NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER ENGINEERING
DEPARTMENT OF INFORMATION TECHNOLOGY AND COMPUTERS

Spring 2021

PROGRAMMING LANGUAGES I.

Exercise 1

Closing date and time of electronic submission: 23/4/2021, 23:59:59

ICU under constant pressure (0.25 + 0.25 = 0.5 points)

Due to the pandemic, health systems around the world are under pressure. Every day some Our fellow human beings are admitted to the ICU (and occupy a bed) and some others either recover or, unfortunately, end up (and free a bed). We study an interval M consecutive days (where $1 \le M \le 500,000$), in each of which we correspond to one integer: the difference of the occupied ICU beds from the previous day.

Suppose that in the country we are interested in there are N hospitals (where $1 \le N \le 1,000$). We will say that a period of K consecutive days is " **good** " for the country, if in this A total of at least one bed per hospital is released over a period of time per day. For example, if the country's hospitals are N = 3 and in a period K = 4 days the sum of the daily differences of occupied beds is −14, then this period is good for the country (because the average number of beds vacated per hospital and per day is $14/12 \ge 1$). Conversely, if in a period K = 6 days the corresponding sum is −17, then this period is not considered good (because the average number of beds are released per hospital and per day is 17/18 <1).

Help the (lifelong and perhaps centuries-old, for such M values we are discussing) its president country to calculate the number of days of the longest "good" period.

The exercise asks you to write two programs (one in C / C ++ and one in ML) which read the input as shown below and print the duration K in its days longer during the "good" period of the country.

The input data will be read from a file as shown in the examples below follow. The first line of the file contains two natural integers: M (days) and N (hospitals). The second line of the file contains M integers (the difference of the entries minus the exports from the ICUs on the respective day), separated by a gap space. You can assume that both input numbers and algebraic sums for each possible period will not exceed 1,000,000,000 in absolute value.

Below is an example in C / C ++ and ML. Based on the data in the file entry, the longest "good" period in the country (`-6 -12 16 -15 -11`) lasted 5 days.

In C / C ++, MLton, or OCaml **In SML / NJ**

```
$ ./longest f.txt
```

```
- longest "f.txt";
val it = (): unit
```

where the input data file is as follows (the **cat** command is a Unix command):

```
$ cat f.txt
11 3
42 -10 8 1 11 -6 -12 16 -15 -11 13
```

Page 2

Loops in video games (0.25 + 0.25 = 0.5 points)

One of the few things that can happen in a pandemic is all kinds of things stupid video games. In one of them, players are placed in a maze from where must come out. The labyrinth consists of N × M rooms in a rectangular layout. In every room there is a puzzle and, if the player solves it, then a door opens leading to a specific adjoining room (up, down, left or right). If a room is located in the contour of the maze and open a door that leads out, then the player comes out and wins. We are talking about overproduction!

The dimensions of the maze and the location of the exit door for each room are known from principle. In each room, the door is marked with one of the letters "U" (up), "D" (down), “L” (left) or “R” (right). So if a player starts from a starting room and solves against order all the puzzles in the rooms he visits, the path he will follow is specific. Unfortunately for the players, it is possible that the exit doors are like that placed so that from some of the original rooms it is not possible to leave the labyrinth! See the example below. The arrows on the right of the figure show course of players from each room .

ULD

LUD

LRL

If a player starts from a starting room marked in yellow, then it will get stuck in a circular path and will never be able to get out of the maze, whatever puzzles he solves. On the contrary, he will manage to get out of the other rooms, if of course solve the puzzles required.

The producers of the game want their video game to be fair, that is, the initials rooms from which players can not win as many puzzles as they solve they are not much. Help them count them!

The input data is read from a file as shown in the following example. The first line of the file has two integers N and M (1 ≤ N, M ≤ 1,000), its dimensions maze (rows and columns, respectively). Each of the following N lines represents one line of the maze and contains exactly M characters, each of which is one of the letters "U", "D", "L" or "R". The letter that corresponds to each room symbolizes the position of the exit door of the room.

The exercise asks you to write two programs (one in C / C ++ and one in ML). Each Your program should read the maze, make the necessary calculations

and print an integer: the number of rooms from which players do not
they can win, whatever puzzles they solve.

Below are two examples in C / C ++ and ML.

In C / C ++, MLton, or OCaml

```
$ ./loop_rooms maze1.txt
4

$ ./loop_rooms maze2.txt
2
```

In SML / NJ

```
- loop_rooms "maze1.txt";
4
val it = (): unit
- loop_rooms "maze2.txt";
2
val it = (): unit
```

Page 3

where the two files with the input data are as follows:

```
$ cat maze1.txt
3 3
ULD
LUD
LRL
```

```
$ cat maze2.txt
2 2
UL
RL
```

The **`maze1.txt`** file **`maze`** corresponds to the above figure.

Further instructions for the exercises

- You can work in groups of up to two people, both in this and the next
  series of exercises. But keep in mind that if you do not pass the course this year, the grades
  of programming exercises are reserved only for those who did not do them in a group but
  they made them themselves.

- You are not allowed to share your programs with fellow students outside your group or
  put them in a place where others can find them (eg on a web page,
  on talk sites,…). In case "strange" similarities are observed
  in programs, the grade of the students involved in *all the series of exercises* is done
  automatically zero regardless of which team ... was "inspired" by the other.

- You can use "helper" code (eg sort code, some code
  managed by a data structure) found on the internet in your programs,
  provided that your program contains in the comments the admission to
  source of this code and a link to it.

- Programs in C / C ++ must be in a file and can be compiled
  without warnings with gcc / g ++ (version ≥ 8.3.0) with commands of the format, e.g.

```
gcc –std = c99 -Wall –Werror -O3 –o longest longest.c
g ++ –std = c ++ 11 -Wall –Werror -O3 –o longest longest.cpp
```

- Programs in ML must also be in a file and work in SML / NJ

≥ v110.79 or in MLton ≥ 20130715 or in Objective Caml version ≥ 4.05.0. The system Online submission lets you choose between these ML dialects.

- The programs will be sent electronically through moodle and so that you can to submit them, your team members (both) should already have an account in moodle. There will be an announcement about the exact submission process when it opens the system. Your programs should read the input as stated and do not must have some other kind of output because they will not be accepted by the system in which will be submitted.