

## 1. Python program to add two Matrices

```
Y = [[5,8,1],
      [6,7,3],
      [4,5,9]]

result = [[0,0,0],
          [0,0,0],
          [0,0,0]]

# iterate through rows
for i in range(len(X)):
    # iterate through columns
    for j in range(len(X[0])):
        result[i][j] = X[i][j] + Y[i][j]

for r in result:
    print(r)
```

## 2. Python program to multiply two matrices

```
X = [[12,7,3],
      [4,5,6],
      [7,8,9]]

A = [[12, 7, 3],
      [4, 5, 6],
      [7, 8, 9]]

# take a 3x4 matrix
B = [[5, 8, 1, 2],
      [6, 7, 3, 0],
      [4, 5, 9, 1]]

result = [[0, 0, 0, 0],
          [0, 0, 0, 0],
          [0, 0, 0, 0]]

# iterating by row of A
for i in range(len(A)):

    # iterating by column by B
    for j in range(len(B[0])):
```

```

        # iterating by rows of B
        for k in range(len(B)):
            result[i][j] += A[i][k] * B[k][j]

for r in result:
    print(r)

```

### 3. Python program for Matrix Product

```

X = [[12,7,3],
     [4 ,5,6],
     [7 ,8,9]]

```

```

A = [[12, 7, 3],
     [4, 5, 6],
     [7, 8, 9]]

```

# take a 3x4 matrix

```

B = [[5, 8, 1, 2],
     [6, 7, 3, 0],
     [4, 5, 9, 1]]

```

```

result = [[0, 0, 0, 0],
          [0, 0, 0, 0],
          [0, 0, 0, 0]]

```

```

# iterating by row of A
for i in range(len(A)):

```

```

    # iterating by column by B
    for j in range(len(B[0])):

```

```

        # iterating by rows of B
        for k in range(len(B)):
            result[i][j] += A[i][k] * B[k][j]

```

```

for r in result:
    print(r)

```

#### 4. Adding and Subtracting Matrices in Python

```
import numpy as np

# creating first matrix
A = np.array([[1, 2], [3, 4]])

# creating second matrix
B = np.array([[4, 5], [6, 7]])

print("Printing elements of first matrix")
print(A)
print("Printing elements of second matrix")
print(B)

# adding two matrix
print("Addition of two matrix")
print(np.add(A, B))
```

#### 5. Transpose a matrix in Single line in Python

```
m = [[1,2],[3,4],[5,6]]
for row in m :
    print(row)
rez = [[m[j][i] for j in range(len(m))] for i in range(len(m[0]))]
print("\n")
for row in rez:
    print(row)
```

#### 6. Python | Matrix creation of n\*n

```
N = 4

# printing dimension
print("The dimension : " + str(N))

# using list comprehension
# matrix creation of n * n
res = [list(range(1 + N * i, 1 + N * (i + 1)))
        for i in range(N)]
```

```
# print result
print("The created matrix of N * N: " + str(res))
```

## 7. Python | Get Kth Column of Matrix

```
test_list = [[4, 5, 6], [8, 1, 10], [7, 12, 5]]
```

```
# printing original list
print("The original list is : " + str(test_list))
```

```
# initialize K
K = 2
res = [sub[K] for sub in test_list]
```

```
# printing result
print("The Kth column of matrix is : " + str(res))
```

## 8. Python – Vertical Concatenation in Matrix

```
test_list = [["Gfg", "good"], ["is", "for"], ["Best"]]
```

```
# printing original list
print("The original list : " + str(test_list))
```

```
# using loop for iteration
res = []
N = 0
while N != len(test_list):
    temp = ""
    for idx in test_list:
        # checking for valid index / column
        try: temp = temp + idx[N]
        except IndexError: pass
    res.append(temp)
    N = N + 1
```

```
res = [ele for ele in res if ele]
```

```
# printing result
print("List after column Concatenation : " + str(res))
```

**9. Python program to check if a string is palindrome or not**

```
def isPalindrome(s):
    return s == s[::-1]
```

```
# Driver code
s = "malayalam"
ans = isPalindrome(s)
```

```
if ans:
    print("Yes")
else:
    print("No")
```

**10. Python program to check whether the string is Symmetrical or Palindrome**

```
def palindrome(a):
    mid = (len(a)-1)//2
    start = 0
    last = len(a)-1
    flag = 0

    # A loop till the mid of the
    # string
    while(start <= mid):

        # comparing letters from right
        # from the letters from left
        if (a[start]== a[last]):
```

```
start += 1
```

```
last -= 1
```

```
else:
```

```
flag = 1
```

```
break;
```

```
# Checking the flag variable to
```

```
# check if the string is palindrome
```

```
# or not
```

```
if flag == 0:
```

```
    print("The entered string is palindrome")
```

```
else:
```

```
    print("The entered string is not palindrome")
```

```
def symmetry(a):
```

```
    n = len(a)
```

```
    flag = 0
```

```
# Check if the string's length
```

```
# is odd or even
```

```
if n%2:
```

```
    mid = n//2 + 1
```

```
else:
```

```
    mid = n//2
```

```
start1 = 0
```

```
start2 = mid
```

```
while(start1 < mid and start2 < n):
```

```
    if (a[start1]== a[start2]):
```

```
        start1 = start1 + 1
```

```
        start2 = start2 + 1
```

```
    else:
```

```
        flag = 1
```

```
        break
```

```
# Checking the flag variable to
```

```
# check if the string is symmetrical
```

```
# or not
```

```
if flag == 0:
```

```
    print("The entered string is symmetrical")
```

```
else:
```

```
    print("The entered string is not symmetrical")
```

```
string = 'amaama'
```

```
palindrome(string)
```

```
symmetry(string)
```