

Ex6.3

<https://www.codingame.com/ide/puzzle/roller-coaster>

```
import sys
import math

l, c, n = [int(i) for i in input().split()]
#print (l, c, n)
groupSize=[]
for i in range(n):
    pi = int(input())
    groupSize.append(pi)
numOfPassForRide=[]
lastGroupIndex=[]
for i in range(n):
    numOfPassForRide.append(0)
    lastGroupIndex.append(0)
for start in range(n):
    for k in range(n):
        index=(start+k)%n
        if numOfPassForRide[start]+groupSize[index]<=l:
            numOfPassForRide[start]=numOfPassForRide[start]+groupSize[index]
            lastGroupIndex[start]=index
        else:
            break;

sum=0
startIndex=0
for i in range(c):
    sum+=numOfPassForRide[startIndex]
    startIndex=(lastGroupIndex[startIndex]+1)%n
print(sum)
```

Roller Coaster

The Goal

You have recently been assigned to a new amusement park's center of analysis and supervision. Your mission is to estimate each day what the earnings will be for each ride that day. You start by looking at the roller coaster.

Rules

You notice that people like the roller coaster so much that as soon as they have finished a ride, they cannot help but go back for another one.

- People queue up in front of the attraction
- They can either be alone or in a group. When groups are in the queue, they necessarily want to ride together, without being separated.
- People never overtake each other in the queue.
- When there isn't enough space in the attraction for the next group in the queue, the ride starts (so it is not always full).
- As soon as the ride is finished, the groups that come out, go back into the queue in the same order.

The attraction contains a limited number l of places.
The attraction can only function c number of times per day.
The queue contains a number n of groups.
Each group contains a number p_i of people.
Each person spends 1 dirham per ride.

Python 3

```
5 l, c, n = [int(i) for i in input().split()]
6 #print (l, c, n)
7 groupSize=[]
8 for i in range(n):
9     pi = int(input())
10    groupSize.append(pi)
11 numOfPassForRide=[]
12 lastGroupIndex=[]
13 for i in range(n):
14     numOfPassForRide.append(0)
15     lastGroupIndex.append(0)
16 for start in range(n):
17     for k in range(n):
18         index=(start+k)%n
19         if numOfPassForRide[start]+groupSize[index]<=l:
20             numOfPassForRide[start]=numOfPassForRide[start]+groupSize[index]
21             lastGroupIndex[start]=index
22         else:
23             break;
24
25 sum=0
26 startIndex=0
27 for i in range(c):
28     sum+=numOfPassForRide[startIndex]
29     startIndex=(lastGroupIndex[startIndex]+1)%n
30 print(sum)
```

Test cases

01 Simple case

Actions

PLAY ALL TESTCASES

Ex6.3

<https://www.codingame.com/ide/puzzle/roller-coaster>

MY REPORT

HISTORY

SCORE

100%

SHARE

DETAILS →

WHAT YOU LEARNED

The solution to this puzzle lies in the following concepts. If you consider that you've acquired the skills listed below, tick the corresponding boxes (and they'll appear on your learning profile). If not, brush up on your knowledge and try again!

☐ Dynamic programming ☐ Simulation

LEARN FROM THE BEST CODERS

Browse the best solutions for this puzzle →

Python 3

```
1 # sum of money
27 for i in range(c):
28     sum+=numberOfPassForRide[startIndex]
29     startIndex=(lastGroupIndex[startIndex]+1)%n
30     print(sum)
31
32 # start index = 0
33 # group list = []
34 # for i in range(n):
35 #     if startIndex not in group list:
36 #         group list.append(startIndex)
37 #         startIndex=(lastGroupIndex[startIndex]+1)%n
38 #     else:
39 #         break
40 # total money = 0
41 # number of groups = len(group list)
42 # for i in range(number of groups):
43 #     times a group = c // number of groups
44 #     if times a group <= 0 and (number of groups > 1):
45 #         times a group = 1
46 #     print(i, times a group, file=sys.stdout, flush=True)
47 #     if times a group > 0:
48 #         total money += times a group * numberOfPassForRide[group list[i]]
49
50 # print(total money)
51
```

Test cases

03 The same groups go on the ride several times d... ▶ PLAY TESTCASE

Actions