Zonal Informatics Olympiad, 2027

Instructions to candidates

1. The question paper carries 80 marks, broken up into four problems of 20 marks each. Each problem has three Test Cases. If you solve all three Test Cases correctly, you get 20 marks for that problem. Otherwise, you get 5 marks for each Test Case that you solve correctly.
2. All the $4 \times 3 = 12$ Test Cases appear as separate Questions in the right panel ("Question Palette").
   The first three Questions correspond to the three Test Cases of Problem 1, the next three correspond to the three Test Cases of Problem 2 and so on.
   A question icon turning green in the Question Palette, does not mean that it is correct. It just denotes that you have attempted it. All the questions will be evaluated later.
3. Attempt all questions. There are no optional questions.
4. There are no negative marks.
5. All expected answers are integers. Type in only the integer. So, if your answer is "162", enter only "162". Not "0162", or "162.0", etc.
6. Remember to save each answer. Only your final saved answers will be considered.
7. Near the top-right corner, you should be able to see a calculator icon. Clicking it pops up a calculator which you may use.

1 Problem 1

There are N people waiting on the ground floor of a building; the ground floor is floor 0. Person i wants to go to floor F_i, where $F_i \geq 1$. The building has one lift, which can carry at most C people at a time.

On each trip, the lift takes some of the waiting people, goes up and drops each of them off at their floor, and then comes back down to the ground floor. The lift takes 1 second to move up or down by one floor. Stopping and people getting in or out take no time.

Your task is to find the minimum total time needed to deliver everyone to their floors, with the lift back on the ground floor at the end.

For example, for the input $N = 3$, $C = 2$, $F = [2, 3, 5]$ (that is, $F_1 = 2$, $F_2 = 3$, and $F_3 = 5$), one way to deliver everyone is:

- **Trip 1:** persons 2 and 3 (going to floors 3 and 5) travel together. The lift goes up to floor 3 and drops off person 2, goes on up to floor 5 and drops off person 3, and then comes back down to the ground floor. This takes 5 seconds up and 5 seconds down, so 10 seconds.
- **Trip 2:** person 1 (going to floor 2) travels alone. The lift goes up to floor 2 and comes back down, which takes $2 + 2 = 4$ seconds.

This takes a total of $10 + 4 = 14$ seconds.

Other ways are slower: for instance, if the people going to floors 2 and 5 travel together (10 seconds) and the person going to floor 3 travels alone (6 seconds), the total is 16 seconds. No way is faster than 14 seconds, so your answer should be 14.

(a) $N = 4$, $C = 2$, $F = [1, 3, 4, 6]$

(b) $N = 9$, $C = 3$, $F = [1, 2, 4, 4, 5, 7, 8, 9, 12]$

(c) $N = 18$, $C = 4$, $F = [2, 3, 4, 5, 6, 8, 9, 11, 13, 14, 16, 17, 19, 21, 22, 25, 27, 30]$

2 Problem 2

There are N lamps in a row in a corridor, numbered 1 to N. Each lamp is either on or off. Next to lamp i there are two switches:

- a **small switch**, which toggles lamp i only;
- a **big switch**, which toggles every lamp from i to the end of the corridor, i.e. lamps $i, i+1, \ldots, N$.

Toggling a lamp turns it off if it was on, and on if it was off. Your task is to find the minimum number of switch presses needed to turn every lamp off.

The input is a string S of length N, where the i-th character of S is `1` if lamp i is on and `0` if lamp i is off.

For example, for the input $N = 5$, $S =$ `11011` (lamps 1, 2, 4, and 5 are on, and lamp 3 is off), one way to turn every lamp off is:

- **First Press:** the big switch at lamp 1. This toggles lamps 1 to 5, so `11011` becomes `00100`.
- **Second Press:** the small switch at lamp 3. This toggles only lamp 3, so `00100` becomes `00000`.

This takes 2 presses.

You can check that it is impossible with only 1 press. Thus, your answer should be 2.

(a) $N = 6$, $S =$ `111001`

(b) $N = 10$, $S =$ `1011010111`

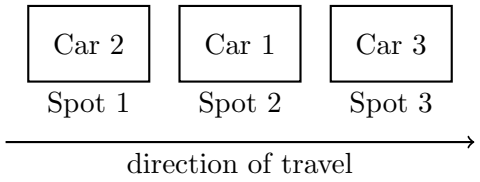
(c) $N = 24$, $S =$ `011010101100011010110110`

3 Problem 3

A one-way street has N parking spots, numbered 1 to N in the direction of travel. Cars $1, 2, \ldots, N$ arrive one after another, in this order. Car i has a favourite spot f_i, where $1 \leq f_i \leq N$. It drives to spot f_i and parks there if the spot is free; otherwise, it keeps driving and parks in the first free spot after f_i. Cars never turn back. If there is no free spot at or after spot f_i, then the car just leaves the street.

One evening, all N cars managed to park, and you noted which car ended up in each spot: spot j holds car A_j. So A is a permutation of $1, 2, \ldots, N$. Your task is to find the number of lists of favourite spots $(f_1, f_2, \ldots, f_N)$ that produce exactly this final arrangement.

For example, for the input $N = 3$, $A = [2, 1, 3]$, the final arrangement is:



Exactly three lists of favourite spots produce this arrangement:

- $f = [2, 1, 1]$: car 1 finds spot 2 free and parks there. Car 2 finds spot 1 free and parks there. Car 3 finds spot 1 taken, drives past spots 1 and 2 (both taken), and parks in spot 3.
- $f = [2, 1, 2]$: cars 1 and 2 park as above. Car 3 finds spot 2 taken, drives on, and parks in spot 3.
- $f = [2, 1, 3]$: cars 1 and 2 park as above. Car 3 finds spot 3 free and parks there.

In each case, spot 1 holds car 2, spot 2 holds car 1, and spot 3 holds car 3, which is exactly $A = [2, 1, 3]$.

On the other hand, $f = [2, 2, 1]$ does not work: car 1 parks in spot 2, car 2 finds spot 2 taken and parks in spot 3, and car 3 parks in spot 1. This gives the arrangement $[3, 1, 2]$ instead.

You can check that no other list produces $A = [2, 1, 3]$. Thus, your answer should be 3.

(a) $N = 4$, $A = [3, 1, 2, 4]$

(b) $N = 8$, $A = [1, 2, 3, 4, 5, 6, 7, 8]$

(c) $N = 15$, $A = [4, 1, 2, 7, 3, 15, 5, 9, 6, 8, 11, 10, 13, 12, 14]$

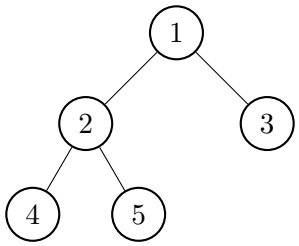
4 Problem 4

A museum has N rooms, numbered 1 to N, and $N-1$ doors. Each door connects two rooms, and by walking through these doors, it is possible to get from any room to any other room. The input consists of the integer N, followed by an array P of length N such that $P[1] = -1$ and $1 \leq P[i] < i$ for all $2 \leq i \leq N$. For each i such that $2 \leq i \leq N$, there is a door between rooms $P[i]$ and i.

A guide starts in room 1 and walks from room to room through the doors until every room has been visited at least once. The tour may end in any room. A tour is written as the sequence of rooms the guide is in, so a room may appear more than once, and each step from one room to the next is called a move.

Among all tours that visit every room, consider the ones with the fewest moves. Your task is to find the number of such shortest tours.

For example, for the input $N = 5$, $P = [-1, 1, 1, 2, 2]$, the doors are 1–2, 1–3, 2–4, and 2–5, as shown below.



Every tour that visits all rooms needs at least 6 moves, and exactly two tours have 6 moves:

- $1, 3, 1, 2, 4, 2, 5$: the guide goes to room 3 and back to room 1, then goes to room 2, goes to room 4 and back to room 2, and ends in room 5.
- $1, 3, 1, 2, 5, 2, 4$: the same, but visiting room 5 before room 4, and ending in room 4.

You can check that both of these tours have 6 moves.

Other tours that visit every room take more moves. For instance, the tour $1, 2, 4, 2, 5, 2, 1, 3$ takes 7 moves, so it is not counted. Thus, your answer should be 2.

(a) $N = 7$, $P = [-1, 1, 1, 1, 2, 2, 4]$

(b) $N = 11$, $P = [-1, 1, 2, 2, 3, 2, 2, 7, 6, 6, 1]$

(c) $N = 21$, $P = [-1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 4, 5, 5, 5, 9, 9, 9, 12, 15, 11, 20]$