Thomas Jefferson Invitational Open in Informatics 2012

Contest Part II (Practical—Programming)

Do not open until told to do so.

Rules:

1. You will have 150 minutes to complete this section.
2. For each problem, submit your solution to the Contest Portal at `http://activities.tjhsst.edu/tjioi/contest/`.
3. Up to five points will be awarded for each problem. Each of your programs will be run on five test cases, with each case worth one point. The first case is always equivalent to one of the example cases provided in the problem. When you submit a program, you will receive feedback for the sample case and one randomly chosen test case; no information will be given about the program's results on the other three. All problems are weighted equally, for a total of 55 possible points (but scaled down to 50).
4. Test cases for each problem are guaranteed to follow the input specifications. These cases are generally more difficult than the example cases and may be tricky (but legal), so be sure to consider all boundary cases that might break your program. Later test cases are harder than earlier ones.

5. Your programs should read from standard input and print to standard output. Do not print anything other than the answer to the problem, exactly as specified in the samples given for each problem.

6. All programs should run in less than 30 seconds and use less than 64 MB of memory.

7. The scoreboard only contains information about the two feedback cases. After the contest, all programs will be regraded with the other three cases.

8. For each test case, the following feedback messages are possible:

   - **OK** Your program correctly solved this test case! The number following the message is the number of seconds the program took on this case.
   - **Wrong: incorrect output** Your program printed an answer, but it did not match our answer.
   - **Wrong: restricted function** Your program used a function that it is not allowed to (probably a system call or an illegal file open). See us for details.
   - **Wrong: memory limit exceeded** Your program used more than the allotted memory limit of 64 MB.
   - **Wrong: time limit exceeded** Your program used more than the allotted time limit of 30 seconds.
   - **Wrong: runtime error** Your program crashed before printing an answer.
   - **Wrong: abnormal termination** Your program returned a nonzero exit code. C/C++ users should make sure there is a 'return 0;' at the end of the program. Python users should make sure that the right version of Python is selected.

9. Problems are arranged in approximate difficulty order.

10. Submit Java programs in a primary class and file given by the ID of the problem. The ID is in parentheses after the title of the problem. For example, if the ID is "add", then the program should be in a file "add.java" and be in a class called "add".

11. Good luck, and have fun!

Programming Problem 0

Problem Statement: The Quest Begins (miles)

King Alex, the beloved ruler of Caramelot, reigns over a good and ice cream-loving people. One day, he learns of a wondrous flavor, Milk and Cookies, made only in the upper reaches of the Carvel mountains. Craving some delicious ice cream goodness, King Alex asks the renowned knight Sir Albert to acquire a sample of this new flavor.

Sir Albert, an experienced adventurer, understands the importance of packing enough supplies before setting out on a quest. From his experience, our hungry knight knows that he will need to eat between 3 and 5 scoops of ice cream for each foot he travels.

Given a series of N ($1 \leq N \leq 100,000$) floating point numbers, h_i ($0 \leq h_i \leq 50,000$), with each value representing a possible total journey length in miles, output the minimum and maximum number of scoops of ice cream he will need for each h_i. Round your answers to the nearest integer.

Note: There are $5,280$ feet in one mile.

Input and Output Format

Input:

- Line 1: A single integer: N.
- Line 2...$N+1$: Line $i+1$ contains a single integer: h_i, the length in miles of journey i.

Output:

- Line 1...N: Line i contains two integers - the minimum and maximum number of scoops needed for journey i.

Examples

Input:

```
3
0.01
1.0
1000.0
```

Output:

```
158 264
15840 26400
15840000 26400000
```

Programming Problem 1

Problem Statement: A Number Game (numbers)

Sir Albert has finished preparations for his quest and is ready to leave. However, his mischievous arch-nemesis Sir Lbert demands that he take a break (already!) and play a game with him before leaving.

The game takes place in stages; in each stage, each player will write down one number. In the first stage, both players write down the number 1. For each of the remaining stages, one player will write down the last digit of the sum of the two numbers written in the previous stage, while the other will simply rewrite whatever digit he last wrote. As the stages progress, Albert and Lbert take turns adding or repeating. For example, let us say Lbert is the first to perform addition. The numbers written down would be:

Lbert: 1, 2, 2, 5, 5, 3, . . .

Albert: 1, 1, 3, 3, 8, 8, . . .

Being in a hurry to satisfy the King, Sir Albert wants to leave as soon as possible. Help him by calculating the value of the Nth number ($1 \leq N \leq 10,000$) written down by the player specified.

Input and Output Format

Input:

- Line 1: The integer N.
- Line 2: The player for whom the Nth number is desired (either "lbert" or "albert").
- Line 3: The player who started (either "lbert" or "albert").

Output:

- Line 1: A single integer that is the answer to this problem.

Examples

Input:

```
6
lbert
lbert
```

Output:

```
3
```

Explanation:

Lbert started the game and the sixth number he had was 3.

Programming Problem 2

Problem Statement: Ice Cream Race (overlap)

As Sir Albert rides off to the far reaches of Caramelot on his faithful steed, Sprinkles, he begins to daydream about some of his more memorable adventures.

In particular, he recalls the peculiar creatures of Coney Island - a strange and wondrous race that produced ice cream at alarming rates. Known only as the Glaciers, these creatures often partook in races to see who could produce more ice cream.

Sir Albert once had the fortune of witnessing a race between two Glaciers, Breezy and Freezy. Breezy turned out ice cream at a rate of A ($1 \leq A \leq 1,000$) scoops per second, while Freezy made them at a rate of B ($1 \leq B \leq 1,000$) scoops per second. To keep things manageable, Sir Albert checked each Glacier's stack at the end of every second. If he ever saw a stack of ice cream exceeding K ($max(A, B) \leq K \leq 100,000$) scoops, he immediately ate K scoops from that Glacier's stack.

If the competition went on for T ($1 \leq T \leq 100,000$) seconds, find the number of times both Glaciers had the same number of scoops in his stack, counting the initial configuration of 0 scoops each. Comparisons should be made at the end of each second, after Sir Albert finishes his eating.

Input and Output Format

Input:

- Line 1: Four integers: A, B, K, and T.

Output:

- Line 1: A single integer: the answer to this problem.

Examples

Input:

```
1 1 4 5
```

Output:

```
6
```

Explanation:

Initially, both Glaciers have 0 scoops. After one second, they each have 1 scoop. After two,

three, and four seconds, Breezy and Freezy both have 2, 3, and 4 scoops each respectively. After the fifth second, both Glaciers have a stack of height 5. Albert immediately eats 4 scoops from each stack, leaving 1 scoop each. The number of scoops match a total of 6 times.

Input:

```
2 3 7 10
```

Output:

```
2
```

Explanation:

From time 0 through 10, Breezy has 0, 2, 4, 6, 1, 3, 5, 7, 2, 4, and 6 scoops (in that order). Similarly, Freezy has 0, 3, 6, 2, 5, 1, 4, 7, 3, 6, and 2 scoops. Note that we only check the height after Albert finishes eating. The two stacks have the same height after 0 seconds and after 7 seconds.

Programming Problem 3

Problem Statement: Buying Ice Cream (buy)

Despite his carefully planned inventory, Sir Albert has consumed all of his ice cream already! He needs to purchase more if he wants to continue on his journey. Fortunately, he is stopped in the Frozen Forests, home of the Gelato Gnomes. He quickly heads over to the local creamery, which offers N ($1 \leq N \leq 10,000$) different flavors of ice cream. A scoop of ice cream of the ith flavor costs C_i ($1 \leq C_i \leq 10,000$) pieces of gold, and the ice cream store has Q_i ($1 \leq Q_i \leq 1,000$) scoops of type i ice cream in stock. Suppose that Sir Albert is willing to part with M ($1 \leq M \leq 1,000,000,000$) gold coins. What is the maximum number of ice cream scoops that he could purchase?

Input and Output Format

Input:

- Line 1: Two integers integer: N and M.
- Line 2...$N+1$: Line $i+1$ contains two integers: C_i and Q_i.

Output:

- Line 1: A single integer: the answer to this problem.

Examples

Input:

```
2 100
30 4
10 1
```

Output:

```
4
```

Explanation:

Sir Albert has a budget of 100 coins. The first type of ice cream costs 30 coins per scoop and can be bought at most 4 times. The second type of ice cream costs 10 coins per scoop and can be bought at most once. Sir Albert can achieve the maximum of 4 ice cream scoops by buying one scoop of the second type of ice cream and three scoops of the first type of ice cream.

Programming Problem 4

Problem Statement: Ice Cream Pizza (fractions)

Leaving Caramelot, Albert enters the sweltering deserts (not desserts!) to the South. The clever inhabitants of this land have gotten around the intransience of frozen ice cream by creating the ice cream pizza. In each of the N ($1 \leq N \leq 10$) houses he visits, Sir Albert is presented with a fraction of a full pizza denoted by the fraction A_i / B_i ($1 \leq A_i \leq B_i \leq 30$). Find the total amount of pizza he gets, expressed as a mixed fraction in lowest terms.

Input and Output Format

Input:

- Line 1: A single integer: contains N.
- Line 2...$N+1$: Line $i+1$ contains two integers: A_i and B_i.

Output:

- Line 1: The whole number portion of the number of pizzas Albert receives.
- Line 2: The fractional portion of the number of pizzas Albert receives. Express this as two space-separated integers, C and D, such that the fractional amount of pizza is equal to C/D and the fraction C/D is in simplest form. If the answer is a whole number, omit this line.

Examples

Input:

```
3
1 2
3 4
5 6
```

Output:

```
2
1 12
```

Explanation:

$1/2 + 3/4 + 5/6 = 2 + 1/12$.

Programming Problem 5

Problem Statement: Ice Cream Fad (fad)

Having crossed the desert, Sir Albert enters Scooptown. The town is currently in the grip of two rather unusual ice cream fads. Two new flavors, Java and Python, are quickly gaining popularity.

Scooptown is arranged in an N by M ($1 \leq N, M \leq 100$) grid. Each cell in the grid currently contains a regular person ('P'), an empty space ('E'), a Java ice cream lover ('X'), or a Python ice cream lover ('Y'). There may eventually exist people who like both flavors of ice cream, but no one likes both flavors initially.

Java-flavored ice cream fans spread the trend to people laterally (up, down, left, and right). Python-flavored ice cream lovers spread their preference to people diagonally (in the four diagonal directions). Neither fad can spread through empty spaces.

Every person who becomes influenced by either trend will spread it according to the rules above. It takes one day for a newly converted person to spread the fad to his neighbors.

After the fads have finished spreading, figure out the number of people who like both Java and Python ice cream. In addition, figure out the number of days that must elapse before both fads stop spreading.

Input and Output Format

Input:

- Line 1: Two integers: N and M.
- Line 2...$N+1$: Line $i+1$ contains M characters, each of which is one of 'P', 'E', 'X', and 'Y'.

Output:

- Line 1: A single integer that is the number of people who like both ice cream flavors.
- Line 2: A single integer that is the number of days it takes for the fad to finish spreading.

Examples

Input:

```
5 6
PPEEPP
PXEPEP
PEEEYE
PPEPEE
EPPEPP
```

Output:

```
5
6
```

Explanation:

There are five people who like both types of ice cream in the end. The following depicts the final arrangement, where 'Z' denotes someone who likes both types of ice cream.

```
ZXEEYP
XZEYEY
ZEEEYE
XZEYEE
EXZEYP
```

It takes 6 days for the fad to finish spreading, because the Java fad does not reach the top left corner until 6 days have elapsed.

Programming Problem 6

Problem Statement: Scrambled Flavors (scramble)

In the absence of Sir Albert, local pranksters have scrambled the names of the flavors in Caramelot's ice cream shop. For example, the flavor "Cookies and Cream" might be scrambled into "Cream and Cookies". In addition, the tricksters have removed some flavors of ice cream and added in new flavors to the store's list, making things even more confusing.

Conveniently, the shop's owner remembers all of the flavors that were originally offered. Given a list of N ($1 \leq N \leq 100$) names of legitimate ice cream flavors and M ($1 \leq M \leq 100$) new ice cream flavors (the potentially scrambled counterparts), decide which of the new flavors match the original flavors.

One ice cream flavor is a scrambled version of another flavor if it is possible to switch around the words in one flavor's name to make the other. A word is defined as a string of letters separated from other words by spaces. No individual letters may be switched ("Strawberry" cannot be scrambled into "Starwberry").

Input and Output Format

Input:

- Line 1: Two integers: N and M.
- Line 2...$N+1$: Line $i+1$ contains a string of characters, no more than 100 characters long, denoting the name of an original flavor of ice cream. The string will consist of uppercase letters, lowercase letters, and spaces.
- Line $N+2$...$N+M+1$: Line $N+j+1$ contains a string of characters, no more than 100 characters long, denoting the name of new flavor of ice cream. The string will consist of uppercase letters, lowercase letters, and spaces.

Output:

- Line 1...?: Output each valid ice cream flavor, in scrambled form. If there are multiple, output them in the order they appeared in the input.

Examples

Input:

```
3 6
Cookies and Cream
Strawberry
```

```
George Washington
Washington George
Starwberry
Strawberry
Cream and Cookies
Cookies and
Cream Cookies and
```

Output:

```
Washington George
Strawberry
Cream and Cookies
Cream Cookies and
```

Programming Problem 7

Problem Statement: Store Managers (boss)

Continuing on his journey, Sir Albert encounters a mythical creature previously known only in legend - the Mousse Mouse. In order to pass, Sir Albert must answer a few questions about the ice cream stores in Carmelot.

Carmelot's stores are organized in a very specific hierachy. There are N ($2 \leq N \leq 100$) store managers, conveniently numbered $1...N$. Every manager has a boss, except for manager 1, the most senior manager. A manager A is called superior to another manager B if A is the boss of B, or A is the boss of another manager C, who is the boss of B (i.e. superiority is transitive). A manager is called "lowly" if he is not the boss of anyone.

Each lowly manager owns an ice cream store containing five ice cream flavors. Each ice cream flavor has some quality, which is an integer between 1 and 10,000 inclusive (10,000 being the highest quality). Because Carmelot's ice cream flavors are quite diverse, no two lowly managers share any flavors in common (although the qualities of some of their ice creams might be the same).

Any manager X who is not a lowly manager also owns an ice cream store whose ice cream flavors are exactly the set of all ice cream flavors in the stores of those managers whom X is superior to.

For each manager, Albert must determine the quality of the 5th best ice cream flavor in that manager's shop.

Input and Output Format

Input:

- Line 1: A single integer: N.
- Line $2...N$: Line i contains P_i ($1 \leq P_i \leq N; P_i! = i$), the boss of manager i. If manager i is lowly (not the boss of anyone), its line will contain five more integers, each denoting the rating of a flavor of ice cream. No two managers will both be bosses of each other.

Output:

- Line $1...N$: Line i contains a single integer that is the rating of the fifth best flavor in the store owned by manager i.

Examples

Input:

```
5
1 1 4 6 7 8
1
3 4 5 7 1 3
3 4 1 2 100 2
```

Output:

```
6
1
4
1
1
```

Explanation:

Manager 2's store has flavors with qualities 1, 4, 6, 7, and 8. Manager 4's store has flavors with qualities 1, 3, 4, 5, and 7. Manager 5's store has flavors with qualities 1, 2, 2, 4, and 100. Thus, manager 3's store has flavors with qualities 1, 1, 2, 2, 3, 4, 4, 5, 7, and 100. Hence, manager 1's store has flavors with qualities 1, 1, 1, 2, 2, 3, 4, 4, 4, 5, 6, 7, 7, 8, and 100.

Programming Problem 8

Problem Statement: Ice Cream Grids (rotate)

In his latest adventure, Sir Albert has arrived at a mythical city full of ice cream stores. In fact, the city consists of a grid of R rows and C columns ($1 \leq R, C \leq 100$), and every cell in the grid contains an ice cream store. In this strange city, each ice cream store only sells ice cream of a single flavor, represented by a single lowercase letter between a and z.

Sir Albert, being very picky, would like to evaluate the ice cream tastiness potential of the city before exploring it further. Therefore, he wants to count the number of times his favorite mini-grid of ice cream flavors appears on the grid of ice cream stores. Sir Albert's mini-grid consists of N rows by N columns ($1 \leq N \leq R, C$), and each cell contains a single flavor. Help our trusty knight out by telling him the number of locations in the city grid in which his mini-grid appears.

There is, however, a slight catch. Albert does not mind rotating his grid clockwise 90 degrees (as many times as he wants) in order to let it fit in the grid. That is, Sir Albert can rotate his grid 0, 90, 180, or 270 degrees clockwise.

Input and Output Format

Input:

- Line 1: Two integers: R and C.
- Line 2...$R+1$: Line $i+1$ contains a string of C characters, each of which is a single lowercase letter denoting the flavor of the ice cream served by the store in that cell.
- Line $R+2$: A single integer: N.
- Line $R+3$...$R+N+2$: Line $j+R+2$ contains a string of N characters, each of which is a single lowercase letter denoting the flavor on Albert's mini-grid.

Output:

- Line 1: A single integer: the answer to this problem.

Examples

Input:

```
3 3
aaa
aaa
aab
```

```
2
aa
aa
```

Output:

```
3
```

Explanation:

The mini-grid [[a, a], [a, a]] appears 3 times with upper left corners at (0, 0), (0, 1), and (1, 0). Coordinates are in the format (row, column) and the top left corner is denoted by (0, 0).

Input:

```
4 4
cdef
cdef
dcdd
fbcc
2
cc
dd
```

Output:

```
2
```

Explanation:

The mini-grid [[c, c], [d, d]] appears 2 times with upper left corners at (0, 0) and (2, 2). Both are rotated versions.

Programming Problem 9

Problem Statement: Toppings (toppings)

Sir Albert is nearing the Carvel mountains. However, he realizes that the King will probably want some toppings on his Milk and Cookies ice cream. Sir Albert decides to collect the tastiest possible arrangement of toppings he can manage. No topping may be used more than once.

Sir Albert knows that the sample he collects can fit toppings taking up at most S ($1 \leq S \leq 1,000$) units of space. There are N ($1 \leq N \leq 100$) toppings he can choose from. The ith topping takes up C_i ($1 \leq C_i \leq 1,000$) units of space and has tastiness T_i ($1 \leq T_i \leq 1,000$). Given the space constraints, what is the maximum total tastiness of the toppings that Sir Albert can use?

Input and Output Format

Input:

- Line 1: Two integers, S and N.
- Line 2...$N+1$: Line $i+1$ contains two integers: C_i and T_i.

Output:

- Line 1: A single integer that is the maximum total tastiness possible.

Examples

Input:

```
10 4
4 10
3 15
8 22
2 2
```

Output:

```
27
```

Explanation:

If Sir Albert chooses the first, second, and fourth toppings, he uses up $4+3+2=9$ units of space (less than the capacity of 10) and achieves a total tastiness of $10+15+2=27$.

Programming Problem 10

Problem Statement: Too Many Flavors (manyflav)

Finally cresting Mount Gelato, Albert is ready to scoop up a sample of Milk and Cookies for King Alex. However, he is pleasantly surprised to see that the Carvel Mountains host a vast collection of ice cream flavors, arranged in bin after bin.

Specifically, the bins are arranged so that the first f_1 bins contain flavor 1, the next f_2 contain flavor 2, etc. In general, there are N ($1 \leq N \leq 100,000$) flavors of ice cream, conveniently labled $1...N$, and f_i ($1 \leq f_i \leq 1,000,000$) bins of flavor i. Before he starts sampling, Sir Albert has Q ($1 \leq Q \leq 100,000$) queries, b_j, about the bins of ice cream. For each query, print the flavor of the ice cream in the b_jth bin. Every query will be valid (within the range of the available buckets).

Input and Output Format

Input:

- Line 1: Two integers: N and Q.
- Line $2...N+1$: Line $i+1$ contains an integer, f_i, the number of bins of flavor i.
- Line $N+2...N+Q+1$: Line $N+j+1$ contains an integer, b_j, the bin Sir Albert wants to know the flavor of.

Output:

- Line $1...Q$: Line j contains a single integer that is the flavor of the b_jth bin.

Examples

Input:

```
3 2
5
2
1
3
8
```

Output:

```
1
3
```

Explanation:

The bins have the following flavors:

1 1 1 1 1 2 2 3

The 3rd and 8th bins have flavors 1 and 3, respectively.