GeeksforGeeks
A computer science portal for geeks

Custom Search

Courses

Login

Write an Article

Programs for printing pyramid patterns using recursion

This article is aimed at giving a recursive implementation for pattern printing.

1. **Simple triangle pattern:**

C++

```
// C++ code to demonstrate star pattern
#include <iostream>
using namespace std;

// function to print a row
void printn(int num)
{
    // base case
    if (num == 0)
        return;
    cout << "* ";

    // recursively calling printn()
    printn(num - 1);
}

// function to print the pattern
void pattern(int n, int i)
{
    // base case
    if (n == 0)
        return;
    printn(i);
    cout << endl;

    // recursively calling pattern()
    pattern(n - 1, i + 1);
}

// driver function
```

```
int main()
{
    int n = 5;
    pattern(n, 1);
    return 0;
}
```

Java

```
// Java code to demonstrate star pattern
import java.io.*;

class GFG
{

// function to print a row
static void printn(int num)
{
    // base case
    if (num == 0)
        return;
    System.out.print ("* ");

    // recursively calling printn()
    printn(num - 1);
}

// function to print the pattern
static void pattern(int n, int i)
{
    // base case
    if (n == 0)
        return;
    printn(i);
    System.out.println();

    // recursively calling pattern()
    pattern(n - 1, i + 1);
}

// Driver code
public static void main (String[] args)
{

    int n = 5;
    pattern(n, 1);
}
}

// This code is contributed by ajit.
```

Python3

```python
# Python 3 code to demonstrate star pattern

# function to print a row
def printn(num):

    # base case
    if (num == 0):
        return
    print("*", end = " ")

    # recursively calling printn()
    printn(num - 1)

# function to print the pattern
def pattern(n, i):

    # base case
    if (n == 0):
        return
    printn(i)
    print("\n", end = "")

    # recursively calling pattern()
    pattern(n - 1, i + 1)

# Driver Code
if __name__ == '__main__':
    n = 5
    pattern(n, 1)

# This code is contributed by
# Surendra_Gangwar
```

C#

```csharp
// C# code to demonstrate star pattern
using System;

class GFG
{

// function to print a row
static void printn(int num)
{
    // base case
    if (num == 0)
        return;
    Console.Write("* ");

    // recursively calling printn()
    printn(num - 1);
}

// function to print the pattern
static void pattern(int n, int i)
{
    // base case
    if (n == 0)
        return;
    printn(i);
    Console.WriteLine();

    // recursively calling pattern()
    pattern(n - 1, i + 1);
}

// Driver code
static public void Main ()
{
    int n = 5;
    pattern(n, 1);
}
}

// This code is contributed by akt_mit.
```

PHP

```
<?php
// PHP code to demonstrate star pattern

// Function to print a row
function printn($num)
{
    // base case
    if ($num == 0)
        return;
    echo "* ";

    // recursively calling printn()
    printn($num - 1);
}

// function to print the pattern
function pattern($n, $i)
{
    // base case
    if ($n == 0)
        return;
    printn($i);
    echo "\n";

    // recursively calling pattern()
    pattern($n - 1, $i + 1);
}

// Driver Code
$n = 5;
pattern($n, 1);
```

```
// This code is contributed by @Tushil..
?>
```

Output:

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

2. **After 180 degree rotation:**

C++

```
// C++ code to demonstrate star pattern

#include <iostream>
using namespace std;

// function to print spaces
void print_space(int space)
{
    // base case
    if (space == 0)
        return;
    cout << " "
         << " ";

    // recursively calling print_space()
    print_space(space - 1);
}

// function to print asterisks
void print_asterisk(int asterisk)
{
    // base case
    if (asterisk == 0)
        return;
    cout << "* ";

    // recursively calling print_asterisk()
    print_asterisk(asterisk - 1);
}

// function to print the pattern
void pattern(int n, int num)
{
    // base case
    if (n == 0)
        return;
    print_space(n - 1);
    print_asterisk(num - n + 1);
    cout << endl;

    // recursively calling pattern()
    pattern(n - 1, num);
}

// driver function
int main()
{
    int n = 5;
    pattern(n, n);
    return 0;
}
```

Java

```
// Java code to demonstrate star pattern
import java.util.*;

class GFG
{

    // function to print spaces
    static void print_space(int space)
    {
        // base case
        if (space == 0)
        {
            return;
        }
        System.out.print(" " + " ");

        // recursively calling print_space()
        print_space(space - 1);
    }

    // function to print asterisks
    static void print_asterisk(int asterisk)
    {
        // base case
        if (asterisk == 0)
        {
            return;
        }
        System.out.print("* ");

        // recursively calling print_asterisk()
        print_asterisk(asterisk - 1);
    }

    // function to print the pattern
    static void pattern(int n, int num)
    {
        // base case
        if (n == 0)
        {
            return;
        }
        print_space(n - 1);
        print_asterisk(num - n + 1);
        System.out.println();

        // recursively calling pattern()
        pattern(n - 1, num);
    }

    // Driver code
    public static void main(String[] args)
    {
        int n = 5;
        pattern(n, n);
    }
}

// This code is contributed by Rajput-Ji
```

C#

```
// C# code to demonstrate star pattern
using System;

class GFG
{

// function to print spaces
static void print_space(int space)
{
  // base case
  if (space == 0)
    return;
  Console.Write(" " + " ");

  // recursively calling print_space()
  print_space(space - 1);
}

// function to print asterisks
static void print_asterisk(int asterisk)
{
  // base case
  if (asterisk == 0)
    return;
  Console.Write("* ");

  // recursively calling print_asterisk()
  print_asterisk(asterisk - 1);
}

// function to print the pattern
static void pattern(int n, int num)
{
    // base case
    if (n == 0)
        return;
    print_space(n - 1);
    print_asterisk(num - n + 1);
    Console.WriteLine();

    // recursively calling pattern()
    pattern(n - 1, num);
}

// Driver code
public static void Main()
{
    int n = 5;
    pattern(n, n);
}
}

// This code is contributed by Akanksha Rai
```

PHP

```
<?php
// PHP code to demonstrate star pattern
// function to print spaces
function print_space($space)
{
    // base case
    if ($space == 0)
        return;
    echo " ",
            " ";

    // recursively calling print_space()
    print_space($space - 1);
}

// function to print asterisks
function print_asterisk($asterisk)
{
    // base case
    if ($asterisk == 0)
        return;
    echo "* ";

    // recursively calling print_asterisk()
    print_asterisk($asterisk - 1);
}

// function to print the pattern
function pattern($n, $num)
{
    // base case
    if ($n == 0)
        return;
    print_space($n - 1);
    print_asterisk(($num - $n) + 1);
    echo "\n";

    // recursively calling pattern()
    pattern($n - 1, $num);
}

// Driver Code
$n = 5;
pattern($n, $n);

// This code is contributed by @Tushil.
?>
```

Output:

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

3. **Printing Pyramid:**

C++

```
// C++ code to demonstrate star pattern

#include <iostream>
using namespace std;

// function to print spaces
void print_space(int space)
{
    // base case
    if (space == 0)
        return;
    cout << " ";

    // recursively calling print_space()
    print_space(space - 1);
}

// function to print asterisks
void print_asterisk(int asterisk)
{
    // base case
    if (asterisk == 0)
```

```
        return;
    cout << "* ";

    // recursively calling asterisk()
    print_asterisk(asterisk - 1);
}

// function to print the pattern
void pattern(int n, int num)
{
    // base case
    if (n == 0)
        return;
    print_space(n - 1);
    print_asterisk(num - n + 1);
    cout << endl;

    // recursively calling pattern()
    pattern(n - 1, num);
}

// driver function
int main()
{
    int n = 5;
    pattern(n, n);
    return 0;
}
```

Java

```
// Java code to demonstrate star pattern
import java.util.*;

class GFG
{

// function to print spaces
static void print_space(int space)
{
    // base case
    if (space == 0)
        return;
    System.out.print(" ");

    // recursively calling print_space()
    print_space(space - 1);
}

// function to print asterisks
static void print_asterisk(int asterisk)
{
    // base case
    if (asterisk == 0)
        return;
    System.out.print("* ");

    // recursively calling asterisk()
    print_asterisk(asterisk - 1);
}

// function to print the pattern
static void pattern(int n, int num)
{
    // base case
    if (n == 0)
        return;
    print_space(n - 1);
    print_asterisk(num - n + 1);
    System.out.println("");

    // recursively calling pattern()
    pattern(n - 1, num);
}

// Driver code
public static void main(String[] args)
{
    int n = 5;
    pattern(n, n);
}
}

// This code is contributed by 29AjayKumar
```

Python3

```
# Python3 code to demonstrate star pattern

# function to print spaces
def print_space(space):

    # base case
    if (space == 0):
        return;
    print(" ", end = "");

    # recursively calling print_space()
    print_space(space - 1);

# function to print asterisks
def print_asterisk(asterisk):

    # base case
    if(asterisk == 0):
        return;
    print("* ", end = "");

    # recursively calling asterisk()
    print_asterisk(asterisk - 1);
```

```
# function to print the pattern
def pattern(n, num):

    # base case
    if (n == 0):
        return;
    print_space(n - 1);
    print_asterisk(num - n + 1);
    print("");

    # recursively calling pattern()
    pattern(n - 1, num);

# Driver Code
n = 5;
pattern(n, n);

# This code contributed by PrinciRaj1992
```

C#

```
// C# code to demonstrate star pattern
using System;

class GFG
{

// function to print spaces
static void print_space(int space)
{
    // base case
    if (space == 0)
        return;
    Console.Write(" ");

    // recursively calling print_space()
    print_space(space - 1);
}

// function to print asterisks
static void print_asterisk(int asterisk)
{
    // base case
    if (asterisk == 0)
        return;
    Console.Write("* ");

    // recursively calling asterisk()
    print_asterisk(asterisk - 1);
}

// function to print the pattern
static void pattern(int n, int num)
{
    // base case
    if (n == 0)
        return;
    print_space(n - 1);
    print_asterisk(num - n + 1);
    Console.WriteLine("");

    // recursively calling pattern()
    pattern(n - 1, num);
}

// Driver code
public static void Main(String[] args)
{
    int n = 5;
    pattern(n, n);
}
}

// This code is contributed by 29AjayKumar
```

PHP

```
<?php
// PHP code to demonstrate star pattern

// function to print spaces
function print_space($space)
{
    // base case
    if ($space == 0)
        return;
    echo " ";

    // recursively calling print_space()
    print_space($space - 1);
}

// function to print asterisks
function print_asterisk($asterisk)
{
    // base case
    if ($asterisk == 0)
        return;
    echo "* ";

    // recursively calling asterisk()
    print_asterisk($asterisk - 1);
}

// function to print the pattern
function pattern($n, $num)
{
    // base case
    if ($n == 0)
```

```
        return;
    print_space($n - 1);
    print_asterisk($num - $n + 1);
    echo "\n";

    // recursively calling pattern()
    pattern($n - 1, $num);
}

// Driver code
    $n = 5;
    pattern($n, $n);

// This code is contributed by ajit.
?>
```

Output:

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

4. **Number Pattern**

C++

```
// C++ code to demonstrate printing pattern of numbers

#include <iostream>
using namespace std;

// function to print a row
void print_row(int no, int val)
{
    // base case
    if (no == 0)
        return;
    cout << val << " ";

    // recursively calling print_row()
    print_row(no - 1, val);
}
```

```
// function to print the pattern
void pattern(int n, int num)
{
    // base case
    if (n == 0)
        return;
    print_row(num - n + 1, num - n + 1);
    cout << endl;

    // recursively calling pattern()
    pattern(n - 1, num);
}
int main()
{
    int n = 5;
    pattern(n, n);
}
```

Java

```
// Java code to demonstrate printing
// pattern of numbers
class GFG
{

// function to print a row
static void print_row(int no, int val)
{
    // base case
    if (no == 0)
        return;
    System.out.print(val + " ");

    // recursively calling print_row()
    print_row(no - 1, val);
}

// function to print the pattern
static void pattern(int n, int num)
{
    // base case
    if (n == 0)
        return;
    print_row(num - n + 1, num - n + 1);
    System.out.println();

    // recursively calling pattern()
    pattern(n - 1, num);
}

// Driver Code
public static void main(String[] args)
{
    int n = 5;
    pattern(n, n);
}
}

// This code is contributed by Code_Mech.
```

Python3

```
# Python code to demonstrate printing
# pattern of numbers

# function to print a row
def print_row(no, val):

    # base case
    if (no == 0):
        return;
    print(val , end=" ");

    # recursively calling print_row()
    print_row(no - 1, val);

# function to print the pattern
def pattern(n, num):
    # base case
    if (n == 0):
        return;
    print_row(num - n + 1, num - n + 1);
    print("");

    # recursively calling pattern()
    pattern(n - 1, num);

# Driver Code
n = 5;
pattern(n, n);

# This code contributed by PrinciRaj1992
```

C#

```
// C# code to demonstrate printing
// pattern of numbers
using System;

class GFG
{

// function to print a row
static void print_row(int no, int val)
{
    // base case
    if (no == 0)
```

```
        return;
    Console.Write(val + " ");

    // recursively calling print_row()
    print_row(no - 1, val);
}

// function to print the pattern
static void pattern(int n, int num)
{
    // base case
    if (n == 0)
        return;
    print_row(num - n + 1, num - n + 1);
    Console.WriteLine();

    // recursively calling pattern()
    pattern(n - 1, num);
}

// Driver Code
public static void Main()
{
    int n = 5;
    pattern(n, n);
}
}

// This code is contributed by Akanksha Rai
```

PHP

```
<?php
// PHP code to demonstrate printing
// pattern of numbers

// function to print a row
function print_row($no, $val)
{
    // base case
    if ($no == 0)
        return;
    echo $val . " ";

    // recursively calling print_row()
    print_row($no - 1, $val);
}

// function to print the pattern
function pattern($n, $num)
{
    // base case
    if ($n == 0)
        return;
    print_row($num - $n + 1,
              $num - $n + 1);
    echo "\n";

    // recursively calling pattern()
    pattern($n - 1, $num);
}

// Driver Code
$n = 5;
pattern($n, $n);

// This code is contributed
// by Akanksha Rai
?>
```

Output:

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

5. **Numbers without re assigning**:

```
// C++ code to demonstrate printing pattern of numbers

#include <iostream>
using namespace std;

// function to print a row
int print_row(int ct, int num)
{
    // base case
    if (num == 0)
        return ct;
    cout << ct << "\t";

    // recursively calling print_row()
    print_row(ct + 1, num - 1);
}

// function to print the pattern
void pattern(int n, int count, int num)
{
    // base case
    if (n == 0)
        return;
    count = print_row(count, num);
    cout << endl;

    // recursively calling pattern()
    pattern(n - 1, count, num + 1);
}

// driver function
int main()
{
    int n = 5;
    pattern(n, 1, 1);
}
```

Output:

```
1
2    3
4    5    6
7    8    9    10
11    12    13    14    15
```

6. **Character Pattern:**

```
// C++ code to demonstrate printing pattern of alphabets

#include <iostream>
using namespace std;

// function to print a row
void print_row(int no, int val)
{
    // base case
    if (no == 0)
        return;
    cout << (char)(val + 64) << " ";

    // recursively calling print_row()
    print_row(no - 1, val);
}

// function to print the pattern
void pattern(int n, int num)
{
    // base case
    if (n == 0)
        return;
    print_row(num - n + 1, num - n + 1);
    cout << endl;

    // recursively calling pattern()
    pattern(n - 1, num);
}
int main()
{
    int n = 5;
    pattern(n, n);
}
```

Output:

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

7. **Continuous Character pattern:**

```
#include <iostream>
using namespace std;

// function to print a row
int print_row(int ct, int num)
{
    // base case
    if (num == 0)
        return ct;
    cout << (char)(ct + 64) << " ";

    // recursively calling print_row()
    print_row(ct + 1, num - 1);
}

// function to print the pattern
void pattern(int n, int count, int num)
{
    // base case
    if (n == 0)
        return;
    count = print_row(count, num);
    cout << endl;

    // recursively calling pattern()
    pattern(n - 1, count, num + 1);
}

// driver function
int main()
{
    int n = 5;
    pattern(n, 1, 1);
}
```

Output:

```
A
B C
D E F
G H I J
K L M N O
```

Recommended Posts:

Programs for printing pyramid patterns in C++

PHP programs for printing pyramid patterns

Programs for printing pyramid patterns in Python

Programs for printing pyramid patterns in Java

Programs to print Interesting Patterns

Print the following pyramid pattern

Program for volume of Pyramid

Program to print Even Odd Number Pyramid

Program to print pyramid pattern

Program to print pyramid pattern

Program to Print Pyramid Pattern using numbers

Program to print hollow pyramid and diamond pattern

Program to print a inverse pyramid character pattern

Print the pyramid pattern with given height and minimum number of stars

Print Patterns in PL/SQL

Yash_R
Check out this Author's contributed articles.

Improved By : jit_t, SURENDRA_GANGWAR, Akanksha_Rai, Code_Mech, Rajput-
more

Article Tags : C++ Programs Recursion School Programming pattern-printing

Practice Tags : pattern-printing Recursion

Be the First to upvote.

□ To-do □ Done

0

No votes yet.

Feedback/ Suggest Improvement Add Notes Improve Article

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments Share this post!

▲