Spring 2015 ACM Programming Contest

Overview

Welcome to the Spring 2015 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 nine problems of (roughly) increasing difficulty. The team who solves the most problems is declared the winner. In the event of a tie, the team that finished their last problem earliest wins. 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, R, or Scratch. 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 .R .sb 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/1NKMdFr`. 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: Production Orders

> *"For the ground trembled. The shafts ran down by many slopes and spiral stairs to caverns far under; there Saruman had treasuries, store-houses, armouries, smithies, and great furnaces. Iron wheels revolved there endlessly, and hammers thudded. At night plumes of vapour steamed from the vents, lit from beneath with red light, or blue, or venomous green."*

During the War of the Ring, Saruman allied himself with Mordor and turned Isengard into a military base, cutting down its trees and grass and replacing them with machinery and forges. He diverted the Angren river to use the hydro-power to drive production.

Saruman used Uruk-hai both as soldiers, and as workers in his forges. Uruk-hai are a powerful race of Orc-men, but are not very intelligent. In particular, they are not able to count any higher than three. This makes it difficult for Saruman to place production orders, as every order greater than three must be broken into several smaller orders.

Saruman has asked you to write a program that will automatically generate work orders for his Uruk-hai foreman at the forge for producing swords. Your program will take as input the total number of swords needed, and break it down into a list of instructions for the foreman using quantities of at most three.

Input

The input will be a singe non-negative integer, N, giving the number of swords needed by Saruman.

Output

Output should be a sequence of lines each giving the foreman an order to make a certain number of swords. Each line should say to make 1, 2, or 3 swords. To reduce the number of instructions, you should use as many instructions to make 3 swords as possible, and give those instructions first, leaving any instructions to make less than three swords for the end of the output. The sum of all the swords together should be N. If you include an instruction to print one sword, you should not pluralize "swords". Do not include an instruction to print 0 swords, this will confuse the foreman!

Sample Input

```
10
```

Sample Output

```
Make 3 swords.
Make 3 swords.
Make 3 swords.
Make 1 sword.
```

Problem B: Arrows Aplenty

> *"And I," said Legolas, "will take all the arrows that I can find, for my quiver is empty." He searched in the pile and on the ground about and found not a few that were undamaged and longer in the shaft than such arrows as the Orcs were accustomed to use.*

Legolas was an Elf who was part of the Fellowship of the Ring in the Third Age. He is the son of Thranduil of Mirkwood, a Prince of Mirkwood, a messenger, and a master bowman. During the War of the Ring he slew many Orcs with his bow.

During a long battle, Legolas needs to scavenge arrows from his fallen enemies in order to restock his quiver. Every now and then, Legolas is unable to reclaim one of his arrows, either because it breaks on impact or because the target was too far away.

Based on the number of arrows Legolas begins with, and the number of arrows he can reclaim before one is unrecoverable, write a program to calculate how many enemies Legolas can defeat before needing to go back to an armory and fully restock his quiver.

You can assume that Legolas never misses a shot, and never needs more than one arrow to dispatch an enemy.

Input

The first line of input is a positive integer giving the number of arrows Legolas begins with. The second line of input is a positive integer giving the number of arrows Legolas is able to recover consecutively before missing one. For example, if this number is 3, Legolas will recover the first three arrows he shoots, then not be able to recover the fourth, then recover the next three, and so on.

Output

Output should be a single integer giving the number of enemies Legolas can defeat before running out of arrows.

Sample Input

```
15
3
```

Sample Output

```
60
```

Problem C: Light the Beacons

> *Aragorn: "The Beacons of Minas Tirith! The Beacons are lit! Gondor calls for aid."*
>
> *Theoden: "And Rohan will answer. Muster the Rohirrim. Assemble the army at Dunharrow. As many men as can be found. You have two days. On the third, we ride for Gondor and war."*

The warning beacons of Gondor were large fireplaces high up in mountains. When lit, a beacon could be seen from miles in all directions. The beacons formed a chain running from the kingdom of Rohan to the kingdom of Gondor and were used to communicate warnings between the two kingdoms.

Each beacon had a keeper who lived at the beacon and was in charge of lighting it. When a keeper sees an adjacent beacon is lit, he lights his own beacon in response. This triggers the next beacon in the chain to light and so on. In the third age, when Gondor was under siege from Mordor, the Hobbit Pippin Took lit the beacon in Gondor which triggered the beacon chain until the message was received in Rohan causing the Rohirrim to come to Gondor's aid.

When either Rohan or Gondor lights its own beacon, there is a delay before the beacon in the opposite kingdom is eventually lit. This delay is based on the number of beacons in the chain, the time it takes to actually light the beacon, as well as the time before each beacon keeper actually sees that the adjacent beacon is lit (after all, the keeper needs some time to sleep).

You will write a program that calculates the minimum and maximum time it could take for a message to propagate across the beacon chain based on these factors. A beacon always takes ten minutes to light. The number of beacons in the chain and the maximum consecutive time each beacon keeper sleeps at one time will be input to your program.

Input

The first line of input contains an integer giving the number, N of beacons in the chain. Following that, there will be N lines of input each giving an integer which represents the maximum number of hours that that beacon keeper sleeps. The first such line is for the first beacon that is lit, the second for the second, and so on.

Output

Your program should output the minimum and maximum time it could take for a message to propagate across the beacon chain – that is the time between the first beacon is lit and the final beacon is lit – in hours and minutes.

You should use correct plurality on the words "hours" and "minutes". If either the number of minutes or hours is zero, you should not include that in your output. For example "The minimum time is 1 hour." is correct, but "The minimum time is 1 hours and 0 minutes." is not.

Sample Input

```
5
6
4
8
6
5
```

Sample Output

```
The minimum time is 40 minutes.
The maximum time is 23 hours and 40 minutes.
```

Problem D: Elvish Translation

> *"The opening word was inscribed on the archway all the time! The translation should have been: Say 'Friend' and enter. I had only to speak the Elvish word for friend and the doors opened. Quite simple. "*

One of the Elvish languages spoken in Middle Earth is Sindarin, called "Grey-Elvish" by men, which is spoken by the Elves of Beleriand. These were Elves of the Third Clan who remained behind in Beleriand after the Great Journey.

A Gondorian merchant has asked you to write a program which will translate simple Sindarin words and phrases into the Common Speech to aid him in his travels through Elvish lands. As neither he nor you know very much Sindarin, your program will only translate a subset of the Sindarin language, as described below.

Your program should translate the following Sindarin words into their Common Speech meaning:

Sindarin Word	**Common Speech Meaning**
amar	earth
menel	sky
nen	water
naur	fire
benn	man
bess	woman
mad	eat
sog	drink
beleg	big
aur	day
du	night

Your program should also recognize the following multi-word phrases:

Sindarin Phrase	**Common Speech Meaning**
nen menel	rain
nen amar	sea
nen lunt	boat
nen naur	alcohol
aur o du	evening
du o aur	morning

Phrases take precedence over single words, so you should translate “nen menel” as “rain” instead of “water sky”.

Any Sindarin words that are not recognized should be output just as they are, as should all whitespace and punctuation.

Input

Input consists of a single line of text containing multiple Sindarin language words. You can assume all letters will be lower-case.

Output

Output should consist of a single line of text containing the translation according to the rules above.

Sample Input

```
soger du o aur revia nen lunt ah pan benn a bess ath nen amar an valinor an mad a sog.
```

Sample Output

```
soger morning revia boat ah pan man a woman ath sea an valinor an eat a drink.
```

Problem E: Seating Arrangements

> *Each year the Bagginses had given very lively combined birthday-parties at Bag End; but now it was understood that something quite exceptional was being planned for that autumn. Bilbo was going to be eleventy-one, 111, a rather curious number and a very respectable age for a Hobbit.*

Bilbo Baggins eleventy-first birthday is coming up, and Frodo and Sam are in charge of all the party planning. They have invited all the Hobbits of Hobbiton to the party, and everyone will be sitting in a single row at an extremely long dining table.

However, due to poor communication, Frodo and Sam have each independently put together a seating chart for all the Hobbits at the dining table. Positions higher up the table are seen as being more valuable, so the relative ordering of each pair of Hobbits should be done according to which is most worthy of being given a more honored seat at the table.

Help Frodo and Sam find out how similar their seating charts are by counting the total number of distinct pairs of Hobbits who appear in different orders in the two charts. For example, if Merry is higher up the table than Pippin on Sam's chart, but lower on Frodo's chart, then that pair is in a different order.

Input

The first line of input will be a positive integer N giving the number of Hobbits invited to the party. The next two lines represent Frodo's and Sam's seating charts, respectively. Each seating chart is specified as a single line of N unique alphabetical strings; the set of strings in each line are guaranteed to be identical.

Output

Output should be a single integer denoting, out of the N choose 2 distinct pairs of Hobbits, how many pairs appear in different orders in Frodo's and Sam's seating arrangements.

Sample Input

```
5
Frodo Pippin Sam Merry Halfast
Pippin Frodo Merry Halfast Sam
```

Sample Output

```
3
```

Problem F: Outpost Separation

> *"Then at night fresh forces came over the river against our camp. All Isengard must be emptied; and Saruman has armed the wild hillmen and herd-folk of Dunland beyond the rivers, and these also he loosed upon us. We were overmastered. The shield-wall was broken. Erkenbrand of Westfold has drawn off those men he could gather towards his fastness in Helm's Deep. The rest are scattered."*

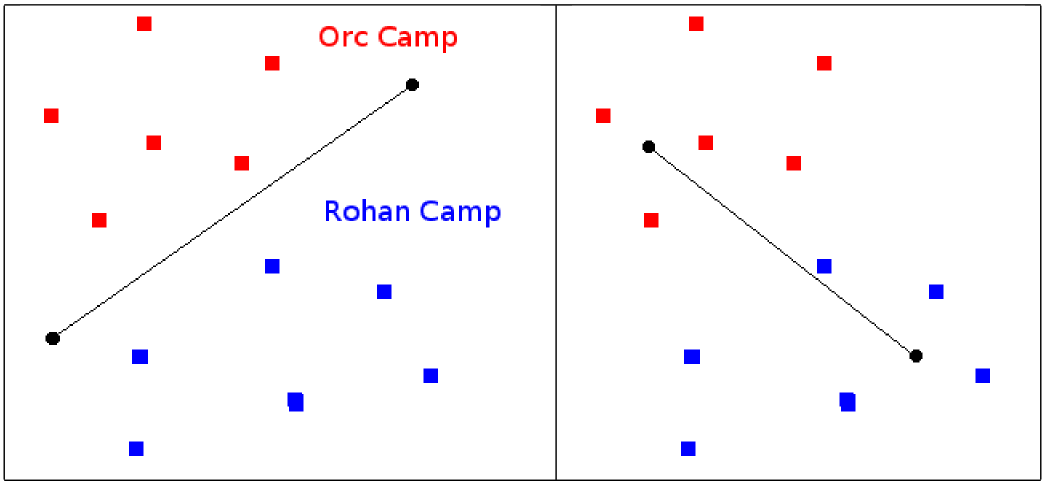
During the third age, the wizard Saruman began amassing an army of Orcs around Isengard with the intent of invading the Kingdom of Rohan. The wizard Gandalf, when alerted to this danger, rode to Rohan to warn King Theoden of this threat. Theoden, after taking some convincing of the danger Saruman posed, gathered his armies as well.

Both the men of Rohan and Saruman's Orcs of Isengard hold a camp in central Middle Earth. Each camp consists of a set of outposts established at a location which is represented by an x,y coordinate.

The men of Rohan wish to build a wall to barricade the advance of the Orcs into their camp. For the wall to be effective, it must completely separate the two camps. That means every Orcish outpost is on one side of the wall and every Rohan outpost is on the other.

You must write a program that, given the location of each of the outposts of each camp, and the location of a potential wall, determines whether or not the wall will completely separate the two camps or not.

Below are two examples. On the left, the wall does completely separate the two camps. On the right, the wall does not.



Input

The first line of input contains two integers. The first is the number of outposts in the Orcish camp, O. The second is the number of outposts in the Rohan camp, R.

After this first line, there will be O lines of input giving the x, y coordinates of each of the Orcish outposts.

Next, there will be R lines of input giving the x, y coordinates of each of the Rohan outposts.

The last two lines of output will be the 2 x, y coordinates of the end points of the wall.

You can assume that a wall is built as a straight line and can be extended infinitely in both directions. You can also assume all coordinates will be given as integers. All x, y coordinate pairs are separated by one space.

Output

Output consists of a single line that either says "Wall separates camps." or "Wall does not separate camps."

Sample Input 1

```
5 4
7 7
2 14
13 16
23 5
6 27
15 26
24 16
28 27
2 40
2 34
30 4
```

Sample Output 1

```
Wall separates camps.
```

Sample Input 2

```
4 3
6 12
18 10
12 4
16 22
30 15
27 5
34 9
32 4
22 28
```

Sample Output 2

```
Wall does not separate camps.
```

Problem G: Goblin Extermination

> *"The Dwarves tell no tale; but even as mithril was the foundation of their wealth, so also it was their destruction: they delved too greedily and too deep, and disturbed that from which they fled, Durin's Bane. Of what they brought to light the Orcs have gathered nearly all, and given it in tribute to Sauron, who covets it."*

In the third age, the Dwarves in the mines of Moria began delving more deeply and more greedily in their search for Mithril. In doing so, they unearthed a Balrog which destroyed the city of Khazad-dûm and killed the Dwarven king Durin VI. Soon after, Goblins began to raid the abandoned city for its Dwarven treasures, and then chose to occupy it, worshiping the Balrog. Gandalf the Grey defeated the Balrog while the fellowship of the ring passed through the mines of Moria on their way to Mordor, but many Goblins remained.

In the fourth age, after the war of the ring, a group of Dwarves led by Balgar II wish to reclaim the city of Khazad-dûm. To do this, they will need to trap each Goblin in one of the many labyrinths in Moria. We will consider the case where there is one Goblin which needs to be trapped in a single labyrinth.

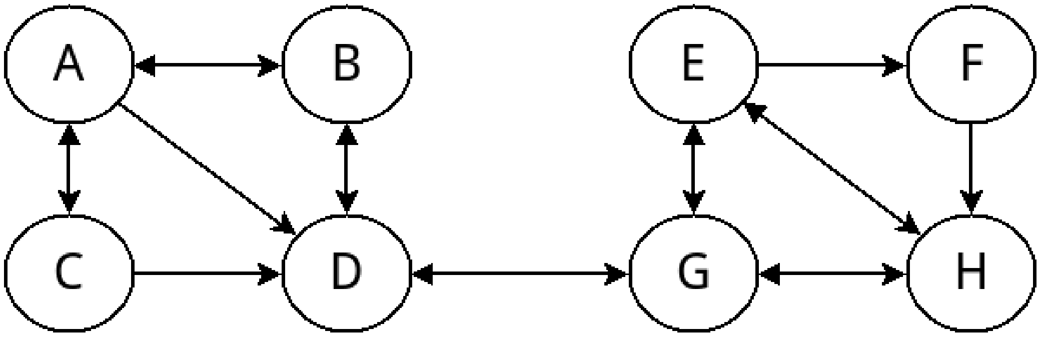
Each labyrinth is a series of caverns connected by reasonably straight passages, some of which can only be traversed in one direction. In order to trap a Goblin, Balgar has brought a large supply of torches into the mines, as the Goblins are deathly afraid of the light.

To trap a Goblin, Balgar wanders around somewhat aimlessly until he hears the Goblin approaching along a tunnel. At this point, he lights a torch and sets off in pursuit. The Goblin will retreat into the cavern it had just left and flee by another passage.

Balgar then follows, slowly gaining, until he reaches the k^{th} cavern since lighting the torch. Here he has enough time to place the lighted torch in the middle of the cavern, light another one from it, and continue the chase. As the chase progresses, a torch is left in each k^{th} cavern passed through, thereby limiting the movement of the Goblin.

Whenever the Goblin enters a cavern, it checks the exits in a particular order, fleeing down the first that does not lead directly to a lit cavern. Remember that, as Balgar is carrying a lit torch, the Goblin never flees a cavern by the tunnel used to enter it. Eventually the Goblin becomes trapped, enabling Balgar to dispatch it with his axe.

Consider the following Labyrinth as an example, where in this case the Goblin checks the exits from a cavern in alphabetical order:



Assume that Balgar is in cavern C when he hears the Goblin approaching from A, and that for this scenario, the value of *k* is 3. He lights a torch and gives chase, pursuing it through A, B, D (leaves a torch), G, E, F (another torch), H, E, G (another), H, E (trapped).

Write a program that will simulate Balgar's pursuit of the Goblin. The description of a labyrinth will identify each cavern by an upper case character and will list the caverns reachable from that cavern in the order that the Goblin will attempt them, followed by the identifiers for the caverns which the Goblin and Balgar were in when contact was first made, followed by the value of *k*.

Input

The first line of input will give the number of caverns in the labyrinth. After that, there will be one line of input for each cavern. These lines will contain a sequence of one or more uppercase letters. The first is the label of that cavern. The rest of the letters represent the caverns directly connected to that cavern, in the order that the Goblin will attempt to flee down them.

After the cavern lines, there will be one line of input giving the cavern that the Goblin begins in, one line of input giving the cavern that Balgar begins, and finally one line of input giving the value *k*.

The sample input below describes the labyrinth given in the diagram above.

Output

Output will consist of two lines. The first will give the list of caverns that were lit during the chase, in alphabetical order, separated by a space. The second line gives the cavern the Goblin was eventually trapped in.

Sample Input

```
8
ABCD
BAD
DBG
FH
GDEH
EFGH
HEG
CAD
A
C
3
```

Sample Output

```
D F G
E
```

Problem H: Redhorn Pass

> *"From signs that we have seen lately," said Gandalf, "I fear that the Redhorn Gate may be watched; and also I have doubts of the weather that is coming up behind. Snow may come. We must go with all the speed that we can. Even so it will take us more than two marches before we reach the top of the pass. Dark will come early this evening. We must leave as soon as you can get ready."*

The Redhorn Pass (also called the Redhorn Gate) was a treacherous pass through the Misty Mountains which run from North to South in central Middle Earth. The pass was used by Hobbits migrating from Gladden Fields into Eriador. The trip through Redhorn Pass was dark and treacherous, so the travelling Hobbits needed to bring a lantern across.

The pass was so narrow that at most two Hobbits could cross the pass at any one time, and no Hobbits could safely travel without the lantern. The Hobbits only have one lantern between them, so to get all N Hobbits safely across, some sort of shuttle arrangement must be made in order to return the lantern so that more Hobbits may cross.

Each Hobbit has a different crossing speed; the speed of a group is determined by the speed of the slower member. Your job is to determine a strategy that gets all N Hobbits across the pass in the minimum time.

Input

The input begins with the number of Hobbits in the group, N, followed by N lines giving the crossing times for each of the Hobbits. There are not more than 1,000 Hobbits and nobody takes more than 100 minutes to cross the pass.

Output

The first line of output should report the total number of minutes required for all N Hobbits to cross the pass. Subsequent lines give a strategy for achieving this time.

Each line contains either one or two integers, indicating which Hobbit or Hobbits form the next group to cross. Each Hobbit is indicated by the crossing time specified in the input. Although many Hobbits may have the same crossing time, this ambiguity is of no consequence.

Note that the crossings alternate directions, as it is necessary to return the lantern so that more may cross. If more than one strategy yields the minimal time, any one will do.

Sample Input

```
4
1
2
5
10
```

Sample Output

```
17
1 2
1
5 10
2
1 2
```

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);
   ```