scanf("%d",&n);

for(int r=1; r<=n; r++)

{

for(int c=1; c<=n; c++)

{

printf("%3c",r+64);

}

printf("\n");

}

return 0;

}

### Pattern Program 6

In the previous pattern (i.e. pattern program 5) one row was containing similar characters.

But in the current pattern one row contain characters starting from ‘A’ to the number of rows and every row is similar.

**Sample Input/ Output:-**

Enter the number of rows: 3  
A B C  
A B C  
A B C

Enter the number of rows: 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

#include<stdio.h>

int main()

{

int n;

printf("Enter the number of rows: ");

scanf("%d", &n);

for(int r=1; r<=n; r++)

{

for(int c=1; c<=n; c++)

{

printf("%3c",c+64);

}

printf("\n");

}

return 0;

}

### Pattern Program 7

**Sample Input/ Output:-**

Enter the number of rows: 3  
a b c  
a b c  
a b c

Enter the number of rows: 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

#include<stdio.h>

int main()

{

int n;

printf("Enter the number of rows: ");

scanf("%d",&n);

for(int r=1; r<=n; r++)

{

for(int c=1; c<=n; c++)

{

printf("%3c",c+96);

}

printf("\n");

}

return 0;

}

### Pattern Program 8

The above pattern contains both uppercase and lowercase characters but in alternate rows. And one row contains a similar character.

In these patterns, if the row number is odd then the small letter alphabet is printed. If the row number is even then the capital letter alphabet is printed.

**Sample Input/ Output:-**

Enter the number of rows: 3  
a a a  
B B B  
c c c

Enter the number of rows: 9  
a a a a a a a a a  
B B B B B B B B B  
c c c c c c c c c  
D D D D D D D D D  
e e e e e e e e e  
F F F F F F F F F  
g g g g g g g g g  
H H H H H H H H H  
i i i i i i i i i

#include<stdio.h>

int main()

{

int n;

printf("Enter the number of rows: ");

scanf("%d", &n);

for(int r=1; r<=n; r++)

{

for(int c=1; c<=n; c++)

{

if(r%2==0) printf("%3c", r+64);

else printf("%3c", r+96);

}

printf("\n");

}

return 0;

}

### Pattern Program 9

**Sample input/outputs:-**

Enter the number of rows: 5  
1 1 1 1 1  
1 1 1 2 2  
1 1 3 3 3  
1 4 4 4 4  
5 5 5 5 5

Enter the number of rows: 10  
1 1 1 1 1 1 1 1 1 1  
1 1 1 1 1 1 1 1 2 2  
1 1 1 1 1 1 1 3 3 3  
1 1 1 1 1 1 4 4 4 4  
1 1 1 1 1 5 5 5 5 5  
1 1 1 1 6 6 6 6 6 6  
1 1 1 7 7 7 7 7 7 7  
1 1 8 8 8 8 8 8 8 8  
1 9 9 9 9 9 9 9 9 9  
10 10 10 10 10 10 10 10 10 10

#include<stdio.h>

int main()

{

int n,r,c;

printf("Enter number of rows: ");

scanf("%d",&n);

// outer loop

for(r=1; r<=n; r++)

{

// inner loop

for(c=1; c<=n; c++)

{

if(c <= n-r) printf("%4d",1);

else printf("%4d",r);

} // end of inner loop

printf("\n");

} // end of outer loop

return 0;

}

### Pattern Program 10

**Sample input/outputs:-**

Enter the number of rows: 5  
\* \* \* \* \*  
\* A B C \*  
\* D E F \*  
\* G H I \*  
\* \* \* \* \*

Enter the number of rows: 10  
\* \* \* \* \* \* \* \* \* \*  
\* A B C D E F G H \*  
\* I J K L M N O P \*  
\* Q R S T U V W X \*  
\* Y Z A B C D E F \*  
\* G H I J K L M N \*  
\* O P Q R S T U V \*  
\* W X Y Z A B C D \*  
\* E F G H I J K L \*  
\* \* \* \* \* \* \* \* \* \*

In the given above pattern, each row starts and ends with a star symbol, and the first and last rows contain only a star symbol whereas the remaining columns contain the alphabet.

In this pattern, when row number and column number are first and last then there are stars only. Otherwise, there are alphabets from ‘A’ to ‘Z’. When the alphabet is reached at ‘Z’ then it again started from ‘A’.

#include<stdio.h>

int main()

{

int n,r,c;

char ch = 'A';

printf("Enter number of rows: ");

scanf("%d",&n);

// outer loop

for(r=1; r<=n; r++)

{

// inner loop

for(c=1; c<=n; c++)

{

if(c==1||r==1||c==n||r==n) printf("\* ");

else printf("%c ",ch++);

if(ch > 'Z') ch='A';

} // end of inner loop

printf("\n");

} // end of outer loop

return 0;

}

## Pattern Programs in C for Half Pyramid

### Half Pyramid using \* and Increment Operator

**Pattern Program 11**

Sample Output:-

\*  
\* \*  
\* \* \*  
\* \* \* \*  
\* \* \* \* \*

#include<stdio.h>

int main()

{

int i,j;

// outer for loop to represent rows

for(i=1;i<=5;i++)

{

// inner for loop to represent columns

for(j=1;j<=i;j++)

{

printf("\* ");

}

// new line

printf("\n");

}

return 0;

}

### Half Pyramid using \*, Increment & Decrement Operators

Here we will use the decrement operator for the outer loop and increment operator for the inner loop to print the above print.

**Pattern Program 12**

#include<stdio.h>

int main()

{

int i,j;

// outer for loop with decrement operator

for(i=5;i>=1;i--)

{

// inner for loop with increment operator

for(j=i;j<=5;j++)

{

printf("\* ");

}

// new line

printf("\n");

}

return 0;

}

Now, we will use the increment operator for the outer loop and decrement operator for inner loop to print same (above) pattern.

#include<stdio.h>

int main()

{

int i,j;

// outer for loop with increment operator

for(i=1;i<=5;i++)

{

// inner for loop with decrement operator

for(j=i;j>=1;j--)

{

printf("\* ");

}

printf("\n");

}

return 0;

}

### Half Pyramid using Numbers

**Pattern Program 13**

Sample Output:-

1  
2 1  
3 2 1  
4 3 2 1  
5 4 3 2 1

#include<stdio.h>

int main()

{

int i,j;

for(i=1;i<=5;i++)

{

for(j=i;j>=1;j--)

{

printf("%d ",j);

}

printf("\n");

}

return 0;

}

## Pattern Programs in C for Floyd’s Triangle

**Pattern Program 14**

Sample input/outputs:-

Enter number of rows: 5

1

2 3

4 5 6

7 8 9 10

11 12 13 14 15

Enter number of rows: 10

1

2 3

4 5 6

7 8 9 10

11 12 13 14 15

16 17 18 19 20 21

22 23 24 25 26 27 28

29 30 31 32 33 34 35 36

37 38 39 40 41 42 43 44 45

46 47 48 49 50 51 52 53 54 55

C Program for the Floyd’s Triangle Pattern,

#include<stdio.h>

int main()

{

int n, r, c, a=1;

printf("Enter number of rows: ");

scanf("%d", &n);

for(r=1; r<=n; r++)

{

for(c=1; c<=r; c++)

printf("%5d",a++);

printf("\n");

}

return 0;

}

## Some Similar Pattern Programs in C

### Pattern Program 15

**Sample input/outputs:-**

Enter number of rows: 5

1

3 2

4 5 6

10 9 8 7

11 12 13 14 15

Enter number of rows: 10

1

3 2

4 5 6

10 9 8 7

11 12 13 14 15

21 20 19 18 17 16

22 23 24 25 26 27 28

36 35 34 33 32 31 30 29

37 38 39 40 41 42 43 44 45

55 54 53 52 51 50 49 48 47 46

In this pattern, when the row is odd then digits are printed from left to right and when the row is even then digits are printed from right to left.

#include<stdio.h>

int main()

{

int n, r, c;

int a=1, b;

printf("Enter number of rows: ");

scanf("%d",&n);

for(r=1; r<=n; r++)

{

b=a+r-1;

for(c=1; c<=r; c++, a++)

{

if(r%2==1) printf("%5d",a);

else printf("%5d",b--);

}

printf("\n");

}

return 0;

}

### Pattern Program 16

**Sample input/outputs:-**

Enter the number of rows: 5  
\*  
$ \*  
\* $ \*  
$ \* $ \*  
\* $ \* $ \*

Enter the number of rows: 10  
\*  
$ \*  
\* $ \*  
$ \* $ \*  
\* $ \* $ \*  
$ \* $ \* $ \*  
\* $ \* $ \* $ \*  
$ \* $ \* $ \* $ \*  
\* $ \* $ \* $ \* $ \*  
$ \* $ \* $ \* $ \* $ \*

In this pattern, if row number + column number is odd then there is a star (\*) symbol at that place else there is a dollar ($) symbol.

#include<stdio.h>

int main()

{

int n, r, c;

printf("Enter number of rows: ");

scanf("%d",&n);

for(r=1; r<=n; r++)

{

for(c=1; c<=r; c++)

{

if((r+c)%2==0) printf("\* ");

else printf("$ ");

}

printf("\n");

}

return 0;

}

### Pattern Program 17

**Sample input/outputs:-**

Enter the number of rows: 5  
A  
1 1  
B B B  
2 2 2 2  
C C C C C

Enter the number of rows: 10  
A  
1 1  
B B B  
2 2 2 2  
C C C C C  
3 3 3 3 3 3  
D D D D D D D  
4 4 4 4 4 4 4 4  
E E E E E E E E E  
5 5 5 5 5 5 5 5 5 5

In this pattern, when the row number is odd then the same alphabet is printed in the row otherwise the same digit is printed in that row.

#include<stdio.h>

int main()

{

int n, r, c, a=0;

char ch='A';

printf("Enter number of rows: ");

scanf("%d",&n);

for(r=1; r<=n; r++)

{

for(c=1; c<=r; c++)

{

if(r%2==1) printf("%5c",ch);

else printf("%5d",a);

}

printf("\n");

if(r%2==0) ch++;

else a++;

}

return 0;

}

### Pattern Program 18

**Sample Input/Output:-**

Enter number of rows: 3  
 \*  
 \*\*  
\*\*\*

Enter number of rows: 5

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

C Program for the above pattern,

#include<stdio.h>

int main()

{

int n, r, c, s;

printf("Enter number of rows: ");

scanf("%d",&n);

for(r=1;r<=n;r++)

{

for(s=1;s<=n-r;s++) printf(" ");

for(c=1;c<=r;c++) printf("\*");

printf("\n");

}

return 0;

}

### Pattern Program 19

**Sample Input/Output:-**

Enter number of rows: 3  
 \*  
 \* \*  
\* \* \*

Enter number of rows: 5

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

C Program to print pyramid of starts,

#include<stdio.h>

int main()

{

int n, r, c, s;

printf("Enter number of rows: ");

scanf("%d",&n);

for(r=1;r<=n;r++)

{

for(s=1;s<=n-r;s++) printf(" ");

for(c=1;c<=r;c++) printf("\* ");

printf("\n");

}

return 0;

}

The difference between the previous program and this program is only one space. There is an extra space in the line *for(c=1;c<=r;c++) printf("\* ");* in comparison to the previous program.

## Pattern Programs in C for Inverted Half Pyramid

### Half Pyramid using \* and Decrement Operators

**Pattern Program 20**

Sample Output:-

\* \* \* \* \*  
\* \* \* \*  
\* \* \*  
\* \*  
\*

#include<stdio.h>

int main()

{

int i,j;

for(i=5;i>=1;i--)

{

for(j=i;j>=1;j--)

{

printf("\* ");

}

printf("\n");

}

return 0;

}

## Some Similar Pattern Programs in C

**Pattern program 21:- Display Given Pattern**

Print below pattern using increment operators only. This pattern is a combination of the previous two patterns, where the sign is changed from \* to -.

- - - - \*  
- - - \* \*  
- - \* \* \*  
- \* \* \* \*  
\* \* \* \* \*

#include<stdio.h>

int main()

{

int i,j,k;

for(i=1;i<=5;i++)

{

for(j=i;j<5;j++)

{

printf("- ");

}

for(k=1;k<=i;k++)

{

printf("\* ");

}

printf("\n");

}

return 0;

}

## Pattern Programs in C for Full Pyramid

**Pattern program 22: Display Full Pyramid**

Sample Input/Output:-

Enter number of rows: 5  
 \*  
 \*\*\*  
 \*\*\*\*\*  
 \*\*\*\*\*\*\*  
\*\*\*\*\*\*\*\*\*

Enter number of rows: 7

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*

C program for the above full pyramid star pattern,

#include<stdio.h>

int main()

{

int n, r, c, s;

printf("Enter number of rows: ");

scanf("%d",&n);

for(r=1;r<=n;r++)

{

for(s=1;s<=n-r;s++) printf(" ");

for(c=1;c<=(2\*r-1);c++) printf("\*");

printf("\n");

}

return 0;

}

### Full Pyramid on the Center of the Screen

**Pattern program 23**

Problem:- Display full pyramid on the center of the computer screen.

Enter number of rows: 7

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*

Generally, on a computer screen, we can print Maximum of 80 characters horizontally. Here we will print a full pyramid for N lines.

#include<stdio.h>

int main()

{

int n,i,j,k,c=80;

printf("Enter number of rows: ");

scanf("%d", &n);

for(i=1;i<=n;i++)

{

for(j=1;j<=(c/2-i);j++)

{

printf(" "); // blank space

}

for(k=1;k<=(2\*i-1);k++)

{

printf("\*");

}

printf("\n");

}

return 0;

}

### Full Pyramid with Numbers

**Pattern program 24**

Sample Input/Output:-

Enter number of rows: 5  
 1  
 121  
 12321  
 1234321  
 123454321

Enter number of rows: 7

1

121

12321

1234321

123454321

12345654321

1234567654321

C Program for the above full pyramid pattern with numbers

#include<stdio.h>

int main()

{

int n, r, c, k, a;

printf("Enter number of rows: ");

scanf("%d",&n);

for(r=1;r<=n;r++)

{

for(c=1;c<=n-r;c++) printf(" ");

for(k=1;k<=(2\*r-1);k++)

{

if(k<r) printf("%d",k);

else if(k==r)

{

printf("%d",k);

a=k;

}

else printf("%d",--a);

}

printf("\n");

}

return 0;

}

In this program, when column number (except space) is less than row number than digits are continuously increasing and after than digits are decreasing.

## Pattern Programs in C for Inverted Full Pyramid

**Pattern program 25**

Sample Input/Output:-

Enter number of rows: 5  
\* \* \* \* \*  
 \* \* \* \*  
 \* \* \*  
 \* \*  
 \*

Enter number of rows: 7

\* \* \* \* \* \* \*

\* \* \* \* \* \*

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

C program for inverted full pyramid of stars

#include<stdio.h>

int main()

{

int n, r, c, s;

printf("Enter number of rows: ");

scanf("%d",&n);

for(r=n;r>=1;r--)

{

for(s=1;s<=n-r;s++) printf(" ");

for(c=1;c<=r;c++) printf("\* ");

printf("\n");

}

return 0;

}

## More Pattern programs in C

**Pattern Program 26**

Sample Input/Output:-

Enter number of rows: 5

\* \*

\*\* \*\*

\*\*\* \*\*\*

\*\*\*\* \*\*\*\*

\*\*\*\*\*\*\*\*\*\*

Enter number of rows: 7

\* \*

\*\* \*\*

\*\*\* \*\*\*

\*\*\*\* \*\*\*\*

\*\*\*\*\* \*\*\*\*\*

\*\*\*\*\*\* \*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*

C program for the above given pattern,

#include<stdio.h>

int main()

{

int n, r, c;

printf("Enter number of rows: ");

scanf("%d",&n);

for(r=1;r<=n;r++)

{

for(c=1;c<=2\*n;c++)

if(c<=r||c>(2\*n-r)) printf("\* ");

else printf(" ");

printf("\n");

}

return 0;

}

**Pattern Program 27**

Sample Input/Output:-

Enter number of lines: 5

ABCDEFGFEDCBA

ABCDE EDCBA

ABCD DCBA

ABC CBA

AB BA

A A

Enter number of lines: 7

ABCDEFGHIHGFEDCBA

ABCDEFG GFEDCBA

ABCDEF FEDCBA

ABCDE EDCBA

ABCD DCBA

ABC CBA

AB BA

A A

C Program for the above given pattern

#include<stdio.h>

int main()

{

int n;

char ch;

printf("Enter number of lines: ");

scanf("%d",&n);

for(int i=0; i<=n; i++)

{

ch = 'A';

for(int j=0; j<=n-i; j++, ch++)

{

printf("%c",ch);

}

if(i==0)

{

printf("%c",ch);

}

else

{

for(int k=0; k<(2\*i)+1; k++){

printf(" ");

}

}

ch--;

for(int j=0; j<=n-i; j++, ch--)

{

printf("%c",ch);

}

printf("\n");

}

return 0;

}

s \*

\* \* \*

\* \* \* \* \*

\* \* \* \* \* \* \*