



**Course Title:** Data Structure & Algorithm Lab II

**Course Code:** CSE 2218

**Trimester & Year:** Spring 2024

**Section:** A

**Credit Hours:** 1.0 (MdmH)

## ASSIGNMENT 02: Dynamic Programming

### Q1: Minimum Cost for Climbing Stairs

You are given an integer array cost where cost[i] is the cost of  $i^{\text{th}}$  step on a staircase. Once you pay the cost, you can either climb one or two steps.

You can either start from the step with index 0, or the step with index 1.

Return the minimum cost to reach the top of the floor.

#### Example 1:

Input: cost = [10,15,20]

Output: 15

Explanation: You will start at index 1.

- Pay 15 and climb two steps to reach the top.

The total cost is 15.

#### Example 2:

Input: cost = [1,100,1,1,1,100,1,1,100,1]

Output: 6

Explanation: You will start at index 0.

- Pay 1 and climb two steps to reach index 2.
- Pay 1 and climb two steps to reach index 4.
- Pay 1 and climb two steps to reach index 6.
- Pay 1 and climb one step to reach index 7.
- Pay 1 and climb two steps to reach index 9.
- Pay 1 and climb one step to reach the top.

The total cost is 6.



## Q2: Cut Into three Segments

You are given an integer 'N' denoting the length of the rod. You need to determine the maximum number of segments you can make of this rod provided that each segment should be of the length 'X', 'Y' or 'Z'.

### **Input Format:**

The first line of input contains an integer 'T' denoting the number of test cases.

The next 'T' lines represent the 'T' test cases.

Each test case contains 4 integers denoting 'N', 'X', 'Y', and 'Z', where 'N' is the length of the rod and 'X', 'Y', 'Z' are the segments into which a given rod can be cut into.

### **Output Format:**

For each test case, return the maximum number of cut segments from the given rod.

Print the output of each test case in a separate line.

### **Constraint:**

$1 \leq T \leq 50$

$1 \leq N \leq 10000$

$1 \leq X, Y, Z \leq N$

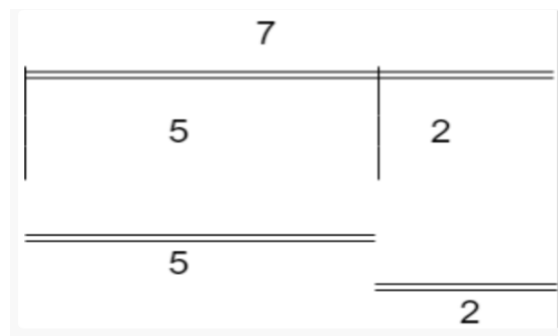
### **Sample Input 1:**

```
2
7 5 2 2
8 3 3 3
```

### **Sample Output 1:**

```
2
0
```

### **Explanation For Sample Input 1:**



In the first test case, cut it into 2 parts of 5 and 2.



**UNITED INTERNATIONAL UNIVERSITY**  
Department of Computer Science and Engineering (CSE)

In the second case, there is no way to cut into segments of 3 length only as the length of the rod is less than the given length.

**Sample Input 2:**

2

7 3 2 2

8 1 4 4

**Sample Output 2:**

3

8

**Explanation:**

In the first test case, cut it into 3 parts of 3, 2 and 2.

In the second case, cut it into 8 parts of length 1.



### Q3: Maximum Sum of Non-Adjacent Elements

You are given an array/list of 'N' integers. You are supposed to return the maximum sum of the subsequence with the constraint that no two elements are adjacent in the given array/list.

*Note: A subsequence of an array/list is obtained by deleting some number of elements (can be zero) from the array/list, leaving the remaining elements in their original order.*

*Input format:*

*The first line contains a single integer 'T' denoting the number of test cases.*

*The first line of each test case contains a single integer 'N' denoting the number of elements in the array.*

*The second line contains 'N' single space-separated integers denoting the elements of the array/list.*

*Output format:*

*For each test case, print a single integer that denotes the maximum sum of the non-adjacent elements.*

*Print the output of each test case in a separate line.*

*Constraints:*

$1 \leq T \leq 500$

$1 \leq N \leq 1000$

$0 \leq \text{ARR}[i] \leq 10^5$

Where 'ARR[i]' denotes the  $i^{\text{th}}$  element in the array/list

**Sample Input 1:**

```
2
3
1 2 4
4
2 1 4 9
```

**Sample Output 1:**

```
5
11
```

**Explanation to Sample Output 1:**

In test case 1, the sum of 'ARR[0]' & 'ARR[2]' is 5 which is greater than 'ARR[1]' which is 2 so the answer is 5.



**UNITED INTERNATIONAL UNIVERSITY**  
Department of Computer Science and Engineering (CSE)

In test case 2, the sum of 'ARR[0]' and 'ARR[2]' is 6, the sum of 'ARR[1]' and 'ARR[3]' is 10, and the sum of 'ARR[0]' and 'ARR[3]' is 11. So if we take the sum of 'ARR[0]' and 'ARR[3]', it will give the maximum sum of sequence in which no elements are adjacent in the given array/list.

**Sample Input 2:**

2

5

1 2 3 5 4

9

1 2 3 1 3 5 8 1 9

**Sample Output 2:**

8

24

**Explanation to Sample Output 2:**

In test case 1, out of all the possibilities, if we take the sum of 'ARR[0]', 'ARR[2]' and 'ARR[4]', i.e. 8, it will give the maximum sum of sequence in which no elements are adjacent in the given array/list.

In test case 2, out of all the possibilities, if we take the sum of 'ARR[0]', 'ARR[2]', 'ARR[4]', 'ARR[6]' and 'ARR[8]', i.e. 24 so, it will give the maximum sum of sequence in which no elements are adjacent in the given array/list.



#### Q4: TASFIA NEEDS TO BAKE A CAKE!!

Tasfia is very happy as tomorrow is her best friend's birthday. She wants to gift her a delicious cake. But Tasfia has spent most of her money in buying chocolate for herself and so she wants to get the best out of her remaining money. Now she has only "**M**" rupees remaining. She decides to make a delicious cake for her best friend with as much ingredients as possible.

So Tasfia went to the super-market and found out that "**I**" ingredients are sold in each bag for a price of "**P**" rupee. There were "**N**" types of bags in the super-market and she can buy any number of bags of each type. She has to buy at least "**K**" number of ingredients to make her cake delicious, otherwise she will not be able to make a delicious cake. Your task is to help Tasfia make the most delicious cake for her friend.

#### *Input Format:*

- The first line of input contains three space-separated integers **N**, **M** and **K** denoting the no types of bags, the amount of money Tasfia has and minimum number of ingredients Tasfia needs to buy to make a delicious cake respectively.
- **N** lines follow. The  $i^{\text{th}}$  line contains two space-separated integers  $I_i$  and  $P_i$  denoting the number of ingredients in the  $i^{\text{th}}$  bag and price of the  $i^{\text{th}}$  bag. ( $1 \leq i \leq N$ )

#### *Output Format:*

For each test case print "**YES**" without quotes along with the maximum number of ingredients Tasfia can buy if she can make a delicious cake else print "**NO**" without quotes.



**UNITED INTERNATIONAL UNIVERSITY**  
Department of Computer Science and Engineering (CSE)

Sample Input	Sample Output	Explanation
3 6 1 6 4 5 3 1 1	Yes 7	Tasfia can buy 7 ingredients by taking the 1 <sup>st</sup> and 3 <sup>rd</sup> type of bag.
3 10 12 8 10 7 11 11 12	No	There is no possible way that Tasfia can make a delicious cake with her money.