Task 2. Ones

There is a hidden array of N bits – $a_0, \ldots, a_{N-1}$. In one query, you can choose a subset of positions in the sequence and flip the bits at those positions. Flipping a 0 bit changes it to a 1 and flipping a 1 bit turns it into a 0. After each query, you are given the length of the longest consecutive subarray of 1s in the new array. **Such queries persist, i.e. a flipped bit from a previous query will stay flipped until flipped back.**

You want to find where the longest subarray of ones in the array (or any one of them if multiple ones exist) is located **after all queries.** Write a program to do this in as few queries as possible.

Implementation details

There are T subtests per test.

Your program needs to implement a function with the following prototype:

```
std::pair<int,int> find_longest_subarray_of_ones(int n);
```

It will be called T times per test and receives the integer N as a parameter – the length of the hidden array. The function should return a `pair{l,r}`, where `l` is the index of the leftmost part of the subarray and `r` is the index of the rightmost part of the subarray.

You should use the function `flip_bits` to make queries:

```
int flip_bits(const std::vector<bool> &flips);
```

It receives a vector of N bits $flips_0, \ldots, flips_{n-1}$, where $flips_i = 1$ would signify that a_i should be flipped. After flipping the required bits, the function returns the length of the longest subarray of ones in the hidden sequence.

You must submit the file `ones.cpp` to the system which contains the `find_longest_subarray_of_ones` function. It may contain other code and functions necessary for your work, but it must not contain the `main` function. Also, you must not read from the standard input or print to the standard output. Your program must also include the `ones.h` header file by instruction to the preprocessor:

```
#include "ones.h"
```

Local testing

You are given the file Lgrader.cpp, which can be compiled with your program to test it. It will read the number of subtests T for the test, followed by a description of each of them. For each subtest, first the integer N will be given, followed by $a_0 \ \ldots a_{N-1}$ on the next line. If it runs correctly, the program will output 1 and the maximum number of queries you have requested. Otherwise, it will print 0.

Constraints

$T = 5$

$1 \leq N \leq 10^4$

$0 \leq a_i \leq 1$

Scoring

If your program is wrong for any subtest, your score will be 0. Otherwise, the score you receive for the problem depends on the maximum number of queries Q your program has used to solve a single subtest. The score that you receive for the problem is:

If $Q >= 60$:

$$\left(\frac{60}{Q}\right)^{0.215} \times 70$$

Else, Q < 60 and the score is:

$$-\frac{3}{2}max(40, Q) + 160$$

Sample Communication

Let the starting array be 1, 0, 1. Then one way for the communication to go is:

Contestant's function	Jury's program
	Calls **find_longest_subarray_of_ones(3)**
Calls **flip_bits({0, 0, 0})**	Returns the value **1**
Calls **flip_bits({0, 1, 0})**	Returns the value **3**
Returns the value **{0, 2}**	