COMPETITIVE PROGRAMMING

Assignment-03

2303A51463

B-07

Assignment 4: Coin Change – Minimum Coins

Algorithm:

1.Input integer n (number of coin types).

2.Input array coins[ ] of size n.

3.Input integer A (amount).

4.Create an array dp[0…A].

5.Initialize dp[0]=0 and dp[1…A]=∞.

6.For each coin value c in coins:

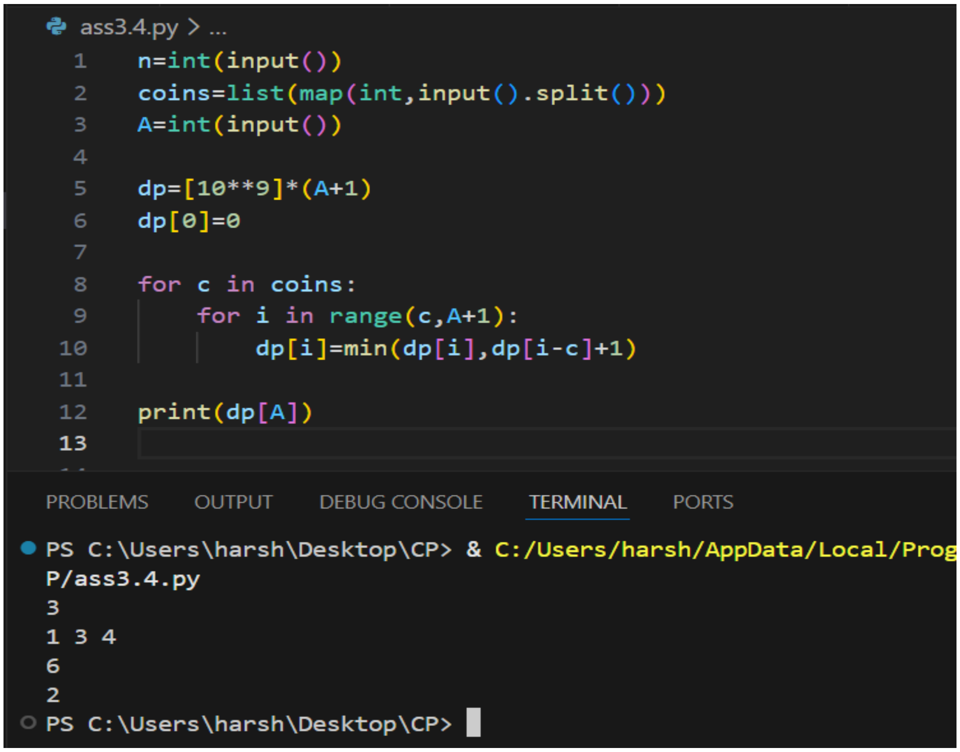
7.For each amount i from c to A:

8.Update dp[i]=min(dp[i],dp[i−c]+1).

9.After all iterations, dp[A] gives the minimum coins required.

10.Print dp[A].

Code and output:



```python
n=int(input())
coins=list(map(int,input().split()))
A=int(input())

dp=[10**9]*(A+1)
dp[0]=0

for c in coins:
    for i in range(c,A+1):
        dp[i]=min(dp[i],dp[i-c]+1)

print(dp[A])

```

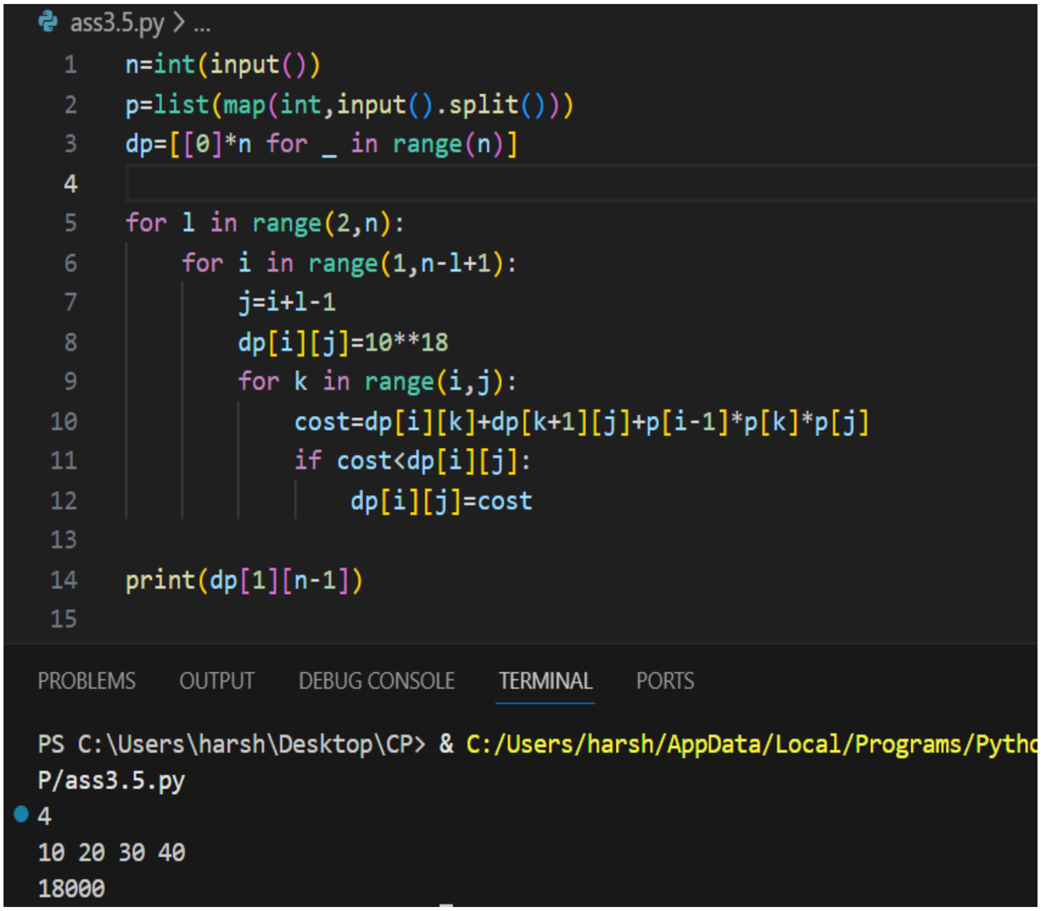
```
PS C:\Users\harsh\Desktop\CP> & C:/Users/harsh/AppData/Local/Prog
P/ass3.4.py
3
1 3 4
6
2
PS C:\Users\harsh\Desktop\CP> 
```

Assignment 5: Matrix Chain Multiplication

Algorithm:

1.Read n (number of dimensions).
2.Read array p[ ] of size n.
3.Create DP table dp[n][n].
4.Set dp[i][i] = 0.
5.For length from 2 to n-1:
6.For each i, find j = i + length - 1.
7.Try all split points k between i and j.
8.Compute cost = dp[i][k] + dp[k+1][j] + p[i-1] * p[k] * p[j].
9.Store minimum cost in dp[i][j].
10.Print dp[1][n-1].

Code and output:



```python
n=int(input())
p=list(map(int,input().split()))
dp=[[0]*n for _ in range(n)]

for l in range(2,n):
    for i in range(1,n-l+1):
        j=i+l-1
        dp[i][j]=10**18
        for k in range(i,j):
            cost=dp[i][k]+dp[k+1][j]+p[i-1]*p[k]*p[j]
            if cost<dp[i][j]:
                dp[i][j]=cost

print(dp[1][n-1])
```

```
PS C:\Users\harsh\Desktop\CP> & C:/Users/harsh/AppData/Local/Programs/Pytho
P/ass3.5.py
4
10 20 30 40
18000
```