The CENTRE for EDUCATION
in MATHEMATICS and COMPUTING

2016 Canadian Computing Competition: Senior Division

Sponsor:

WATERLOO
MATHEMATICS

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:
   - if your supervising teacher is grading your solutions, all input is from a file named `sX.in`, where `X` is the problem number ($1 \leq \texttt{X} \leq 5$);
   - if you are using the On-line CCC grader, all input is from standard input;
   - all output is to standard output (i.e., to the screen).

   There is no need for prompting. Be sure your output matches the expected output in terms of order, spacing, etc. IT MUST MATCH EXACTLY!
5. The CCC is designed with the goal of having at least some marks for each problem that are obtainable for most competitors. Read and consider solutions to all problems.
6. Do your own work. Cheating will be dealt with harshly.
7. Do not use any features that the judge (your teacher or the On-line Grader) will not be able to use while evaluating your programs. In particular, take note of the type and version of the compiler used for your programming language on the On-line Grader if you are using the On-line Grader.
8. 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.
9. Applications other than editors, compilers, debuggers or other standard programming tools are **not** allowed. Any use of other applications will lead to disqualification.
10. If your teacher is grading, please use file names that are unique to each problem: use `s1.pas` or `s1.c` or `s1.java` (or some other appropriate extension) for Problem S1. If you are using the On-line Grader, follow naming rules described there (and take particular note of file and class names for Java programs).
11. 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.

12. The Senior problems shall be given 5 seconds of execution time per test case (if graded in schools), and similarly, 5 seconds of execution time on the on-line grader (if graded using the on-line grader). If this time limit is exceeded, no points will be awarded for that test case.

13. If you finish in the top 20 competitors on this competition, you will be invited to participate in Canadian Computing Olympiad (CCO), held at the University of Waterloo in May 2016. We will select the Canadian International Olympiad in Informatics (IOI) team from among the top contestants at the CCO. You should note that IOI 2016 will be held in Russia. Note that you will need to know C, C++, Java or Pascal if you are invited to the CCO. But first, do well on this contest!

14. 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/contests/computing.html`

Problem S1: Ragaman

Problem Description
An *anagram* of a string is formed by rearranging the letters in the string. For example, the anagrams of *aab* are *aab*, *aba*, and *baa*.

A *wildcard anagram* of a string is an anagram of the string where some of the letters might have been replaced with an asterisk (*). For example, two possible wildcard anagrams of *aab* are **ab* and **b**.

Given two strings, determine whether the second string is a wildcard anagram of the first string.

Input Specification
The two lines of input will both consist of N ($1 \leq N \leq 100$) characters. Each character in the first line will be a lowercase letter. Each character in the second line will be either a lowercase letter or an asterisk.

For 8 of the 15 available marks, the second line will not contain any asterisk characters.

Output Specification
Output the character `A` if the string on the second line is a wildcard anagram of the string on the first line. Otherwise, output the character `N`.

Sample Input 1
```
abba
baaa
```

Output for Sample Input 1
```
N
```

Sample Input 2
```
cccrocks
socc*rk*
```

Output for Sample Input 2
```
A
```

Problem S2: Tandem Bicycle

Problem Description
Since time immemorial, the citizens of Dmojistan and Pegland have been at war. Now, they have finally signed a truce. They have decided to participate in a tandem bicycle ride to celebrate the truce. There are N citizens from each country. They must be assigned to pairs so that each pair contains one person from Dmojistan and one person from Pegland.

Each citizen has a cycling speed. In a pair, the fastest person will always operate the tandem bicycle while the slower person simply enjoys the ride. In other words, if the members of a pair have speeds a and b, then the *bike speed* of the pair is $\max(a, b)$. The *total speed* is the sum of the N individual *bike speeds*.

For this problem, in each test case, you will be asked to answer one of two questions:

- Question 1: what is the minimum total speed, out of all possible assignments into pairs?
- Question 2: what is the maximum total speed, out of all possible assignments into pairs?

Input Specification
The first line will contain the type of question you are to solve, which is either 1 or 2.

The second line contains N ($1 \leq N \leq 100$).

The third line contains N space-separated integers: the speeds of the citizens of Dmojistan.

The fourth line contains N space-separated integers: the speeds of the citizens of Pegland.

Each person's speed will be an integer between 1 and $1\ 000\ 000$.

For 8 of the 15 available marks, questions of type 1 will be asked. For 7 of the 15 available marks, questions of type 2 will be asked.

Output Specification
Output the maximum or minimum total speed that answers the question asked.

Sample Input 1
```
1
3
5 1 4
6 2 4
```

Output for Sample Input 1
```
12
```

Explanation for Output for Sample Input 1
There is a unique optimal solution:

- Pair the citizen from Dmojistan with speed 5 and the citizen from Pegland with speed 6.
- Pair the citizen from Dmojistan with speed 1 and the citizen from Pegland with speed 2.
- Pair the citizen from Dmojistan with speed 4 and the citizen from Pegland with speed 4.

Sample Input 2
```
2
3
5 1 4
6 2 4
```

Output for Sample Input 2
```
15
```

Explanation for Output for Sample Input 2
There are multiple possible optimal solutions. Here is one optimal solution:

- Pair the citizen from Dmojistan with speed 5 and the citizen from Pegland with speed 2.
- Pair the citizen from Dmojistan with speed 1 and the citizen from Pegland with speed 6.
- Pair the citizen from Dmojistan with speed 4 and the citizen from Pegland with speed 4.

Sample Input 3
```
2
5
202 177 189 589 102
17 78 1 496 540
```

Output for Sample Input 3
```
2016
```

Explanation for Output for Sample Input 3
There are multiple possible optimal solutions. Here is one optimal solution:

- Pair the citizen from Dmojistan with speed 202 and the citizen from Pegland with speed 1.
- Pair the citizen from Dmojistan with speed 177 and the citizen from Pegland with speed 540.
- Pair the citizen from Dmojistan with speed 189 and the citizen from Pegland with speed 17.
- Pair the citizen from Dmojistan with speed 589 and the citizen from Pegland with speed 78.
- Pair the citizen from Dmojistan with speed 102 and the citizen from Pegland with speed 496.

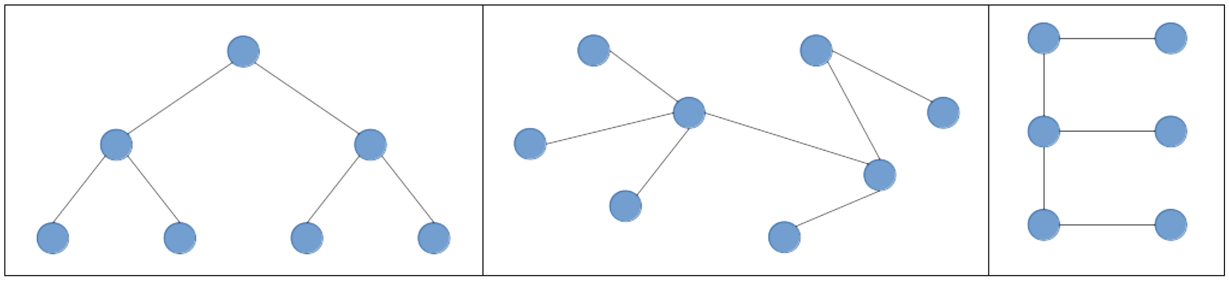
This sum yields $202 + 540 + 189 + 589 + 496 = 2016$.

Problem S3: Phonomenal Reviews

Problem Description
Jo is a blogger, specializing in the critique of restaurants. Today, she wants to visit all the Vietnamese Pho restaurants in the Waterloo area, in order to determine which one is the best.

There are N restaurants in the city of Waterloo, numbered 0 to N-1. However, only M of them are Pho restaurants. Jo can choose to start at any restaurant. There are N-1 roads in Waterloo, each road connecting exactly two restaurants. It is possible to reach every restaurant from any restaurant using these roads. It takes Jo exactly 1 minute to travel along any road.

In computer science, a road network with this structure is called a *tree*. Here are three examples of trees:



One property that is true for *all* trees is that there is exactly one path that does not repeat any roads between any two points in the tree.

What is the minimal length of time that Jo needs to spend on travelling on roads to visit all of the Pho restaurants?

Input Specification
The first line of input contains 2 integers, N and M ($2 \leq M \leq N \leq 100\,000$).

The second line of input contains M distinct integers indicating the restaurants which are Pho restaurants.

The next N-1 lines contain 2 integers each. The i-th line contains a_i and b_i, ($0 \leq a_i, b_i \leq N-1$), representing a path between the two restaurants numbered a_i and b_i.

For 3 of the 15 available marks, $M = 2$ and $N \leq 100$.
For an additional 3 of the 15 available marks, $M \leq 3$ and $N \leq 100$.
For an additional 3 of the 15 available marks, $M \leq 8$ and $N \leq 100$.
For an additional 4 of the 15 available marks, $M \leq N \leq 1000$.

Output Specification
Your program should output one line, containing one integer - the minimum amount of time Jo needs to spend travelling on roads in order to visit all Pho restaurants, in minutes.

Sample Input 1

```
8 2
5 2
0 1
0 2
2 3
4 3
6 1
1 5
7 3
```

Output for Sample Input 1

```
3
```

Explanation for Output for Sample Input 1

The path between 5 and 2 goes through $5 \rightarrow 1 \rightarrow 0 \rightarrow 2$, which uses 3 roads.

Sample Input 2

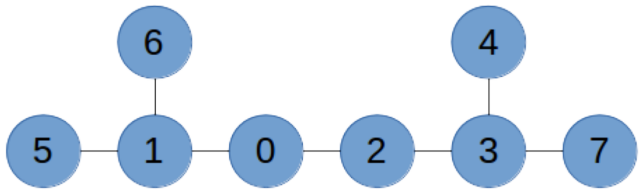
```
8 5
0 6 4 3 7
0 1
0 2
2 3
4 3
6 1
1 5
7 3
```

Output for Sample Input 2

```
7
```

Explanation for Output for Sample Input 2

If Jo begins at restaurant 6, she will only need to use 7 roads. One possible path that she can take is: $6 \rightarrow 1 \rightarrow 0 \rightarrow 2 \rightarrow 3 \rightarrow 7 \rightarrow 3 \rightarrow 4$. Notice that she doesn't need to visit restaurant 5, since it is not a Pho restaurant. A diagram of the road network is shown below:



Problem S4: Combining Riceballs

Problem Description
Alphonse has N rice balls of various sizes in a row. He wants to form the largest rice ball possible for his friend to eat. Alphonse can perform the following operations:

- If two **adjacent** rice balls have the same size, Alphonse can combine them to make a new rice ball. The new rice ball's size is the sum of the two old rice balls' sizes. It occupies the position in the row previously occupied by the two old rice balls.
- If two rice balls have the same size, and there is **exactly one rice ball between them**, Alphonse can combine all three rice balls to make a new rice ball. (The middle rice ball does not need to have the same size as the other two.) The new rice ball's size is the sum of the three old rice balls' sizes. It occupies the position in the row previously occupied by the three old rice balls.

Alphonse can perform each operation as many times as he wants.

Determine the size of the largest rice ball in the row after performing 0 or more operations.

Input Specification
The first line will contain the integer, N ($1 \leq N \leq 400$).

The next line will contain N space separated integers representing the sizes of the riceballs, in order from left to right. Each integer is at least 1 and at most 1 000 000.

- For 1 of the 15 available marks, $N = 4$.
- For an additional 2 of the 15 available marks, $N \leq 10$.
- For an additional 5 of the 15 available marks, $N \leq 50$.

Output Specification
Output the size of the largest riceball Alphonse can form.

Sample Input 1
```
7
47 12 12 3 9 9 3
```

Output for Sample Input
```
48
```

Explanation for Output for Sample Input 1
One possible set of moves to create a riceball of size 48 is to combine 12 and 12, forming a riceball

of size 24. Then, combine 9 and 9 to form a riceball of size 18. Then, combine 3, 18 and 3 to form a riceball of size 24. Finally, combine the two riceballs of size 24 to form a riceball of size 48.

Sample Input 2

```
4
1 2 3 1
```

Output for Sample Input 2

```
3
```

Explanation for Output for Sample Input 2

There are no moves to make, thus the largest riceball in the row is size 3.

Problem S5: Circle of Life

Problem Description

You may have heard of *Conway's Game of Life*, which is a simple set of rules for cells on a grid that can produce incredibly complex configurations. In this problem we will deal with a simplified version of the game.

There is a one-dimensional circular strip of N cells. The cells are numbered from 1 to N in the order you would expect: that is, cell 1 and cell 2 are adjacent, cell 2 and cell 3 are adjacent, and so on up to cell $N - 1$, which is adjacent to cell N. Since the strip is circular, cell 1 is also adjacent to cell N.

Each cell is either alive (represented by a '1') or dead (represented by a '0'). The cells change over a number of generations. If **exactly** one of a cell's neighbours is alive in the current generation, then the cell will be alive in the next generation. Otherwise, the cell will be dead in the next generation.

Given the initial state of the strip, find the state after T generations.

Input Specification

The first line will contain two space-separated integers N and T ($3 \leq N \leq 100\,000$; $1 \leq T \leq 10^{15}$). The second line will contain a string consisting of exactly N characters, representing the initial configuration of the N cells. Each character in the string will be either '0' or '1'. The initial state of cell i is given by the i-th character of the string. The character '1' represents an alive cell and the character '0' represents a dead cell.

For 1 of the 15 available marks, $N \leq 15$ and $T \leq 15$.

For an additional 6 of the 15 available marks, $N \leq 15$.

For an additional 4 of the 15 available marks, $N \leq 4000$ and $T \leq 10\,000\,000$.

Note that for full marks, solutions will need to handle 64-bit integers. For example

- in C/C++, the type `long long` should be used;
- in Java, the type `long` should be used;
- in Pascal, the type `int64` should be used.

Output Specification

Output the string of N characters representing the final state of the cells, in the same format and order as the input.

Sample Input 1

```
7 1
0000001
```

Output for Sample Input 1

```
1000010
```

Explanation for Output for Sample Input 1

Cell 1 and cell $N - 1$ are adjacent to cell N, and thus are alive after one generation.

Sample Input 2

```
5 3
01011
```

Output for Sample Input 2

```
10100
```

Explanation for Output for Sample Input 2

After one generation, the configuration becomes `00011`.

After two generations, the configuration becomes `10111`.