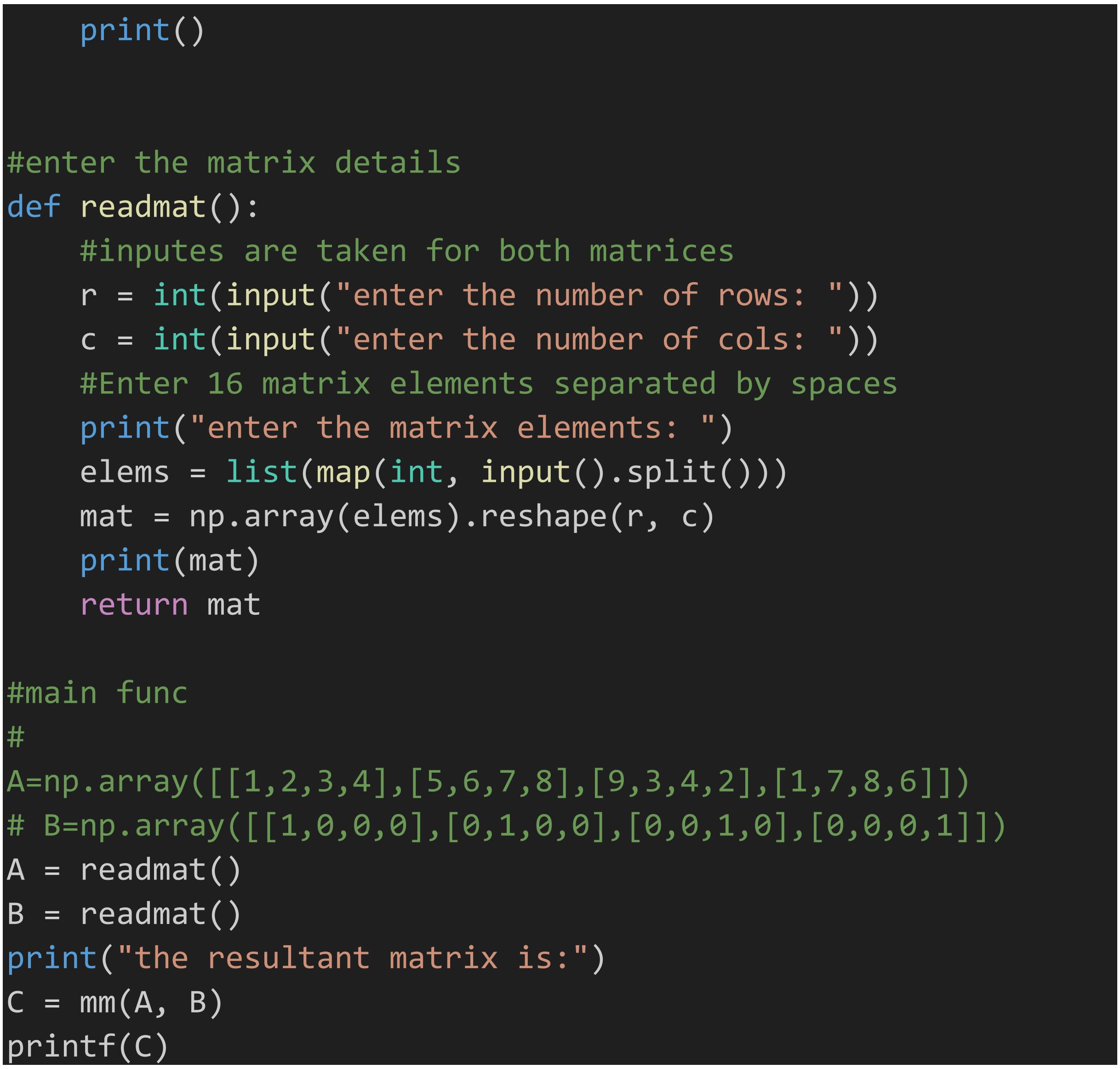
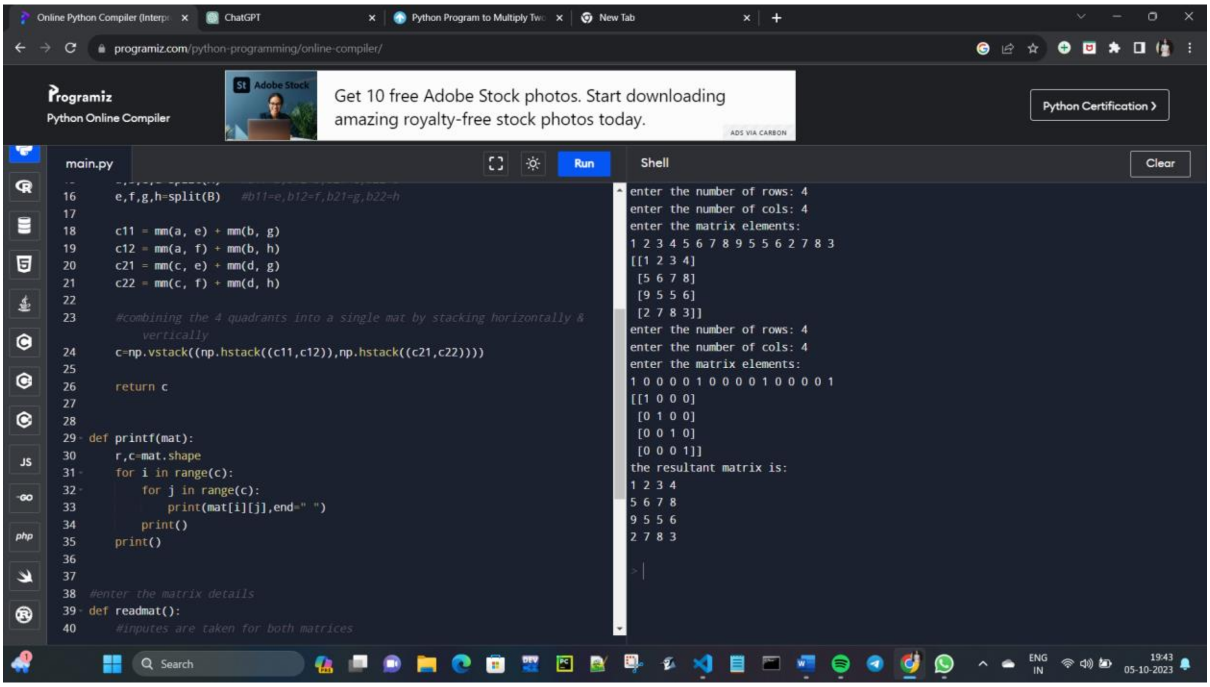
```
#matrix mul for large dim
import numpy as np

def split(mat):
    r,c=mat.shape
    r2,c2=r//2,c//2
    return
mat[:r2,:c2],mat[:r2,c2:],mat[r2:,:c2],mat[r2:,c2:]

#base case when size of mat is 1X1
def mm(A,B):
    if len(A)==1:
        return A*B

    #splitting the matrices into quadrants(each of 4
matrices),this will be donerecursively untill the base case
is reached
    a,b,c,d=split(A)   #a11=a,a12=b,a21=c,a22=d
    e,f,g,h=split(B)   #b11=e,b12=f,b21=g,b22=h

    c11 = mm(a, e) + mm(b, g)
    c12 = mm(a, f) + mm(b, h)
    c21 = mm(c, e) + mm(d, g)
    c22 = mm(c, f) + mm(d, h)

    #combining the 4 quadrants into a single mat by stacking
horizontally & vertically
    c=np.vstack((np.hstack((c11,c12)),np.hstack((c21,c22))))

    return c


def printf(mat):
    r,c=mat.shape
    for i in range(c):
        for j in range(c):
            print(mat[i][j],end=" ")
        print()
```



```
    print()


#enter the matrix details
def readmat():
    #inputes are taken for both matrices
    r = int(input("enter the number of rows: "))
    c = int(input("enter the number of cols: "))
    #Enter 16 matrix elements separated by spaces
    print("enter the matrix elements: ")
    elems = list(map(int, input().split()))
    mat = np.array(elems).reshape(r, c)
    print(mat)
    return mat

#main func
#
A=np.array([[1,2,3,4],[5,6,7,8],[9,3,4,2],[1,7,8,6]])
# B=np.array([[1,0,0,0],[0,1,0,0],[0,0,1,0],[0,0,0,1]])
A = readmat()
B = readmat()
print("the resultant matrix is:")
C = mm(A, B)
printf(C)
```


Online Python Compiler (Interp
ChatGPT
Python Program to Multiply Two
New Tab
programiz.com/python-programming/online-compiler/
Programiz
Python Online Compiler
Get 10 free Adobe Stock photos. Start downloading amazing royalty-free stock photos today.
ADS VIA CARBON
Python Certification ›
main.py
Run
Shell
Clear
Search
ENG
IN
19:43
05-10-2023

```python
    e,f,g,h=split(B)   #b11=e,b12=f,b21=g,b22=h

    c11 = mm(a, e) + mm(b, g)
    c12 = mm(a, f) + mm(b, h)
    c21 = mm(c, e) + mm(d, g)
    c22 = mm(c, f) + mm(d, h)

    #combining the 4 quadrants into a single mat by stacking horizontally &
        vertically
    c=np.vstack((np.hstack((c11,c12)),np.hstack((c21,c22))))

    return c


def printf(mat):
    r,c=mat.shape
    for i in range(c):
        for j in range(c):
            print(mat[i][j],end=" ")
        print()
    print()


#enter the matrix details
def readmat():
    #inputes are taken for both matrices
```

```
enter the number of rows: 4
enter the number of cols: 4
enter the matrix elements:
1 2 3 4 5 6 7 8 9 5 5 6 2 7 8 3
[[1 2 3 4]
 [5 6 7 8]
 [9 5 5 6]
 [2 7 8 3]]
enter the number of rows: 4
enter the number of cols: 4
enter the matrix elements:
1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1
[[1 0 0 0]
 [0 1 0 0]
 [0 0 1 0]
 [0 0 0 1]]
the resultant matrix is:
1 2 3 4
5 6 7 8
9 5 5 6
2 7 8 3

> 
```