Pattern Printing

Note Book Owner: Emdadul Hoque

Problem 01: Print Character Pattern A B C D E F G H I J

In [8]:
```python
print("Print Character Pattern", end="")
char = 64
for i in range(5):
    for j in range(1, i+1):
        print(chr(char+1), end=" ")
        char+=1
    print()
```

```
Print Character Pattern
A
B C
D E F
G H I J
```

Problem 02: Hollow Rectangle Pattern ***** * * * * *****

In [59]:
```python
print("Hollow Rectangle pattern")

def hollow_rectangle(total_rows, total_colums):
    for i in range(1, total_rows+1):
        for j in range(1, total_colums+1):

            if i==1 or i==total_rows or j==1 or j==total_colums:

                print("*", end=" ")
            else:
                print(" ", end=" ")
        print()

hollow_rectangle(4,5)
```

```
Hollow Rectangle pattern
* * * * *
*       *
*       *
* * * * *
```

Problem 03: Inverted and Rotated half-pyramid * * * * * * * * * *

In [1]:
```python
print("Inverted and Rotated half-pyramid")

def inverted_roated_half_pyramid(total_rows, total_columns):
    for i in range(1, total_rows+1):
        n = total_columns-i

        for j in range(1, n+1):
            print(" ", end = " ")
        for j in range(1, i+1):
            print("*", end = " ")

        print()

inverted_roated_half_pyramid(7,7)
```

```
Inverted and Rotated half-pyramid
            *
          * *
        * * *
      * * * *
    * * * * *
  * * * * * *
* * * * * * *
```

In [36]:
```python
def print_pattern(n, m):
  for i in range(1, n+1):
    n = m - i
    print("  " *n, end="")
    print(" *" * i)

if __name__ == "__main__":
    print_pattern(6, 6)
```

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

Problem 04: Inverted half-pyramid with numbers 1 2 3 4 5 1 2 3 4 1 2 3 1 2 1

In [123…
```python
print("Inverted half-pyramid with numbers")

def inverted_half_pyramid_with_numbers(total_rows, total_columns):
    for i in range(1, total_rows+1):
        inner_loop_terminator = total_columns-i+1
        for j in range(1, inner_loop_terminator+1):
            print(j, end=" ")
        print()

inverted_half_pyramid_with_numbers(10,10)
```

```
Inverted half-pyramid with numbers
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

Problem 05: Floyd Triangle 1 2 3 4 5 6 7 8 9 10 11 12 13 15

In [183…
```python
print("Flyd triangle pattern print")
def floyd_triangle(total_rows):
    counter = 1
    for i in range(1, total_rows+1):
        for j in range(1, i+1):
            print(counter, end=" ")
            counter += 1
        print()
floyd_triangle(5)
```

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

Problem 06: 0-1 Triangle 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1

In [189…

```python
print("problem: 0-1 Triangle")

def zero_one_triangle(total_rows):
    for i in range(total_rows):
        for j in range(i+1):
            n = i + j
            if n%2 == 0:
                print("1", end=" ")
            else:
                print("0", end=" ")
        print()
zero_one_triangle(5)
```

```
problem: 0-1 Triangle
1
0 1
1 0 1
0 1 0 1
1 0 1 0 1
```

In [185…

```python
def print_triangle_pattern(n):
    for i in range(n):
        for j in range(i + 1):
            print((i + j) % 2, end=" ")
        print()

# Example usage
n = int(input("Enter the number of rows for the triangle pattern: "))
print_triangle_pattern(n)
```

```
Enter the number of rows for the triangle pattern: 5
0
1 0
0 1 0
1 0 1 0
0 1 0 1 0
```

Problem 07: Butterfly Pattern *

In [172…

```python
def butterfly_patters(total_rows):

    for i in range(1, total_rows+1):
        loop_terminator = 2*(total_rows-i)
        for j in range(1, i+1):
            print("*", end=" ")
        for j in range(1, loop_terminator+1):
            print(" ", end=" ")
        for j in range(1, i+1):
            print("*", end=" ")
        print()


    for i in range(total_rows, 0, -1):
        loop_terminator = 2*(total_rows-i)
        for j in range(1, i+1):
            print("*", end=" ")
        for j in range(1, loop_terminator+1):
            print(" ", end=" ")
        for j in range(1, i+1):
            print("*", end=" ")
        print()

butterfly_patters(5)
```

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

In [174…

```python
def print_butterfly_pattern(rows):
    # Upper half of the butterfly pattern
    for i in range(1, rows + 1):
        for j in range(1, 2 * rows + 1):
            if j <= i or j > 2 * rows - i:
                print("*", end="")
            else:
                print(" ", end="")
        print()

    # Lower half of the butterfly pattern
    for i in range(rows, 0, -1):
        for j in range(1, 2 * rows + 1):
            if j <= i or j > 2 * rows - i:
                print("*", end="")
            else:
                print(" ", end="")
        print()

# Test the function
num_rows = int(input("Enter the number of rows for the butterfly pattern: "))
print_butterfly_pattern(num_rows)
```

```
Enter the number of rows for the butterfly pattern: 5
*        *
**      **
***    ***
****  ****
**********
**********
****  ****
***    ***
**      **
*        *
```

In [176…

```python
def butterfly_pattern(n):
  for i in range(1, n + 1):
    print("*" * i, end="")
    print(" " * (n - i) * 2, end="")
    print("*" * i)
  for i in range(n, 0, -1):
    print("*" * i, end="")
    print(" " * (n - i) * 2, end="")
    print("*" * i)

if __name__ == "__main__":
  n = 6
  butterfly_pattern(n)
```

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

In [22]:
```python
def butterfly(n):
    for i in range(1, n+1):
        print(" * " * i, end="")
        print("   "* 2*(n - i), end="")
        print(" * " * i)
    for i in range(n, 0, -1):
        print(" * " * i, end="")
        print("   "* 2*(n - i), end="")
        print(" * " * i)


if __name__ == "__main__":
    butterfly(5)
```

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

problem 08: rombus pattern *

In [23]:
```python
def rombus(n):
    for i in range(1, n+1):
        space = n - i
        print("  " * space, end=" ")
        print(" *" * 5)
rombus(5)
```

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

Problem 9: Hollow rombus * * * * * * * * * * * * * * * *

In [24]:
```python
def hollow_rombus(n):
    for i in range(1, n+1):
        space = n - i
        print("  " * space, end=" ")

        for j in range(1, n+1):
            if i==1 or i == n or j==1 or j==n:

                print("*", end=" ")
            else:
                print("  ", end="")
        print()
```

```
hollow_rombus(5)
```

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

Problem 10: Dimond Pattern * *** ***** ******* ******* ***** *** *

```
In [25]: def dimond_pattern(n):

             for i in range(1, n+1):
                 star = 2*i - 1
                 space = n - i
                 print(" " * space, end=" ")
                 print("*" * star)
             for i in range(n, 0, -1):
                 star = 2*i - 1
                 space = n - i
                 print(" " * space, end=" ")
                 print("*" * star)

         dimond_pattern(4)
```

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

Problem 11: Hollow inverted Half pyramid pattern *

```
In [97]: print("Hollow Inverted Half Pyramid")

         def hollow_inverted_half_pyramid(n):
             m = n+1
             for i in range(1, n+1):
                 m -= 1
                 for j in range(1, n+1):
                     if i==1 or j == 1 or j == m:
                         print("*", end=" ")

                     else:
                         print(" ", end=" ")
                 print()


         hollow_inverted_half_pyramid(8)
```

```
Hollow Inverted Half Pyramid
* * * * * * * *
*           *
*         *
*       *
*     *
*   *
* *
*
```

```
In [96]: def hollow_inverted_half_pyramid(n):
             for i in range(n, 0, -1):
                 for j in range(i):
                     if i==n or j == 0 or j == i - 1:
```

```
                print("*", end=" ")
            else:
                print(" ", end=" ")
        print()

if __name__ == "__main__":
    n = 5
    hollow_inverted_half_pyramid(n)
```

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

Problem 12: Hollow full pyramid

In [139…
```
def number_pyramid(n):
    for i in range(1, n+1):
        print(" "*(n-i), end=" ")
        print((str(i)+" ")*i,)
number_pyramid(7)
```

```
       1
      2 2
     3 3 3
    4 4 4 4
   5 5 5 5 5
  6 6 6 6 6 6
 7 7 7 7 7 7 7
```

In [141…
```
str(7)+" "
```

Out[141…
```
'7 '
```

Problem 13: Palindromic pattern with numbers

In [151…
```
def palindormic_pattern(n):
    for i in range(1, n+1):
        print(" "*(n-i), end="")
        for j in range(i,0,-1):
            print(j, end="")
        for k in range(2, i+1):
            print(k,end="" )
        print()
palindormic_pattern(5)
```

```
    1
   212
  32123
 4321234
543212345
```

Problem 14:

In [155…
```
def pattern_printing(n):
    for i in range(1, n+1):
        for j in range(1, i+1):
            print(j, end="")
        print()
    for i in range(n-1,0,-1):
        for j in range(1, i+1):
            print(j, end="")
        print()
pattern_printing(4)
```

```
1
12
123
1234
123
12
1
```

```
In [ ]:
```