Spring 2018 ACM Programming Contest

Overview

Welcome to the Spring 2018 ACM Programming Competition! This is a contest wherein you compete amongst your fellow students, and professors, in an effort to solve a set of programming problems as fast as possible.

Problems

There are eight problems of (roughly) increasing difficulty. The team who solves the most problems is declared the winner. In the event of a tie, the team with the best time wins. The best time also included penalties for wrong submissions. Each problem has a general description, details on the formats for input and output, and example input and output.

Input is done from standard input and output is done to standard output. *Do not open any files inside of your program.* Output of your program must match that of the correct output *exactly*. You can solve the problems in C, C++, Haskell, Java, PHP, Python, Racket, Ruby, or R. Up to date versions of these languages will be used to judge your entries. You should save your code using the standard extension for your language of choice (.c .cpp .hs .java .php .py .rkt .rb and .R respectively).

At the end of your packet, there is also a brief overview of how to perform input and output in C++, Python and Java in case you need a refresher.

Rules

You are allowed to have any printed material such as books or printouts. You are also allowed to access standard library documentation for your language of choice. *No other use of the internet is permitted.* You are also not allowed to copy and paste code from any source be it a thumb drive, or another file that you have. Only one computer is allowed to a team.

Logging In

To log in to the system, direct your browser to `http://bit.ly/2GXI5ri`. You will then need to enter your team name and password which are provided for you. This will bring up the interface for you to submit solutions to the problems, ask questions (which will be broadcast to all participants), and see how your team is doing.

Submitting

To submit a solution, first make sure that you select the correct problem letter on the top of the site. Then click the "Choose File" button. Then choose the file that contains your source code (be sure it has the correct extension). After that it will be submitted for judging. It will receive an automatic judging response which is marked pending. Your solution will then be checked manually and a final response will be assigned.

DO NOT TURN OVER THIS SHEET UNTIL THE CONTEST STARTS

Problem A: Petal Timer

One cold night, a prince refused to take in a wandering beggar. The beggar turned out to be a sorceress, and laid a curse on the prince and his household in retribution. The prince was transformed into a fearsome beast, and his servants were transformed into household objects.

The sorceress also gave the prince an enchanted rose. The petals of the rose will fall off one by one to mark the passing of time. If the beastly prince can find true love before all of the petals fall off, he will be transformed back into a human. If not, he will be trapped as a beast forever.

Naturally, the beast would like to know exactly how long he has. He has asked you to write a program which will be given how many petals the rose has, and how many days each petal will last before falling. Your program will then output how many days the beast has left to find true love. You are to assume that all petals will take the same amount of time to fall.

Input

Input consists of two lines. On the first line is P, the number of petals on the rose given to the beast. On the second line is D, the number of days it takes for one petal to fall off of the rose.

Output

Output will be a single line containing the sentence "The beast has X days to find true love!", where X is the number of days before all of the petals fall off of the rose. You can assume that X will be more than 1.

Sample Input

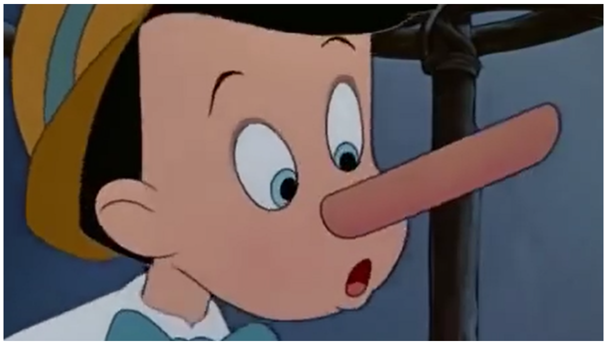
```
7
21
```

Sample Output

```
The beast has 147 days to find true love!
```

Problem B: Count The Lies

After being magically given life, the puppet Pinocchio is able to walk and talk. If he is able to prove himself brave and honest, he will be turned into a real boy. While he is still a puppet, every lie he tells has an unusual effect – it makes his nose grow one half inch. When getting in trouble, Pinocchio tells several lies making his nose grow quite long.



You should write a program which will figure out how long Pinocchio's nose will be after telling a given number of lies. It starts as one half inch long, and grows another half inch for each lie he tells.

Input

Input will consist of a single non-negative integer L, giving the number of lies that Pinocchio has told.

Output

Output should be one line printing the length of Pinocchio's nose.

If the length is a whole number, then do not print the number of half inches. Also, be sure to get the pluralization correct – if Pinocchio's nose is one inch, or one half inch, then do not pluralize the word inch.

Sample Input 1

```
5
```

Sample Output 1

```
Pinocchio’s nose is 3 inches.
```

Sample Input 2

```
12
```

Sample Output 2

```
Pinocchio’s nose is 6 and 1 half inches.
```

Problem C: Cruella's Coat Emporium

Cruella de Vil is high-end fashion designer specializing in fur coats. In anticipation of receiving a large shipment of puppies, she is designing an order form for puppy fur coats. The coats come in three sizes: small medium and large. A small coat is made from 5 puppy skins, a medium one is made from 7, and a large one is made from 9.

Of course using puppies for coats is not economical, but their soft fur is what Cruella de Vil's customers have come to expect.

Customers indicate how many coats of each size that they are ordering. Cruella then calculates the cost of the coats. She does this by calculating how many puppy skins are needed in total. She then uses the following cost per skin based on how many the customer needs:

- 15 or fewer skins – \$300 per skin.
- 16 - 30 skins – \$280 per skin
- 31 or more skins – \$250 per skin.

This encourages customers to place larger orders.

Cruella has asked you to write a program to calculate the total cost of a customer's order (she has actually blackmailed you into it, but that's not important now).

Input

Input will consist of three lines. The first line contains an integer S giving the number of small coats. The second line contains an integer M giving the number of medium coats. The third line contains an integer L giving the number of large coats.

Output

Output will be a single line containing the string “Total comes to $\$X$.” where X is the number of dollars the customer’s order will cost according to the pricing scheme above.

Sample Input

```
1
2
1
```

Sample Output

```
Total comes to $7840.
```

Problem D: Wandering Pirates

The pirate known as Captain Hook plunders the seas surrounding Never Land. He is aided by his trusty bumbling sidekick, Mr. Smee. Captain Hook instructed Mr. Smee to bury some treasure on the coast of Never Land. He couldn't do it himself as he was worried the crocodile who pursues him would turn up.

Mr. Smee buried the treasure, but didn't do a very good job making directions to find the treasure again. In making his directions, Mr. Smee has the pirates crossing over their own path unnecessarily, and turning more than is needed. Each line of the directions either says to walk forward some number of paces, or turn to face a different direction.

Captain Hook wants you to write a program which will read in Mr. Smee's directions to the treasure and print the location of the treasure relative to the pirate's current location.

The pirates begin facing north.

Input

The first line of input will contain an integer N giving the number of lines in Mr. Smee's directions.

Following that will be N lines, each giving one direction. Each of these directions will say either "Walk P paces", where P is a positive integer; or "Turn DIR" where DIR is one of north, south, east, or west.

Output

Output should consist of one line giving the final location of the treasure relative to the starting position. It should have the form "Treasure is Y paces to the [north/south] and X paces to the [east/west]."

If either X or Y are zero, you should not print that part of the message (as in Sample Output 2). If both X and Y are 0, you should simply output "Treasure is right here."

Sample Input 1

```
7
Walk 4 paces
Turn east
Walk 10 paces
Turn south
Walk 6 paces
Turn east
Walk 2 paces
```

Sample Output 1

```
Treasure is 2 paces to the south and 12 paces to the east.
```

Sample Input 2

```
8
Turn east
Walk 4 paces
Turn west
Walk 7 paces
Turn north
Walk 5 paces
Turn east
Walk 3 paces
```

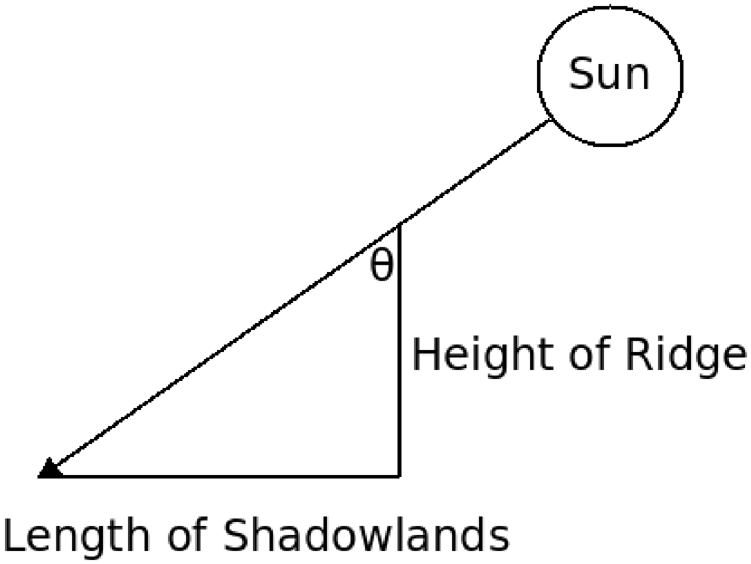
Sample Output 2

```
Treasure is 5 paces to the north.
```

Problem E: Measuring the Shadowlands

Mufasa tells his son Simba that their kingdom contains all of the land that the light touches. This does not include a shadowy place outside of their kingdom which apparently never directly receives sunlight. Mufasa would like to measure the size of these Shadowlands in order to know the true size of his kingdom.

For the light to truly never touch this part of the savannah, there must be a mountain ridge which blocks the sun from reaching the ground. The sun must never go directly overhead, but must always be to the far side. The size of the Shadowlands are then defined by the height of the mountain ridge and the angle, θ, at which the sun reaches the mountain ridge. This is depicted in the following diagram:



Of course the angle of the sun will vary throughout the year. Mufasa has asked you to write a program which will read in the height of the mountain ridge, along with a number of readings of the angle θ throughout the year. Your program must then calculate the minimum and maximum length of the Shadowlands over the year.

Recall that the tangent of an angle is equal to the side opposite the angle divided by the side adjacent to that angle.

Input

The first line of input will be a real number giving the height of the mountain ridge in meters. The second line will contain an integer, N, giving the number of angle readings. Following that will be N lines each containing one real number for a value of θ, in degrees. θ will be greater than 0 and less than 90.

Output

Your program should output one line saying "The Shadowlands are between X and Y meters.", where X is the minimum length and Y is the maximum length. Both X and Y should be rounded to the nearest integer.

Sample Input

```
100.5
6
52.4
61.1
57.9
49.8
51.6
54.9
```

Sample Output

```
The Shadowlands are between 119 and 182 meters.
```

Problem F: Smaller or Larger

While Alice was chasing the white rabbit, she comes upon a potion labelled “Drink Me”. Upon drinking the potion she shrinks to tiny fraction of her original size. Later, she eats a biscuit labelled “Eat Me” which makes her grow much larger than her true size.

Alice has once again drunk a potion which has changed her original size, and you need to help her get back to her original size. She is in a room with a number of other potions which are labelled to indicate how they will change Alice’s size. She is only allowed to drink one or two additional potions to return to her original size (as in large quantities they are poisonous).

You will write a program to determine which potion or pair of potions Alice should drink, or determine there is no solution. For instance, in the case where Alice is at one third of her size, she could drink a potion to triple her size. In the case where Alice is twice her original size, she could drink a potion which divides her size by six, followed by another which triples her size.

Input

The first line of input indicates how the first potion changed Alice’s size. It consists of either the character / or *, followed by a space, and then an integer. If the character is a / it indicates that the potion made Alice *smaller*. For instance, if the first line says `/ 3`, the initial potion shrunk Alice to one third of her original size. If the character is a *, then Alice grew larger after drinking the potion. If the first line is `* 2`, then she is twice as big as she originally was.

The second line of input is an integer N giving the number of potions Alice has to choose from to get back to her original size. Following that are N lines, each giving the sizing effect of the N potions, using the same format as the first input line.

Output

If there is one potion that Alice can drink which will restore her size, your program should output “Alice should drink potion X.” where X is the number of the of the potion as it appeared in the input. The potions are numbered from 1 to N.

If there is a pair of potions that Alice can drink which will restore her size, your program should output "Alice should drink potions X and Y." where X and Y are the potion numbers in the input. The potions should be output in ascending order – that is $X < Y$.

If there is a solution with one potion, and one with two potions, your program should output the one potion solution – Alice would prefer to only drink one potion if she can. If there are multiple solutions of any type, you should output the one with the smallest potion numbers.

Your program should output "There is no solution." if there is no potion or pair of potions which can restore her to her original size.

Sample Input 1

```
/ 5
3
/ 2
* 5
* 4
```

Sample Output 1

```
Alice should drink potion 2.
```

Sample Input 2

```
* 2
4
/ 3
* 3
* 4
/ 6
```

Sample Output 2

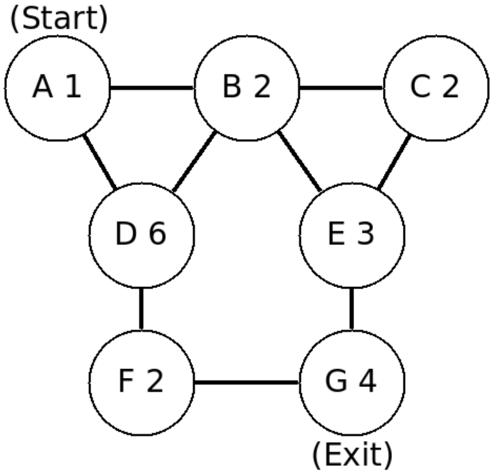
```
Alice should drink potions 2 and 4.
```

Problem G: Escape the Cave of Wonders

Aladdin is tricked by Jafar into retrieving the magic lamp from the Cave of Wonders. Aladdin knows not to touch anything except for the lamp – touching any of the other fabulous treasures will cause the cave to collapse on itself. Aladdin finds the lamp and picks it up to return it to Jafar.

Unfortunately just as Aladdin is retrieving the lamp, his pet monkey Abu is unable to resist a large red ruby. When he picks it up, the cave begins to collapse. The Cave of Wonders contains a number of rooms connected together by passageways. The rooms collapse on themselves at different times. In his escape, Aladdin needs to reach the exit without going through a room which has already collapsed, or being in a room while it collapses. You need to write a program which will read in the structure of a cave including when each room collapses, and determine if and how he can reach the cave's exit.

Below is an example cave. Aladdin begins in room A and must get to the exit, room G. The numbers refer to the step after which the room collapses. So room A will collapse after Aladdin's first step, and room B after his second. Aladdin always takes 1 step to travel from one room to another.



Aladdin can reach rooms B and D immediately. If he goes to room D, he will not be able to reach the exit because room F will collapse after his second move (when he would reach room F), so he must start by going to room B. Likewise Aladdin cannot continue to room C, but must escape via rooms E then finally G. This case appears as Sample Input 1 below.

Sample Input 2 is identical, except that room E collapses after step 2 instead of 3. That makes the cave impossible to exit – in this case Aladdin has no escape!

Input

The first line of input contains an integer *N* giving the number of rooms in the cave. Following that are *N* lines, one for each room. Each of these lines contains an upper-case character which gives the name of the room, and an integer giving the step at which the room collapses.

The first room listed is the one in which Aladdin begins, and the last room is the one he must reach in order to escape.

Next is a line containing an integer *M* giving the number of passageways in the cave. Following that are *M* lines which give the passageways. Each of these lines consists of two characters, indicating which two rooms the passageway connects. The passageways are bi-directional and only listed for one direction – so BC can be used to travel from room B to C or from C to B.

Output

If there is a way for Aladdin to escape the cave, your program should print "Aladdin can escape: " followed by the sequence of moves by which Aladdin can escape the cave. The sequence should be given as a string of upper-case characters which indicate which rooms Aladdin travelled through (including the starting cave and exit). If there are multiple solutions, print any of them.

Your program should output "Aladdin can not escape!" if there is no escape from the cave.

Sample Input 1

```
7
A 1
B 2
C 2
D 6
E 3
F 2
G 4
```

```
9
AB
AD
BC
BD
BE
CE
DF
EG
FG
```

Sample Output 1

```
Aladdin can escape: ABEG
```

Sample Input 2

```
7
A 1
B 2
C 2
D 6
E 2
F 2
G 4
9
AB
AD
BC
BD
BE
CE
DF
EG
FG
```

Sample Output 2

```
Aladdin can not escape!
```

Problem H: Improved Conscription

When the Huns breached the Great Wall and invaded China, the Chinese army held a draft – one man from each family was required to join the ranks. Even though Fa Zhou was older and in poor health, he was required to enlist as the only male in his family. In order to protect her father, Fa Mulan went in his stead and proved to be a capable warrior herself.

Now that the conflict is resolved, the Chinese army is considering alternative ways of organizing the draft. The scheme they have chosen is to line all potential soldiers up in a line, grouped by family. They will then give each potential recruit a score based on his or her physical fitness. Finally they will choose those potential soldiers with the best scores, but without ever taking two recruits who are standing next to each other. This rule is in place to prevent ever taking more than half of the potential soldiers from one family (since families are standing next to each other). Captain Li Shang believes this method will yield a stronger army and be more equitable to Chinese families.

Captain Shang has asked you to write a program which will be given the list of the recruits physical fitness score, and select the recruits which gives the maximum total physical fitness score – while never selecting two consecutive potential soldiers.

Input

The first line of input contains an integer N giving the number of potential soldiers. The following N lines of input each give the overall physical fitness score of the N soldiers.

Output

Your program should output "The maximum strength of the army is X." where X is the maximum total physical fitness of the recruits.

Sample Input

```
7
34
87
94
86
39
42
96
```

Sample Output

```
The maximum strength of the army is 269.
```

Doing Input and Output

This guide contains the basic way of doing input and output in C++, Java, and Python, in case you need a quick refresher.

In programming contests, you should always do input from the terminal screen (e.g. using the keyboard), and do output to the terminal screen. You should not open any files for input or output.

You should also not output any prompts. If a problem instructs you to read a line containing a number, just read it in without any kind of prompt saying "Enter a number" or similar.

C++

Input

1. To skip over whitespace (spaces or new lines), and read in a single value (such as an integer or string), use
   `cin > > value;`. For example:

   ```
   // read one integer
   int number;
   cin >> number;

   // read one character string
   char str1[100];
   cin >> str1;

   // read one string object
   string str2;
   cin >> str2;
   ```

2. To read in one entire line of input, which may contain spaces, use `cin.getline`. For example:

   ```
   // read in a character string of up to 100 characters
   char str1[100];
   cin.getline(str1, 100)

   // read in a string object
   string str2;
   getline(cin, str2);
   ```

Output

Output is done with `cout < <` in C++ which can take any built-in data type. For example:

```
// print a message, then an integer, then a new line
int x;
cout << "X is equal to " << x << endl;
```

Python

Input

1. The `input` function in Python read in one line of input and returns it as a string. For example:

   ```
   # read in a string
   line = input()
   ```

2. In order to convert from a string to a number, you can use the `int` function for integers, or the `float` function for real numbers:

   ```
   # read in an integer by passing the input string to int()
   number = int(input())
   ```

3. In order to break a line of input into multiple strings, separated by spaces, you can use the `.split()` function which returns a list of strings:

   ```
   # read in a whole line
   line = input()

   # split it into separate strings based on spaces
   words = line.split()
   ```

Output

1. Output in Python is done with the `print` function which outputs all of its arguments, separated by spaces, and puts a newline at the end:

   ```
   # print a message, a number and a new line
   print("X is equal to", x)
   ```

2. In order to prevent `print` from putting spaces between each item, pass `sep=’ ’` as an argument:

   ```
   # now there is no space between message and value
   print("X is equal to", x, sep= )
   ```

3. In order to prevent `print` from putting a new line at the end, pass `end=’ ’` as an argument:

   ```
   # now there is no new line added
   print("X is equal to", x, end= )
   ```

Java

Input

Input in Java can be done with the `Scanner` class which must be imported first:

```
import java.util.Scanner;
```

Then a `Scanner` object must be created:

```
java.util.Scanner in = new java.util.Scanner(System.in);
```

1. To read in one line of input into a string, use the scanner's `nextLine` method:

   ```
   String line = in.nextLine();
   ```

2. To read in a single word into a String, stopping at a space, use the `next` method:

   ```
   String word = in.next();
   ```

3. To read a numerical value use the `nextInt` method for integers, or the `nextDouble` method for real numbers:

   ```
   int number1 = in.nextInt();
   double number2 = in.nextDouble();
   ```

Output

1. To output a string constant or variable, use the `System.out.println` function which takes one argument, prints it to the screen, then prints a new line:

   ```
   // print a message
   System.out.println("This will be printed")

   // print a value
   int x;
   System.out.println(x);
   ```

2. To output something without a new line at the end, use the `System.out.print` function which outputs its argument with no newline:

   ```
   // print a message with no new line
   System.out.print("The value of X is ")

   // print a value with no newline
   int x;
   System.out.print(x);
   ```