Lëtzebuerger Informatiksolympiad 2025

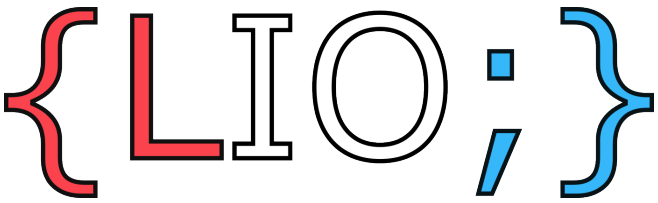
Semi-Finals

Task descriptions

Instructions

- The allowed programming languages are Python 3, Java and C/C++.
- All the programs must be realized in the form of a console application. For instructions how to realize a console application in the allowed programming languages, please refer to the remarks on the site `www.infosolympiad.lu` under the heading *The tasks*.
- Under the input of the program is meant either the direct entry of data from the keyboard or the redirection from a text file in console mode. Under output of the program is meant either the direct display of data to the screen or the redirection to a text file in console mode.
- The formats of the input and output data shown in the execution examples must absolutely be respected.
- For testing, submitting and evaluating a program, the source file with a file extension `py`, `java` or `c/cpp` must be uploaded to the automated online judge CMS (Contest Management System), accessible via the homepage `www.infosolympiad.lu` or directly via the URL `http://158.64.50.79/cws/`. Please use your personal login (username and password) to access your account on the CMS. The filename of the single source file should be the same than the task name. Please refer to the CMS for technical details on how to test and submit a program.
- Please refer to the CMS for technical details like time limits and memory limits as well as compilation commands.
- You have the right to ask questions via the CMS, but the answers will not teach you how to use a programming language nor tell you how to solve the tasks by using a specific algorithm. The questions should be in relation with the CMS or should treat clarification issues concerning the task descriptions.

Note: Solving Subtask 3 of `chickencompat` may require a language other than Python, due to its speed.



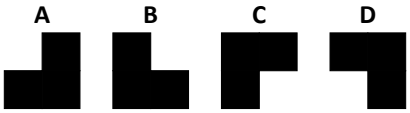
Tilly's Tetris Tower

Description

Tilly really likes Tetris, so she came up with her own version of the game. In her game there are only 2×2 L-shaped *tetrominoes* (see picture below). These tetrominoes fall from the sky in some specific order and can **not** be rotated.

The block types are defined as follows:

- **A**: The top-left cell is missing.
- **B**: The top-right cell is missing.
- **C**: The bottom-right cell is missing.
- **D**: The bottom-left cell is missing.



Tilly knows in advance the order in which the N tetrominoes will fall. She wants your help to calculate the resulting height of the tower, after all N tetrominoes haven dropped (in that order).

Task

Your task is to calculate the final height of the tower after all N blocks have fallen, given the order of the falling tetrominoes.

Constraints

- $1 \leq N \leq 10^6$
- The input contains only the capitalised characters: 'A', 'B', 'C', and 'D'.

Input and output of program

Input data

The first line contains N - the number of tetrominoes.
A single string A (of length N), describing the order of tetrominoes falling: the i-th character of this string describes the type of the i-th tetromino ('A', 'B', 'C', or 'D').

Output data

The answer

Execution example

Input

```
6
AACBDD
```

Output

```
10
```

Input

```
5
AACCA
```

Output

```
9
```

The following illustrations show the resulting towers for both examples.

Figure 1: Final tower for input `6 AACBDD`

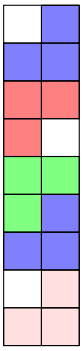


Figure 2: Final tower for input `5 AACCA`

Distribution of points

Subtask	Points	Constraints/Description
1	4	$N = 2$
2	9	The input only contains 'A' and 'C' tetrominoes.
4	17	No additional constraints

Technical constraints

Task name	tetris
Input file	standard input
Output file	standard input
Time limit	1 second
Memory limit	256 megabytes

Chicken Compatibility

Description

Lia's chickens are at it again, squabbling amongst themselves. The problem stems from their diverse clucking styles - because their calls are so different, the chickens do not seem to get along.
To reduce infighting in the barn, Lia has devised a method to determine the compatibility between two chickens based on their clucks. She writes down the clucks she hears as a string of a certain length N, and then, for two chickens, she tries to find the longest sequence of letters that appears in both cluck strings, in the same order. The length of that longest sequence is the compatibility score.
As the intonation of the chickens is irrelevant, the strings contain only lowercase letters.

Task

Given the clucks of two chickens (and the length), give their compatibility score.

Example

To test her method, Lia writes down very short clucks of size $N = 5$ and ends up with `bawkk` and `bokok`. The longest common part these two clucks have is `bkk`, therefore the compatibility between the two chickens is 3.

Constraints

- $0 \leq N \leq 2 * 10^4$
- the strings contain only lowercase letters.

Input and output of program

Input data

The first line contains N the length of the clucks. The two following lines each contain a string of length N that represents a cluck. The string only contains lowercase letters.

Output data

The compatibility between the two chickens.

Execution example

This is the input-output for the example given above.

Input

```
5
bawkk
bokok
```

Output

```
3
```

Distribution of points

Subtask	Points	Constraints/Description
1	10	$N \leq 10$
2	15	$N \leq 100$
3	10	No additional constraints

Technical constraints

Task name	chickencompat
Input file	standard input
Output file	standard output
Time limit	1 second
Memory limit	512 megabytes

Failgrade

Description

You are teacher and you have a class of S students and grade your papers from 0 to G_{max} (only integer grades). You have just finished grading all exams and now you have a list of N pairs (g, n), where g is a grade and n is the number of students who received that grade. You want to find the **highest** threshold T, $0 \leq T \leq G_{\text{max}}$, such that at most 50% of the students have grades strictly less than T (i.e., failed the exam).

Task

Given the grade-count pairs, find the threshold T.

Example

For example, suppose $N = 3$, $G_{\text{max}} = 10$ and the grade-count pairs are:

- Grade 3: 4 students,
- Grade 5: 5 students,
- Grade 8: 11 students.

Then the total number of students is $S = 4 + 5 + 11 = 20$, so 50% corresponds to 10 students.

- If $T = 5$, then only the 4 students with grade 3 have a grade strictly less than T, which is acceptable.
- If $T = 8$, then the students with grades 3 and 5 (totaling $4 + 5 = 9$ students) have grades strictly less than T, which is exactly 45%.
- If T were set higher than 8, more than 50% of the students will fall below the threshold.

Thus, the highest valid threshold in this case is 8.

Constraints

- $1 \leq N \leq 10^5$
- $1 \leq G_{\text{max}} \leq 10^9$

Input and output of program

Input data

The first line contains N and G_{max}, separated by a space.
Each of the next N lines contains two integers, g and n, representing a grade and the number of students who received that grade.

Output data

Output a single integer: the threshold T.

Execution example

Input

```
3 10
3 4
5 5
8 11
```

Output

```
8
```

Distribution of points

Subtask	Points	Constraints/Description
1	10	$N \leq 1000$ and $G_{\max} \leq 10^4$
2	15	$N \leq 10^4$
3	10	No additional constraints

Technical constraints

Task name	failgrade
Input file	standard input
Output file	standard input
Time limit	1 second
Memory limit	256 megabytes