

# Patterns

## # 1. Solid Square Pattern

```
n = 5
for i in range(n):
    for j in range(n):
        print("*",end=" ")
    print()
```

## # 2. Solid Rectangle

```
n, m = 3, 5
for i in range(n):
    for j in range(m):
        print("*",end=" ")
    print()
```

## # 3. Right-Angled Triangle (Left-Aligned)

```
for i in range(1,6):
    print("*"*i)
```

## # 4. Right-Angled Triangle (Right - Aligned)

```
n = 5
for i in range(1,n+1):
    for j in range(n-i):
        print(" ", end=" ")
    for j in range(i):
        print("*", end=" ")
    print()
```

## # 5. Inverted Right-Angled Triangle (Left-Aligned)

```
for i in range(5,0,-1):
    print("*"*i)
```

## # 6. Inverted Right-Angled Triangle (Right - Aligned)

```
n = 5
for i in range(n):
    for j in range(i):
        print(" ", end=" ")
    for j in range(n-i):
        print("*", end=" ")
    print()
```

### # 7. Centered Pyramid

n = 5

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

### # 8. Diamond

n = 4

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

### # 9. Butterfly pattern

n = 5

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

### # 10. Half Diamond Left aligned

```
n = 5
for i in range(n):
    for j in range(i):
        print("*", end=" ")
    print()

for i in range(n):
    for j in range(n-i):
        print("*", end=" ")
    print()
```

### # 11. Half Diamond Right aligned

```
n = 5
for i in range(n):
    for j in range(n-i):
        print(" ", end=" ")
    for j in range(i):
        print("*", end=" ")
    print()

for i in range(n):
    for j in range(i):
        print(" ", end=" ")
    for j in range(n-i):
        print("*", end=" ")
    print()
```

### # 12. Sand glass pattern

```
n = 5
for i in range(n):
    for j in range(i):
        print(" ", end=" ")
    for j in range(n-i):
        print("*", end=" ")
    print()

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

### # 13. Increasing Width Triangle

```
for i in range(1,6):
    for j in range(i):
        print("*",end=" ")
    print()
```

### # 14. Decreasing Width Triangle

```
n=5
for i in range(n):
    for j in range(n-i):
        print("*",end=" ")
    print()
```

### # 15. Right Aligned Hill Pattern

```
n=5
for i in range(1,n+1):
    for j in range(n-i):
        print(" ",end=" ")
    for j in range(i):
        print("*",end=" ")
    print()
```

## Hollow Patterns

### # 16. Hollow square

```
n = 5
for i in range(n):
    for j in range(n):
        if i==0 or j==0 or i==n-1 or j==n-1:
            print("*",end=" ")
        else:
            print(" ",end=" ")
    print()
```

### # 17. Hollow rectangle

```
for i in range(1,5):
    for j in range(1,8):
        if i==1 or i==4 or j==1 or j==7:
            print("*",end=" ")
        else:
            print(" ",end=" ")
    print()
```

### # 18. Hollow Right angle Triangle Left Aligned

```
for i in range(1,6):
    for j in range(1,i+1):
        if i== 5 or j==1 or j==i:
            print("*",end=" ")
        else:
            print(" ",end=" ")
    print()
```

### # 19. Hollow Right angle Triangle Right aligned

```
n = 5
for i in range(1,n+1):
    for j in range(n-i):
        print(" ",end=" ")
    for j in range(1,i+1):
        if i==n or j==i or j==1:
            print("*",end=" ")
        else:
            print(" ",end=" ")
    print()
```

### # 20. Hollow Right angle Triangle Inverted Left Aligned

```
n = 5
for i in range(n,0,-1):
    for j in range(1,i+1):
        if i==n or j==1 or i==j:
            print("*",end=" ")
        else:
            print(" ",end=" ")
    print()
```

### # 21. Hollow Right angle Triangle Inverted Right Aligned

```
n = 5
for i in range(n):
    for j in range(i):
        print(" ",end=" ")
    for j in range(n-i):
        if i==0 or j==0 or j==(n-i-1):
            print("*",end=" ")
        else:
            print(" ",end=" ")
    print()
```

## # 22. Hollow pyramid

```
n = 5
for i in range(1,n+1):
    for j in range(n-i):
        print(" ",end=" ")
    for j in range(2*i-1):
        if i==n or j==0 or j==(2*i-2):
            print("*",end=" ")
        else:
            print(" ",end=" ")
    print()
```

## # 23. Hollow Diamond Pattern

```
n = 4
for i in range(n):
    for j in range(n-i-1):
        print(" ", end=" ")
    for j in range(2*i+1):
        if j==0 or j==(2*i):
            print("*", end=" ")
        else:
            print(" ", end=" ")
    print()
for i in range(1,n+1):
    for j in range(i):
        print(" ",end=" ")
    for j in range(2*(n-i)-1):
        if j==0 or j==(2*(n-i)-2):
            print("*",end=" ")
        else:
            print(" ",end=" ")
    print()
```

## # 24. Hollow butterfly

```
n = 5
for i in range(n,0,-1):
    s = " "*(n-i)
    if i == 1:
        print("*"+s+"*"(i)+s+"*")
    else:
        print("*"+s+"* "+" "(i-2)+"*"+s+"*")
for i in range(2,n+1):
    s = " "*(n-i)
    print("*"+s+"* "+" "(i-2)+"*"+s+"*")
print()
```

### # 25. Hollow Hour Glass Pattern

```
n = 5
for i in range(n,0,-1):
    print(" "*(n-i)+"*"(2*i-1))

for i in range(2,n+1):
    print(" "*(n-i)+"*"(2*i-1))
```

### # 26. Increasing number triangle

```
n = 5
for i in range(1,n+1):
    for j in range(1,i+1):
        print(j,end=" ")
    print()
```

### # 27. Repeating row number triangle

```
n = 5
for i in range(1,n+1):
    for j in range(1,i+1):
        print(i,end=" ")
    print()
```

### # 28. Continous Number Triangle

```
rows = 4
num = 1
for i in range(1,rows+1):
    for j in range(1,i+1):
        print(num, end=" ")
        num += 1
    print()
```

### # 29. Reverse Row Number Triangle

```
n = 5
for i in range(n):
    for j in range(i+1,0,-1):
        print(j,end=" ")
    print()
```

### # 30. Inverted Number Triangle

```
n = 5
for i in range(n,0,-1):
    for j in range(i,0,-1):
        print(j,end=" ")
    print()
```

### # 31. Right Aligned Number Triangle

```
n = 5
for i in range(1,n+1):
    for j in range(n-i+1):
        print(" ", end=" ")
    for j in range(1,i+1):
        print(j,end=" ")
    print()
```

### # 32. Centered Number Pyramid

```
n = 4
for i in range(1,n+1):
    print(" "*(n-i),end="")

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

    for j in range(i-1,0,-1):
        print(j,end="")
    print()
```

### # 33. Even Number Right Angle triangle

```
n = 5
for i in range(1,n+1):
    for j in range(1,i+1):
        print(j*2, end=" ")
    print()
```

### # 34. Odd Number Right Angle triangle

```
n = 5
for i in range(1,n+1):
    for j in range(1,i+1):
        print(2*j-1,end=" ")
    print()
```

### # 35. Pascals Triangle

```
rows = 5
for i in range(rows):
    print(" "*(rows-i-1),end=" ")

    num = 1
    for j in range(i+1):
        print(num,end=" ")
        num = num*(i-j) // (j+1)
    print()
```



# Matrix Problems

## # 1. Check if Matrix is square

```
def isSquare(matrix):  
    if not matrix:  
        return False  
  
    rows = len(matrix)  
    cols = len(matrix[0])  
  
    for row in matrix:  
        if len(row) != cols:  
            return False  
    return rows == cols
```

```
matrix = [[1,2],[3,4]]  
print(isSquare(matrix))
```

## # 2. Print Diagonal Elements

```
def diagonal(matrix):  
    n = len(matrix)  
    for i in range(n):  
        print(matrix[i][i], end=" ")
```

```
matrix = [[1,2],[3,4]]  
diagonal(matrix)
```

## # 3. Print Anti - Diagonal Elements

```
def antiDiagonal(matrix):  
    n = len(matrix)  
    for i in range(n):  
        j = n-1-i  
        print(matrix[i][j],end=" ")
```

```
matrix = [[1,2],[3,4]]  
antiDiagonal(matrix)
```

## # 4. Print Non - Diagonal Elements

```
def nonDiagonal(matrix):  
    n = len(matrix)  
    for i in range(n):  
        for j in range(len(matrix[i])):  
            if i != j:  
                print(matrix[i][j],end=" ")
```

```
matrix = [[1,2],[3,4]]  
nonDiagonal(matrix)
```

### # 5. Print Non anti diagonal elements

```
def non_anti_diagonal(matrix):  
    n = len(matrix)  
  
    for i in range(n):  
        for j in range(len(matrix[i])):  
            if i + j != n - 1:  
                print(matrix[i][j],end=" ")  
  
matrix = [[1,2],[3,4]]  
non_anti_diagonal(matrix)
```

### # 6. Lower Triangle of Matrix

```
def lower_triangle(matrix):  
    n = len(matrix)  
    result = []  
    for i in range(n):  
        row = []  
        for j in range(n):  
            if i >= j:  
                row.append(matrix[i][j])  
            else:  
                row.append(0)  
        result.append(row)  
    return result  
  
A = [[1, 2], [3, 4]]  
print(lower_triangle(A))
```

### # 7. Upper Triangle of Matrix

```
def upper_triangle(matrix):  
    n = len(matrix)  
    result = []  
    for i in range(n):  
        row = []  
        for j in range(n):  
            if i <= j:  
                row.append(matrix[i][j])  
            else:  
                row.append(0)  
        result.append(row)  
    return result  
  
A = [[1, 2], [3, 4]]  
print( upper_triangle(A))
```

### # 8. Transpose of Matrix

```
def transpose(matrix):
    n = len(matrix)
    result = []
    for i in range(n):
        row = []
        for j in range(n):
            row.append(matrix[j][i])
        result.append(row)
    return result
A = [[1, 2], [3, 4]]
print(transpose(A))
```

### # 9. Check if Diagonal Elements are Same

```
def check_diagonal_same(matrix):
    n = len(matrix)
    value = matrix[0][0]
    for i in range(n):
        if matrix[i][i] != value:
            return False
    return True
A = [[1, 2], [3, 4]]
print(check_diagonal_same([[5,0],[0,5]]))
```

### # 10. Check if Anti-Diagonal Elements are Same

```
def check_antidiagonal_same(matrix):
    n = len(matrix)
    value = matrix[0][n - 1]
    for i in range(n):
        if matrix[i][n - 1 - i] != value:
            return False
    return True
A = [[1, 2], [3, 4]]
print(check_antidiagonal_same([[0,3],[3,0]]))
```

### # 11. Convert Diagonal Elements to Zero

```
def diagonal_to_zero(matrix):
    n = len(matrix)
    result = []
    for i in range(n):
        row = []
        for j in range(n):
            if i == j:
                row.append(0)
            else:
                row.append(matrix[i][j])
        result.append(row)
    return result
A = [[1, 2], [3, 4]]
print(diagonal_to_zero(A))
```

### # 12. Convert Anti-Diagonal Elements to Zero

```
def antidiagonal_to_zero(matrix):
    n = len(matrix)
    result = []
    for i in range(n):
        row = []
        for j in range(n):
            if i + j == n - 1:
                row.append(0)
            else:
                row.append(matrix[i][j])
        result.append(row)
    return result
A = [[1, 2], [3, 4]]
print(antidiagonal_to_zero(A))
```

### # 13. Convert Non-Diagonal Elements to Zero

```
def non_diagonal_to_zero(matrix):
    n = len(matrix)
    result = []
    for i in range(n):
        row = []
        for j in range(n):
            if i == j:
                row.append(matrix[i][j])
            else:
                row.append(0)
        result.append(row)
    return result
A = [[1, 2], [3, 4]]
print(non_diagonal_to_zero(A))
```

#### # 14. Sum of All Elements in Matrix

```
def sum_matrix(matrix):
    total = 0
    for row in matrix:
        for val in row:
            total += val
    return total
A = [[1, 2], [3, 4]]
print(sum_matrix(A))
```

#### # 15. Matrix Multiplication

```
def multiply_matrices(A, B):
    rows_A, cols_A = len(A), len(A[0])
    rows_B, cols_B = len(B), len(B[0])
    if cols_A != rows_B:
        raise ValueError("Incompatible dimensions for multiplication")

    result = []
    for i in range(rows_A):
        row = []
        for j in range(cols_B):
            s = 0
            for k in range(cols_A):
                s += A[i][k] * B[k][j]
            row.append(s)
        result.append(row)
    return result

A = [[1, 2], [3, 4]]
B = [[5, 6], [7, 8]]
print(multiply_matrices(A, B))
```