**1. Square pattern in C**

The simplest pattern you can draw using C is a square pattern. It has a shape of a square or rectangle.

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

**Follow these simple steps to create the square pattern in C:**

1. Take the size of the square or take user input.
2. Create a nested loop where the external loop prints rows and the inner loop prints each star in the column.
3. Print the stars using the command printf("\*").

#include <stdio.h>

int main() {

// take a size

// you can take user input

int size = 5;

// external loop

for (int i = 0; i < size; i++) {

// internal loop

for (int j = 0; j < size; j++) {

printf("\*");

}

printf("\n");

}

return 0;

}

**Output:**

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

**2. Hollow Square pattern in C**

hollow square is a variation of square pattern. It has a similar shape but is hollow inside.

\*\*\*\*\*

\* \*

\* \*

\* \*

\*\*\*\*\*

**Steps to create a hollow square pattern in C are as follows:**

1. Start with the size of the square.
2. Create a nested loop. Here the inner loop is a bit complex.
3. If the row is first or last print only star.
4. If the row is not first or last then print star at first and last position of row and space at the middle position.
5. Create a new line at the end of each row.

#include <stdio.h>

int main() {

// size of the square

int size = 5;

// external loop

for (int i = 0; i < size; i++) {

// innternal loop

for (int j = 0; j < size; j++) {

// in first and last row print only stars

if (i == 0 || i == size - 1) {

printf("\*");

}

else {

// at first and last position

// of row print stars else print spaces

if (j == 0 || j == size - 1) {

printf("\*");

}

else {

printf(" ");

}

}

}

printf("\n");

}

return 0;

}

**Output:**

\*\*\*\*\*

\* \*

\* \*

\* \*

\*\*\*\*\*

**3. Right triangle pattern program in C**

The right triangle star pattern in C is a right angle triangle that has its perpendicular line at the right side of the triangle.

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

You can see the pattern above. Before printing stars, we have to print spaces in a certain pattern to push stars to the right side of the triangle.

Follow the steps below to create the right triangle star pattern in C:

1. Start with creating an external loop with 2 inner loops, 1st to print spaces and 2nd to print stars.
2. In the 1st inner loop start printing spaces for the size minus row number.
3. In the 2nd inner loop start printing stars for the row number.
4. At the end of both inner loops print a new line.

#include <stdio.h>

int main() {

// set size

// or take user input

int size = 5;

for (int i = 0; i < size; i++) {

// printing spaces before stars

for (int j = 1; j < size-i; j++) {

printf(" ");

}

// printing stars

for (int k = 0; k <= i; k++) {

printf("\*");

}

printf("\n");

}

return 0;

}

**Output:**

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

report this ad

**4. Right Down Triangle**

The right down triangle is another triangle pattern which is a water image of a right triangle.

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

You can see the pattern above. It is very much the same as the right triangle star pattern with just little modifications in the code. Here is the complete code for this.

#include <stdio.h>

int main() {

// take size

int size = 5;

for (int i = 0; i < size; i++) {

// printing spaces

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

printf(" ");

}

// printing stars

for (int j = size; j > i; j--) {

printf("\*");

}

printf("\n");

}

return 0;

}

**Output:**

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

**5. Left triangle star pattern in C**

The left triangle star pattern is a triangle with a perpendicular line at the left side of the triangle.

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

**Follow the steps below to create the left triangle star pattern in C:**

1. Take the size of the triangle.
2. Start with creating a nested loop where the inner loop prints the star a number of times as the row number.
3. At the end of each row print new line by print("\n").

#include <stdio.h>

int main() {

// take size of triangle

int size = 5;

for (int i = 0; i < size; i++) {

// printing star in a row

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

printf("\*");

}

printf("\n");

}

return 0;

}

**Output:**

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

**6. Left Down Triangle**

The left down triangle is a variation of the left triangle star pattern. It is a water image of this pattern.

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

With just a little modification in code, you can reverse the order of printing stars and spaces and create this pattern. Here is the complete code for this.

#include <stdio.h>

int main() {

// size of the triangle

int size = 5;

for (int i = 0; i < size; i++) {

// printing stars

for (int j = 0; j < size-i; j++) {

printf("\*");

}

printf("\n");

}

return 0;

}

**Output:**

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

**7. Hollow triangle star pattern in C**

The hollow triangle star pattern is a little complex structure of a triangle. It is a triangle with a hollow space in the middle.

\*

\*\*

\* \*

\* \*

\* \*

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

You can see the pattern above. You have to manage the spaces and stars in a certain way to create this pattern.

1. Set the size of your triangle.
2. Use nested loops where the external loop handles rows and the internal loop handles each column of the row.
3. In the internal loop check if it's not the last row and is 1st and last position of the row then print star, else print space.
4. In the last row print only stars.

#include <stdio.h>

int main() {

int size = 5;

// creating hollow triangle

for (int i = 1; i <= size; i++) {

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

// operation of non-last row

if (i != size) {

// 1st or last position of row print star

if (j == 0 || j == i-1) {

printf("\*");

} else {

printf(" ");

}

}

// last row print only star

else {

printf("\*");

}

}

printf("\n");

}

return 0;

}

**Output:**

\*

\*\*

\* \*

\* \*

\* \*

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

**8. Pyramid star pattern in C**

The pyramid pattern in C is a very famous pattern shape that looks like an equilateral triangle.

\*

\*\*\*

\*\*\*\*\*

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

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

The complete step to create a pyramid star pattern in C is given below:

1. Take the size of the pyramid as input or a fixed number (here 5).
2. Create an external loop with 2 internal loops inside. 1st loop prints spaces and 2nd loop prints stars.
3. Print spaces for a number of times as size minus row number.
4. Print stars for 2 times as row number minus 1.
5. Break the line after each row.

#include <stdio.h>

int main() {

int size = 5;

// creating pyramid

for (int i = 0; i < size; i++) {

// print spaces before stars

for (int j = 0; j < size-i-1; j++) {

printf(" ");

}

// print stars

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

printf("\*");

}

printf("\n");

}

return 0;

}

**Output:**

\*

\*\*\*

\*\*\*\*\*

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

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

**Stay Ahead, Learn More**

* [Pattern Programs in C++](https://www.tutorialstonight.com/pattern-program-in-c++)
* [C Program to Print Pyramid Pattern of Numbers](https://www.tutorialstonight.com/c-program-to-print-pyramid-pattern-of-numbers)

**9. Hollow pyramid pattern program in C**

The hollow pyramid is a normal pyramid but with a hollow space in the middle.

\*

\* \*

\* \*

\* \*

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

You can see the pattern above. This is a bit complicated to create this pattern. Here are steps to follow:

1. Create a nested loop with 2 internal loops to print spaces and stars.
2. In the first loop print only spaces for a number of times as size minus row number.
3. The functioning of the second loop is quite complex. In this loop check if it is the first or the last row then only print stars. If it is another row then check if it is 1st and last position of the row then print star, else print space.
4. Break the line after 2nd internal loop.

#include <stdio.h>

int main() {

// size of the pyramid

int size = 5;

for (int i = 0; i < size; i++) {

// print spaces

for (int j = 0; j < size-i-1; j++) {

printf(" ");

}

// print stars

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

if(k == 0 || k == 2\*i || i == size-1) {

printf("\*");

} else {

printf(" ");

}

}

printf("\n");

}

return 0;

}

**Output:**

\*

\* \*

\* \*

\* \*

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

**10. Reverse pyramid star pattern in C**

The reverse pyramid star pattern in C is another version of the pyramid star pattern that has nothing but the 180-degree rotation of the pyramid star pattern.

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

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

\*\*\*\*\*

\*\*\*

\*

Here is the complete code to create a reverse pyramid pattern in C.

#include <stdio.h>

int main() {

// size of the pyramid

int size = 5;

for (int i = 0; i < size; i++) {

// print spaces

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

printf(" ");

}

// print stars

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

printf("\*");

}

printf("\n");

}

return 0;

}

**Output:**

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

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

\*\*\*\*\*

\*\*\*

\*

**11. Hourglass star pattern in C**

The hourglass pattern is a pattern having a shape like an hourglass. It is a combination of a pyramid and a hollow pyramid.

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

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

\*\*\*\*\*

\*\*\*

\*

\*\*\*

\*\*\*\*\*

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

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

Here is the complete code to create an hourglass pattern in C.

#include <stdio.h>

int main() {

int size = 5;

// reversed pyramid star pattern

for (int i = 0; i < size; i++) {

// printing spaces

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

printf(" ");

}

// printing star

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

printf("\*");

}

printf("\n");

}

// pyramid star pattern

for (int i = 2; i <= size; i++) {

// printing spaces

for (int j = size; j > i; j--) {

printf(" ");

}

// printing star

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

printf("\*");

}

printf("\n");

}

return 0;

}

**Output:**

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

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

\*\*\*\*\*

\*\*\*

\*

\*\*\*

\*\*\*\*\*

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

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

**12. Diamond star pattern in C**

Shown below is the diamond star pattern. It is a mixture of the pyramid and reverses the pyramid star patterns.

\*

\*\*\*

\*\*\*\*\*

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

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

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

\*\*\*\*\*

\*\*\*

\*

Here is the complete code to create a diamond star pattern in C.

#include <stdio.h>

int main() {

int size = 5;

// upside pyramid

for (int i = 1; i <= size; i++) {

// printing spaces

for (int j = size; j > i; j--) {

printf(" ");

}

// printing star

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

printf("\*");

}

printf("\n");

}

// downside pyramid

for (int i = 1; i <= size-1; i++) {

// printing spaces

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

printf(" ");

}

// printing star

for (int k = (size-i)\*2-1; k > 0; k--) {

printf("\*");

}

printf("\n");

}

return 0;

}

**Output:**

\*

\*\*\*

\*\*\*\*\*

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

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

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

\*\*\*\*\*

\*\*\*

\*

**13. Hollow diamond star pattern in C**

The hollow diamond pattern is a simple diamond pattern with stars only at the boundary.

\*

\* \*

\* \*

\* \*

\* \*

\* \*

\* \*

\* \*

\*

Here is the complete code to create a hollow diamond star pattern in C.

#include <stdio.h>

int main() {

int size = 5;

// upside pyramid

for (int i = 1; i <= size; i++) {

// printing spaces

for (int j = size; j > i; j--) {

printf(" ");

}

// printing star

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

if (k == 0 || k == 2\*i-2) {

printf("\*");

}

else {

printf(" ");

}

}

printf("\n");

}

// downside triangle

for (int i = 1; i < size; i++) {

// printing spaces

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

printf(" ");

}

// printing star

for (int k = (size-i)\*2-1; k >= 1; k--) {

if (k == 1 || k == (size-i)\*2-1) {

printf("\*");

}

else {

printf(" ");

}

}

printf("\n");

}

return 0;

}

**Output:**

\*

\* \*

\* \*

\* \*

\* \*

\* \*

\* \*

\* \*

\*

**14. Right pascal star pattern**

Shown below is the right pascal star pattern.

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

You can see the pattern above it is a mixture of the left triangle and reverse left triangle. The complete code of the right pascal star pattern in C is given below.

#include <stdio.h>

int main() {

// right pasal triangle

int size = 5;

for (int i = 1; i <= size; i++) {

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

printf("\*");

}

printf("\n");

}

for (int i = 1; i <= size-1; i++) {

for (int j = 0; j < size-i; j++) {

printf("\*");

}

printf("\n");

}

return 0;

}

**Output:**

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

**15. Left pascal star pattern**

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

The above pattern is the left pascal star pattern. Code to create this is given below.

#include <stdio.h>

int main() {

// left pasal triangle

int size = 5;

for (int i = 1; i <= size; i++) {

for (int j = 0; j < size-i; j++) {

printf(" ");

}

for (int k = 0; k < i; k++) {

printf("\*");

}

printf("\n");

}

for (int i = 1; i <= size-1; i++) {

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

printf(" ");

}

for (int k = 0; k < size-i; k++) {

printf("\*");

}

printf("\n");

}

return 0;

}

**Output:**

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

**16. Plus pattern program in C**

plus pattern has a shape of the mathematical plus sign(+).

\*

\*

\*\*\*\*\*

\*

\*

The code to create plus pattern in C is given below.

#include <stdio.h>

int main() {

// size of plus, use odd number

int size = 5;

for (int i = 0; i < size; i++) {

for (int j = 0; j < size; j++) {

// print only stars in middle row

if (i == size / 2) {

printf("\*");

}

// other than middle row, print star only at index size/2

else {

if (j == size / 2) {

printf("\*");

} else {

printf(" ");

}

}

}

printf("\n");

}

return 0;

}

**Output:**

\*

\*

\*\*\*\*\*

\*

\*

**17. Cross pattern program in C**

The cross pattern has the shape of the mathematical cross sign(x).

\* \*

\* \*

\*

\* \*

\* \*

The code to create cross pattern in C is given below.

#include <stdio.h>

int main() {

// size of cross, use odd number

int size = 5;

for (int i = 0; i < size; i++) {

for (int j = 0; j < size; j++) {

if (i==j || i+j==size-1) {

printf("\*");

} else {

printf(" ");

}

}

printf("\n");

}

return 0;

}

**Output:**

\* \*

\* \*

\*

\* \*

\* \*

**18. Heart pattern**

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

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

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

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

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

\*\*\*\*\*

\*\*\*

\*

The Heart star pattern in C is quite complex structure. The code to create this is given below.

#include <stdio.h>

int main() {

// heart star pattern

int size = 6;

for (int i = size / 2; i < size; i += 2) {

// print first spaces

for (int j = 1; j < size-i; j += 2) {

printf(" ");

}

// print first stars

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

printf("\*");

}

// print second spaces

for (int j = 1; j < size-i+1; j++) {

printf(" ");

}

// print second stars

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

printf("\*");

}

printf("\n");

}

// lower part

// inverted pyramid

for (int i = size; i > 0; i--) {

for (int j = 0; j < size-i; j++) {

printf(" ");

}

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

printf("\*");

}

printf("\n");

}

return 0;

}

**Output:**

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

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

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

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

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

\*\*\*\*\*

\*\*\*

\*

Above we have covered 18 different **star pattern in C** now let's look at some number patterns.

**19. Right triangle number pattern**

1

12

123

1234

12345

You can see above right triangle number pattern. The code to create this is given below.

#include <stdio.h>

int main() {

int size = 5;

// loop to print the pattern

for (int i = 0; i < size; i++) {

// print spaces

for (int j = 1; j < size-i; j++) {

printf(" ");

}

// print number

for (int k = 0; k <= i; k++) {

printf("%d", (k+1));

}

printf("\n");

}

return 0;

}

**Output:**

1

12

123

1234

12345

**20. Left triangle number pattern**

1

12

123

1234

12345

You can see above left triangle number pattern. The code to create this is given below.

#include <stdio.h>

int main() {

int size = 5;

// loop to print the pattern

for (int i = 0; i < size; i++) {

// print column

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

printf("%d", (j+1));

}

printf("\n");

}

return 0;

}

**Output:**

1

12

123

1234

12345

**21. Number pyramid pattern**

The number pyramid pattern has the shape of the pyramid that is made up of numbers.

1

123

12345

1234567

123456789

Here is [pyramid pattern programs using numbers using C](https://www.tutorialstonight.com/c-program-to-print-pyramid-pattern-of-numbers) in detail. The code to create this is given below.

#include <stdio.h>

int main() {

int size = 5;

for (int i = 0; i < size; i++) {

// print spaces

for (int j = 0; j < size-i-1; j++) {

printf(" ");

}

// print number

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

printf("%d", k+1);

}

printf("\n");

}

return 0;

}

**Output:**

1

123

12345

1234567

123456789

**22. Hollow number pyramid pattern**

The hollow number pyramid pattern is a normal number pyramid with numbers only at boundaries.

1

1 2

1 2

1 2

123456789

The code to create a hollow number pyramid pattern is given below.

#include <stdio.h>

int main() {

int size = 5;

for (int i = 0; i < size; i++) {

// print spaces

for (int j = 0; j < size-i-1; j++) {

printf(" ");

}

// print number

int num = 1;

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

if (i == 0 || i == size-1) {

printf("%d", num++);

} else {

if (k == 0 || k == 2\*i) {

printf("%d", num++);

} else {

printf(" ");

}

}

}

printf("\n");

}

return 0;

}

**Output:**

1

1 2

1 2

1 2

123456789

**23. Number pyramid reverse pattern**

The reverse number pyramid pattern is a number pyramid pattern rotated 180 degrees.

123456789

1234567

12345

123

1

The code to create a reverse number pyramid pattern in C is given below.

#include <stdio.h>

int main() {

int size = 5;

for (int i = 0; i < size; i++) {

// print spaces

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

printf(" ");

}

// print number

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

printf("%d", k+1);

}

printf("\n");

}

return 0;

}

**Output:**

123456789

1234567

12345

123

1

**24. Number diamond pattern**

The number diamond pattern is the same as the diamond pattern we have seen above but this one is just made of numbers.

1

123

12345

1234567

123456789

1234567

12345

123

1

The code to create a number diamond pattern in C is given below.

#include <stdio.h>

int main() {

int size = 5;

int num = 1;

// upside pyramid

for (int i = 1; i <= size; i++) {

// printing spaces

for (int j = size; j > i; j--) {

printf(" ");

}

// printing number

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

printf("%d", num++);

}

// set the number to 1

num = 1;

printf("\n");

}

// downside pyramid

for (int i = 1; i <= size-1; i++) {

// printing spaces

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

printf(" ");

}

// printing number

for (int k = (size-i)\*2-1; k > 0; k--) {

printf("%d", num++);

}

// set num to 1

num = 1;

printf("\n");

}

return 0;

}

**Output:**

1

123

12345

1234567

123456789

1234567

12345

123

1

**25. Hollow number diamond pattern**

The hollow number diamond pattern is hollow diamond pattern made up of numbers.

1

1 2

1 2

1 2

1 2

1 2

1 2

1 2

1

The code to create hollow number diamond pattern in C is given below.

#include <stdio.h>

int main() {

int size = 5, num = 1;

// upside pyramid

for (int i = 1; i <= size; i++) {

// printing spaces

for (int j = size; j > i; j--) {

printf(" ");

}

// printing number

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

if (k == 0 || k == 2\*i-2) {

printf("%d", num++);

} else {

printf(" ");

}

}

// set the number to 1

num = 1;

printf("\n");

}

// downside triangle

for (int i = 1; i < size; i++) {

// printing spaces

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

printf(" ");

}

// printing number

for (int k = (size-i)\*2-1; k >= 1; k--) {

if (k == 1 || k == (size-i)\*2-1) {

printf("%d", num++);

} else {

printf(" ");

}

}

// set the number to 1

num = 1;

printf("\n");

}

return 0;

}

**Output:**

1

1 2

1 2

1 2

1 2

1 2

1 2

1 2

1

LNow look at some alphabet pattern in C.

**26. Alphabet diamond pattern**

The alphabet diamond pattern is normal diamond pattern made up of alphabet.

A

ABC

ABCDE

ABCDEFG

ABCDEFGHI

ABCDEFG

ABCDE

ABC

A

To print character in C use printf("%c", ch) .The complete code for the alphabet diamond pattern program in C is given below.

#include <stdio.h>

int main() {

int size = 5;

int alpha = 65;

int num = 0;

// upside pyramid

for (int i = 1; i <= size; i++) {

// printing spaces

for (int j = size; j > i; j--) {

printf(" ");

}

// printing alphabets

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

printf("%c", alpha + num++);

}

// set the number to 0

num = 0;

printf("\n");

}

// downside pyramid

for (int i = 1; i <= size-1; i++) {

// printing spaces

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

printf(" ");

}

// printing alphabets

for (int k = (size-i)\*2-1; k > 0; k--) {

printf("%c", alpha + num++);

}

// set num to 0

num = 0;

printf("\n");

}

return 0;

}

**Output:**

A

ABC

ABCDE

ABCDEFG

ABCDEFGHI

ABCDEFG

ABCDE

ABC

A

**27. Hollow alphabet diamond pattern**

The hollow alphabet diamond pattern is shown below.

A

A B

A B

A B

A B

A B

A B

A B

A

The code to create hollow alphabet diamond pattern in C is given below.

#include <stdio.h>

int main() {

int size = 5;

int alpha = 65;

int num = 0;

// upside pyramid

for (int i = 1; i <= size; i++) {

// printing spaces

for (int j = size; j > i; j--) {

printf(" ");

}

// printing alphabets

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

if (k == 0 || k == 2\*i-2) {

printf("%c", alpha + num++);

} else {

printf(" ");

}

}

// set the number to 0

num = 0;

printf("\n");

}

// downside triangle

for (int i = 1; i < size; i++) {

// printing spaces

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

printf(" ");

}

// printing alphabets

for (int k = (size-i)\*2-1; k >= 1; k--) {

if (k == 1 || k == (size-i)\*2-1) {

printf("%c", alpha + num++);

} else {

printf(" ");

}

}

// set the number to 0

num = 0;

printf("\n");

}

return 0;

}

**Output:**

A

A B

A B

A B

A B

A B

A B

A B

A

**28. Alphabet pyramid pattern**

The alphabet pyramid pattern is a normal pyramid pattern made up of the alphabet.

A

ABC

ABCDE

ABCDEFG

ABCDEFGHI

The code to create alphabet pyramid pattern in C is given below.

#include <stdio.h>

int main() {

int size = 5, alpha = 65;

for (int i = 0; i < size; i++) {

// print spaces

for (int j = 0; j < size-i-1; j++) {

printf(" ");

}

// print alphabets

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

printf("%c", alpha+k);

}

printf("\n");

}

return 0;

}

**Output:**

A

ABC

ABCDE

ABCDEFG

ABCDEFGHI

**29. Hollow alphabet pyramid pattern**

Here is how hollow alphabet pyramid pattern looks like.

A

B C

D E

F G

HIJKLMNOP

The code to create hollow alphabet pyramid pattern in C is given below.

#include <stdio.h>

int main() {

int size = 5, alpha = 65, num = 0;

for (int i = 0; i < size; i++) {

// print spaces

for (int j = 0; j < size-i-1; j++) {

printf(" ");

}

// print alphabets

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

if (i == 0 || i == size - 1) {

printf("%c", alpha + num++);

} else {

if (k == 0 || k == 2\*i) {

printf("%c", alpha + num++);

} else {

printf(" ");

}

}

}

printf("\n");

}

return 0;

}

**Output:**

A

B C

D E

F G

HIJKLMNOP

**30. Reverse alphabet pyramid pattern**

The reverse alphabet pyramid pattern is alphabet pattern in reverse order.

ABCDEFGHI

ABCDEFG

ABCDE

ABC

A

The code to create reverse alphabet pyramid pattern in C is given below.

#include <stdio.h>

int main() {

// size of the square

int size = 5, alpha = 65;

for (int i = 0; i < size; i++) {

// print spaces

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

printf(" ");

}

// print alphabets

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

printf("%c", alpha + k);

}

printf("\n");

}

return 0;

}

**Output:**

ABCDEFGHI

ABCDEFG

ABCDE

ABC

A