Name:-Devansh Koyani

Er no:-22162171007

Batch:-54

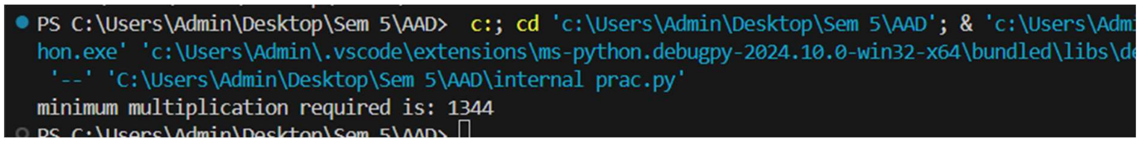
Internal Practical

Given a sequence of matrices/dimension is (4, 10, 3, 12, 20, and 7), we want to find the most efficient way to multiply these matrices together to obtain the minimum number of multiplications. The problem is not actually to perform the multiplication of the matrices but to obtain the minimum number of multiplications. We have many options because matrix multiplication is an associative operation, meaning that the order in which we multiply does not matter. The optimal order depends only on the dimensions of the matrices.
The brute-force algorithm is to considers all possible orders and takes the minimum. This is a very inefficient method. Implement the minimum multiplication algorithm using dynamic programming.

Code:-

```
def mul(p):

  n = len(p) - 1

  m = [[0] * n for _ in range(n)]


  for l in range(2, n + 1):

    for i in range(n - l + 1):

      j = i + l - 1

      m[i][j] = float('inf')

      for k in range(i, j):

        q = m[i][k] + m[k + 1][j] + p[i] * p[k + 1] * p[j + 1]

        if q < m[i][j]:

          m[i][j] = q


  return m[0][n-1]


d = [4, 10, 3, 12, 20, 7]

cost = mul(d)

print(f"minimum multiplication required is: {cost}")
```

Output:-



```
PS C:\Users\Admin\Desktop\Sem 5\AAD>  c:; cd 'c:\Users\Admin\Desktop\Sem 5\AAD'; & 'c:\Users\Admi
hon.exe' 'c:\Users\Admin\.vscode\extensions\ms-python.debugpy-2024.10.0-win32-x64\bundled\libs\de
 '--' 'C:\Users\Admin\Desktop\Sem 5\AAD\internal prac.py'
minimum multiplication required is: 1344
PS C:\Users\Admin\Desktop\Sem 5\AAD>
```