



The CENTRE for EDUCATION
in MATHEMATICS and COMPUTING

*2011
Canadian
Computing
Competition:
Senior
Division*

Sponsor:



Canadian Computing Competition

Student Instructions for the Senior Problems

1. You may only compete in one competition. If you wish to write the Junior paper, see the other problem set.
2. Be sure to indicate on your **Student Information Form** that you are competing in the **Senior** competition.
3. You have three (3) hours to complete this competition.
4. You should assume that
 - all input is from a file named `sX.in`, where X is the problem number ($1 \leq X \leq 5$).
 - all output is to the screen

Since your program will read from a file, there is no need for prompting. Be sure your output matches the output in terms of order, spacing, etc. **IT MUST MATCH EXACTLY!**
5. Do your own work. Cheating will be dealt with harshly.
6. Do not use any features that the judge (your teacher) will not be able to use while evaluating your programs.
7. Books and written materials are allowed. Any machine-readable materials (like other programs which you have written) are *not* allowed. However, you are allowed to use “standard” libraries for your programming languages; for example, the STL for C++, `java.util.*`, `java.io.*`, etc. for Java, and so on.
8. Applications other than editors, compilers, debuggers or other standard programming tools are **not** allowed. Any use of other applications will lead to disqualification.
9. Please use file names that are unique to each problem: for example, please use `s1.pas` or `s1.c` or `s1.java` (or some other appropriate extension) for Problem S1. This will make the evaluator’s task a little easier.
10. Your program will be run against test cases other than the sample ones. Be sure you test your program on other test cases. Inefficient solutions may lose marks for some problems, especially Problems 4 and 5. Be sure your code is as efficient (in terms of time) as possible.
11. Note that the top 2 Senior competitors in each region of the country will get a plaque and \$100, and the schools of these competitors will also get a plaque. The regions are:
 - West (BC to Manitoba)
 - Ontario North and East
 - Metro Toronto area

- Ontario Central and West
 - Quebec and Atlantic
12. If you finish in the top 20 competitors on this competition, you will be invited to participate in CCC Stage 2, held at the University of Waterloo in May 2011. We will select the Canadian IOI team from among the top contestants at Stage 2. You should note that IOI 2011 will be held in Thailand. Note that you will need to know C, C++ or Pascal if you are invited to Stage 2. But first, do well on this contest!
 13. Check the CCC website at the end of March to see how you did on this contest, and to see who the prize winners are. The CCC website is:

`www.cemc.uwaterloo.ca/ccc`

Problem S1: English or French?

Problem Description

You would like to do some experiments in *natural language processing*. Natural language processing (NLP) involves using machines to recognize human languages.

Your first idea is to write a program that can distinguish English text from French text.

After some analysis, you have concluded that a very reasonable way of distinguishing these two languages is to compare the occurrences of the letters “t” and “T” to the occurrences of the letters “s” and “S”. Specifically:

- if the given text has more “t” and “T” characters than “s” and “S” characters, we will say that it is (probably) English text;
- if the given text has more “s” and “S” characters than “t” and “T” characters, we will say that it is (probably) French text;
- if the number of “t” and “T” characters is the same as the number of “s” and “S” characters, we will say that it is (probably) French text.

Input Specification

The input will contain the number N ($0 < N < 10000$) followed by N lines of text, where each line has at least one character and no more than 100 characters.

Output Specification

Your output will be one line. This line will either consist of the word `English` (indicating the text is probably English) or `French` (indicating the text is probably French).

Sample Input 1

```
3
The red cat sat on the mat.
Why are you so sad cat?
Don't ask that.
```

Output for Sample Input 1

```
English
```

Sample Input 2

```
3
Lorsque j'avais six ans j'ai vu, une fois,
une magnifique image,
dans un livre
```

Output for Sample Input 2

French

(Note: Sample Input 2 is the first sentence of “Le Petit Prince” by Antoine de Saint-Exupéry.)

Problem S2: Multiple Choice

Problem Description

Your teacher likes to give multiple choice tests. One benefit of giving these tests is that they are easy to mark, given an answer key. The other benefit is that students believe they have a one-in-five chance of getting the correct answer, assuming the multiple choice possibilities are A,B,C,D or E.

Write a program that your teacher can use to grade one multiple choice test.

Input Specification

The input will contain the number N ($0 < N < 10000$) followed by $2N$ lines. The $2N$ lines are composed of N lines of student responses (with one of A,B,C,D or E on each line), followed by N lines of correct answers (with one of A,B,C,D or E on each line), in the same order as the student answered the questions (that is, if line i is the student response, then line $N + i$ contains the correct answer to that question).

Output Specification

Output the integer C ($0 \leq C \leq N$) which corresponds to the number of questions the student answered correctly.

Sample Input 1

```
3
A
B
C
A
C
B
```

Output for Sample Input 1

```
1
```

Sample Input 2

```
3
A
A
A
A
B
A
```

Output for Sample Input 2

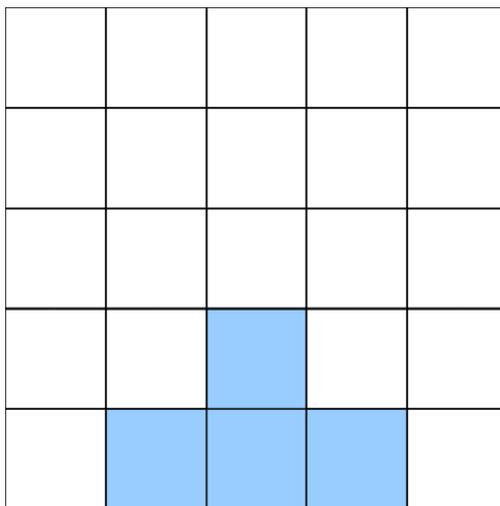
```
2
```

Problem S3: Alice Through the Looking Glass

Problem Description

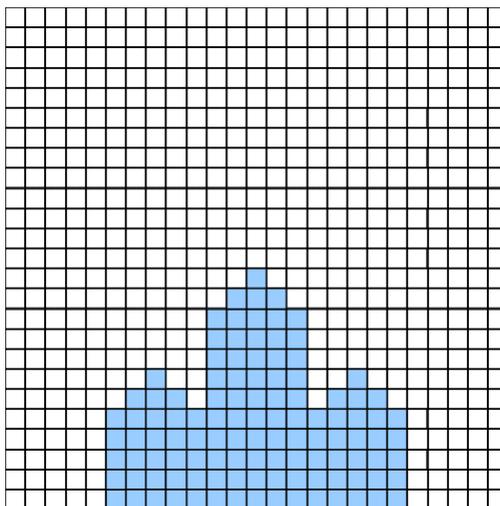
Alice is looking at a crystal through a microscope. Alice's microscope has the interesting feature that it can superimpose grid lines over the image that she is looking at.

At level 1 of magnification, Alice sees the image as follows:



Notice that at level 1, there is a 5×5 grid superimposed over the image.

However, as Alice increases the magnification, the leaf pattern becomes more intricate.



At level 2 of the magnification, Alice sees the image with a 25×25 grid, and notices that three of the four larger squares in the original image have the small four square pattern on top. In fact, for this particular crystal, this self-similarity repeats for each magnification level.

Given that Alice's microscope has up to 13 levels of magnification, she would like to try to quantify the detail of each grid cell at every one of these magnification levels.

Specifically, since there is a $5^m \times 5^m$ grid at magnification level m , Alice will call the bottom-left corner grid cell $(0, 0)$, the bottom-right grid cell $(5^m - 1, 0)$, the top-left grid cell $(0, 5^m - 1)$, and the top-right grid cell $(5^m - 1, 5^m - 1)$.

Given an integer magnification level m ($1 \leq m \leq 13$) and a grid position (x, y) (where $0 \leq x < 5^m$ and $0 \leq y < 5^m$), Alice would like to know if her crystal will fill that grid cell, or if that grid cell will be empty space.

Input Specification

The first line of input will be T ($0 < T \leq 10$) which is the number of test cases. On each of the next T lines there will be three integers: m , the magnification level, followed by x and y , the position of the grid cell that Alice wishes to examine.

Output Specification

The output will be T lines. Each line of output will either be `empty`, if the specified grid cell is empty, or `crystal` if that grid cell contains crystal.

Sample Input

```
4
1 1 1
1 1 0
1 2 1
2 8 5
```

Output for Sample Input

```
empty
crystal
crystal
crystal
```

Note: At least 40% of the test cases will have $m \leq 4$.

Problem S4: Blood Distribution

Problem Description

At the Canadian Cardiac Centre there are four types of blood available: O, A, B, and AB. Each of these types of blood has an Rh factor, which is either “positive” or “negative”. There are many patients who each require 1 unit of blood. Each patient’s blood type determines the type of blood s/he may receive:

- Each Type O patient requires Type O blood.
- Each Type A patient may receive blood of Type A or Type O.
- Each Type B patient may receive blood of Type B or Type O.
- The Type AB patients may receive blood of any type.

Patients who have Rh-negative blood can accept Rh-negative blood only, but patients with Rh-positive blood can accept either Rh-positive or Rh-negative types of blood.

We want as many patients as possible to receive a unit of blood. What is the maximum number of patients that can receive blood?

Input Specification

The first line of input contains 8 integers: the number of units of blood of Type O negative, O positive, A negative, A positive, B negative, B positive, AB negative and AB positive. This is followed by a line containing eight numbers: the number of patients whose blood type is O negative, O positive, A negative, A positive, B negative, B positive, and AB negative and AB positive. Each of these integers will be at least 0 and less than 10^7 .

Output Specification

The output of your program should be a single number: the maximum number of patients that can receive blood.

Sample Input

```
5 5 3 1 2 11 5 12
2 4 9 2 3 9 7 3
```

Output for Sample Input

```
33
```

An Explanation

- 2 Type O- patients receive Type O- blood

- 4 Type O+ patients receive Type O+ blood
- 3 Type A- patients receive Type A- blood
- 3 Type A- patients receive Type O- blood
- 1 Type A+ patients receive Type A+ blood
- 1 Type A+ patients receive Type O+ blood
- 2 Type B- patients receive Type B- blood
- 9 Type B+ patients receive Type B+ blood
- 5 Type AB- patient receives Type AB- blood
- 3 Type AB+ patients receive Type AB+ blood

Note: At least 30% of the test cases for this problem will have at most 1000 units of each type of blood.

Problem S5: Switch

Problem Description

You are walking by a row of K ($4 \leq K \leq 25$) lights, some of which are on and some of which are off. In this initial configuration, there is no consecutive sequence of four lights that are on.

Whenever four or more consecutive lights are on, the lights in that consecutive block will turn off.

You can only turn on lights that are off.

What is the minimum number of lights you need to turn on in order to end up with all K lights off?

Input Description

The first line of input will consist of the integer K , indicating the number of lights. Each of the next K lines will have either the integer 0 (to represent a light that is off) or the integer 1 (to represent a light that is on).

Output Specification

Your program should output the minimum number of lights that must be turned on in order to have all K lights be off.

Sample Input 1

```
5
1
1
0
1
1
```

Output for Sample Input 1

```
1
```

Explanation of Sample 1

Notice that turning on the third light will create five consecutive lights that are on, which will in turn cause all of these five lights to be off.

Note: At least 30% of the test cases will have $K \leq 10$.