

### 1.Dice throw problem:-

Code:-

```
import random
```

```
def dice_throw():
```

```
    return random.randint(1, 6)
```

```
# Test the function
```

```
result = dice_throw()
```

```
print("The dice rolled:", result)
```

Output:-

Programiz Python Online Compiler

Google Ads Get Rs.20,000 ad credit... Get Started

main.py Save Run Output Clear

```
1 import random
2
3 def dice_throw():
4     return random.randint(1, 6)
5
6 # Test the function
7 result = dice_throw()
8 print("The dice rolled:", result)
9
```

The dice rolled: 5

=== Code Execution Successful ===

## 2. Sum subset problem:-

Code:-

```
def sum_subset(arr, target):
```

```
    if target == 0:
```

```
        return True
```

```
    if not arr or target < 0:
```

```
        return False
```

```
    return sum_subset(arr[1:], target - arr[0]) or sum_subset(arr[1:], target)
```

### # Example Usage

```
arr = [3, 34, 4, 12, 5, 2]
```

```
target_sum = 9
```

```
print(sum_subset(arr, target_sum)) # Output: True
```

Output:-

The screenshot shows a web browser window with the URL `programiz.com/python-programming/online-com...`. The page features the Programiz logo and a banner for Google Workspace. Below the banner is a code editor with a file named `main.py`. The code in the editor is as follows:

```
1 def sum_subset(arr, target):
2     if target == 0:
3         return True
4     if not arr or target < 0:
5         return False
6     return sum_subset(arr[1:], target -
7                       arr[0]) or sum_subset(arr[1:],
8                       target)
9
10 # Example Usage
11 arr = [3, 34, 4, 12, 5, 2]
12 target_sum = 9
13 print(sum_subset(arr, target_sum)) #
    Output: True
```

The code is executed, and the output is displayed on the right side of the interface:

```
True
=== Code Execution Successful ===
```

### 3.Assembly line scheduling:-

Code:-

```

def assembly_line_scheduling(processing_time, switching_time, entry_time, exit_time):

    f1 = [0] * len(processing_time)
    f2 = [0] * len(processing_time)

    f1[0] = entry_time[0] + processing_time[0][0]
    f2[0] = entry_time[1] + processing_time[1][0]

    for i in range(1, len(processing_time)):

        f1[i] = min(f1[i - 1] + processing_time[0][i], f2[i - 1] + switching_time[1] +
processing_time[0][i])

        f2[i] = min(f2[i - 1] + processing_time[1][i], f1[i - 1] + switching_time[0] +
processing_time[1][i])

    f1[-1] += exit_time[0]
    f2[-1] += exit_time[1]

    return min(f1[-1], f2[-1])

# Example input
processing_time = [[7, 9, 3, 4, 8], [8, 5, 6, 4, 5]]
switching_time = [2, 3]
entry_time = [2, 3]
exit_time = [3, 2]

print(assembly_line_scheduling(processing_time, switching_time, entry_time, exit_time))

```

Output;-

```
main.py
3 f2 = [0] * len(processing_time)
4 f1[0] = entry_time[0] +
  processing_time[0][0]
5 f2[0] = entry_time[1] +
  processing_time[1][0]
6
7 for i in range(1, len(processing_time
  )):
8     f1[i] = min(f1[i - 1] +
  processing_time[0][i], f2[i -
  1] + switching_time[1] +
  processing_time[0][i])
9     f2[i] = min(f2[i - 1] +
  processing_time[1][i], f1[i -
  1] + switching_time[0] +
  processing_time[1][i])
10
11 f1[-1] += exit_time[0]
12 f2[-1] += exit_time[1]
13
14 return min(f1[-1], f2[-1])
15
16 # Example input
17 processing_time = [[7, 9, 3, 4, 8], [8, 5
  , 6, 4, 5]]
18 switching_time = [2, 3]
19 entry_time = [2, 3]
20 exit_time = [3, 2]
21
22 print(assembly_line_scheduling
  (processing_time, switching_time,
  entry_time, exit_time))
```

Output

18

=== Code Execution Successful ===

#### 4. Longest palindromic subsequence:-

Code:-

```
def longest_palindromic_subsequence(s):
```

```
    n = len(s)
```

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

```
    for i in range(n-1, -1, -1):
```

```
        dp[i][i] = 1
```

```
for j in range(i+1, n):  
    if s[i] == s[j]:  
        dp[i][j] = 2 + dp[i+1][j-1]  
    else:  
        dp[i][j] = max(dp[i+1][j], dp[i][j-1])  
  
return dp[0][n-1]
```

**# Example**

```
s = "character"
```

```
print(longest_palindromic_subsequence(s))
```

**Output:-**

FREE AI Code Generator: Generat... x

Online Python Compiler (Interpre... x


+

—

□

×














← → ↺ 🌐 programiz.com/python-programming/online-com... 🔍 ☆ 🌐 📁 | A ⋮



  
Python Online  
Compiler

Let clients pick a time 📅 that works for them  
with one-click appointment scheduling.

Google Workspace [Sign up](#)

P  
P

main.py   [Save](#) [Run](#)

Output [Clear](#)

```
1 def longest_palindromic_subsequence(s):
2     n = len(s)
3     dp = [[0] * n for _ in range(n)]
4
5     for i in range(n-1, -1, -1):
6         dp[i][i] = 1
7         for j in range(i+1, n):
8             if s[i] == s[j]:
9                 dp[i][j] = 2 + dp[i+1][j-1]
10            else:
11                dp[i][j] = max(dp[i+1][j],
12                               dp[i][j-1])
13
14     return dp[0][n-1]
15
16 # Example
17 s = "character"
18 print(longest_palindromic_subsequence(s))
```

5  
  
=== Code Execution Successful ===