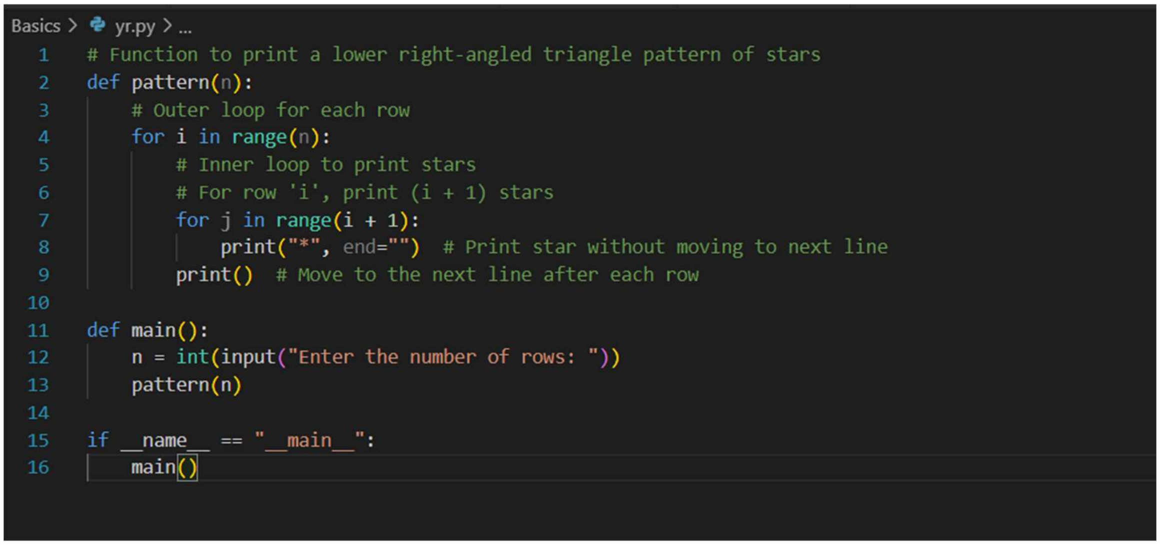
CELEBAL WEEK 1 ASSIGNMENT

Code for lower triangular:

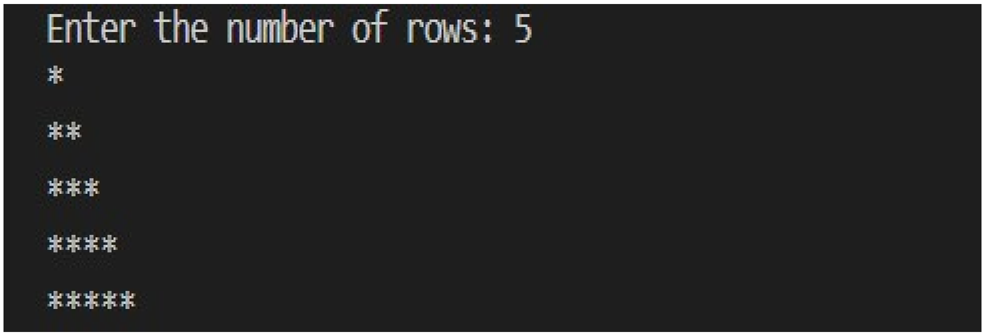


```python
# Function to print a lower right-angled triangle pattern of stars
def pattern(n):
    # Outer loop for each row
    for i in range(n):
        # Inner loop to print stars
        # For row 'i', print (i + 1) stars
        for j in range(i + 1):
            print("*", end="")  # Print star without moving to next line
        print()  # Move to the next line after each row

def main():
    n = int(input("Enter the number of rows: "))
    pattern(n)

if __name__ == "__main__":
    main()
```

Output:



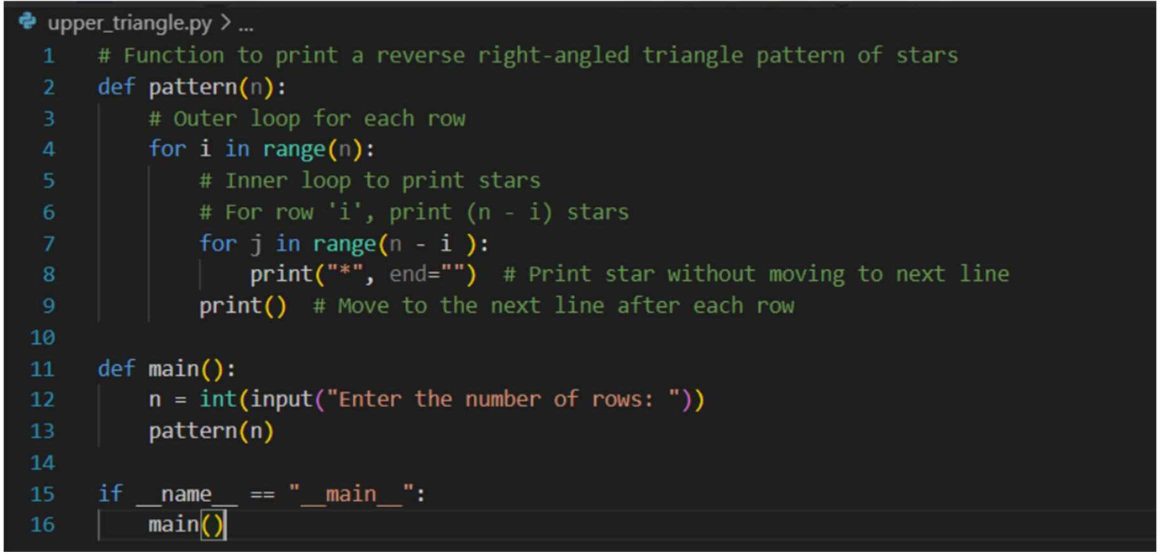
```
Enter the number of rows: 5
*
**
***
****
*****
```

Algorithm:

1. Input the number of rows n.
2. Loop from i = 0 to n - 1 (each i represents a row).
3. For each row i, loop from j = 0 to i:
   Print * on the same line.
4. After inner loop, move to the next line.
5. Repeat until all rows are printed.

Name: Devanshi Mittal

Code for upper triangular:



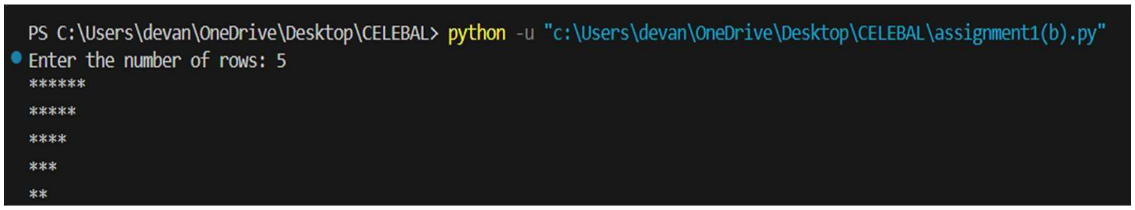
upper_triangle.py > ...

```python
# Function to print a reverse right-angled triangle pattern of stars
def pattern(n):
    # Outer loop for each row
    for i in range(n):
        # Inner loop to print stars
        # For row 'i', print (n - i) stars
        for j in range(n - i ):
            print("*", end="")  # Print star without moving to next line
        print()  # Move to the next line after each row

def main():
    n = int(input("Enter the number of rows: "))
    pattern(n)

if __name__ == "__main__":
    main()
```

Output:

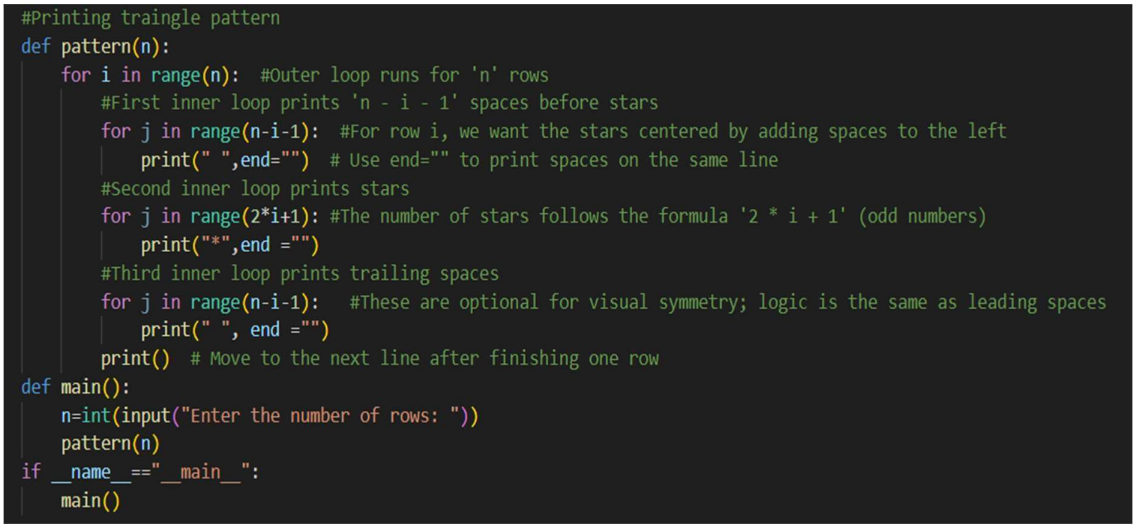


```
PS C:\Users\devan\OneDrive\Desktop\CELEBAL> python -u "c:\Users\devan\OneDrive\Desktop\CELEBAL\assignment1(b).py"
Enter the number of rows: 5
******
*****
****
***
**
```

Algorithm:

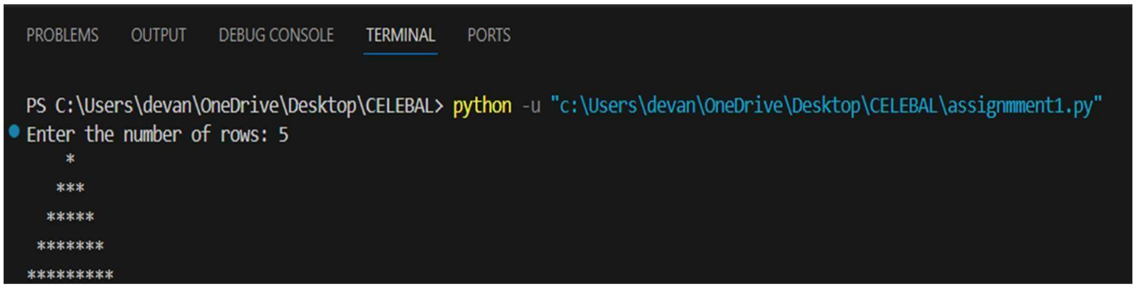
1. Start
2. Input an integer n (number of rows)
3. Repeat for each row i from 0 to n - 1:
   - Calculate number of stars as (n - i)
   - Print (n - i) stars on the same line
   - Move to the next line
4. End

Code of Pyramid:



```python
#Printing traingle pattern
def pattern(n):
    for i in range(n):  #Outer loop runs for 'n' rows
        #First inner loop prints 'n - i - 1' spaces before stars
        for j in range(n-i-1):  #For row i, we want the stars centered by adding spaces to the left
            print(" ",end="")  # Use end="" to print spaces on the same line
        #Second inner loop prints stars
        for j in range(2*i+1): #The number of stars follows the formula '2 * i + 1' (odd numbers)
            print("*",end ="")
        #Third inner loop prints trailing spaces
        for j in range(n-i-1):   #These are optional for visual symmetry; logic is the same as leading spaces
            print(" ", end ="")
        print()  # Move to the next line after finishing one row
def main():
    n=int(input("Enter the number of rows: "))
    pattern(n)
if __name__=="__main__":
    main()
```

Output:



```
PS C:\Users\devan\OneDrive\Desktop\CELEBAL> python -u "c:\Users\devan\OneDrive\Desktop\CELEBAL\assignmment1.py"
Enter the number of rows: 5
    *
   ***
  *****
 *******
*********
```

Algorithm:

```
1. Start
2. Read input n (number of rows)
3. For each row i from 0 to n - 1, do:
    a. Calculate number of leading spaces:
       spaces = n - i – 1
    b. Print spaces number of spaces (without newline)
    c. Calculate number of stars:
       stars = 2 * i + 1
    d. Print stars number of * characters (without newline)
    e. Print same number of trailing spaces as leading for symmetry
    f. Move to next line (print newline)
4. End For
5. End
```

Name: Devanshi Mittal