

# The ACM International Collegiate Programming Contest Sponsored by IBM

## The Third Lebanese Collegiate Programming Contest

*Sponsored by*  
American University of Science and Technology  
**Bank of Beirut**

Berytech  
Netiks  
BMB  
CCT

American University of Science and Technology  
Beirut, Lebanon  
July 2011

Program:	fold.(c cpp java)
Input:	fold.in
Balloon Color:	blue

## [A] Sequence Folding

Alice and Bob are practicing hard for the new ICPC season. They hold many private contests where only the two of them compete against each other. They almost have identical knowledge and skills, the matter which results many times in ties in both the number of problems solved and in time penalty! To break the tie, Alice and Bob invented a tie breaker technique called sequence folding! The following are the steps of this technique:

- 1- Generate a random integer  $N \geq 2$ .
- 2- Generate a sequence of  $N$  random integers.
- 3- If  $N = 2$  go to step 6.
- 4- Fold the sequence by adding the  $N_{th}$  element to the first, the  $N-1_{th}$  element to the second and so on, if  $N$  is odd then the middle element is added to itself, figure 1 illustrates the folding process.
- 5- Set  $N = \text{ceil}(N/2)$  and go to step 3.
- 6- The sequence now contains two numbers, if the first is greater than the second then Alice wins, otherwise Bob wins.

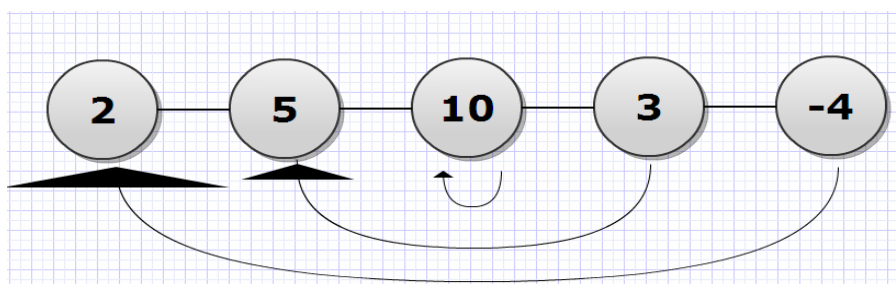


Figure 1.a Before Folding

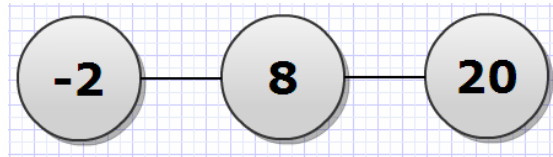


Figure 1.b After one step of folding

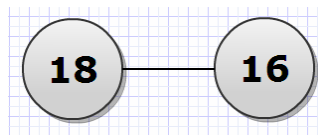


Figure 1.c After two steps of folding, Alice wins!

In this problem you're given the sequence of  $N$  integers and are asked determine the contest winner using the sequence folding tie breaker technique.

### Input Specification

The first line contains  $T$  ( $1 \leq T \leq 100$ ), the number of test cases. The first line of each test case contains an integer ( $2 \leq N \leq 100$ ), the number of elements of the sequence. The next line contains  $N$  space separated integers. The sum of any subset of the numbers fit in a 32 bit signed integer.

### Output Specification

For each test case print the name of the winner. Follow the output format below.

### Sample Input

```
2
5
2 5 10 3 -4
3
5 4 -3
```

### Sample Output

```
Case #1: Alice
Case #2: Bob
```

Program:	coprime.(c cpp java)
Input:	coprime.in
Balloon Color:	gold

## [B] Co-prime

Given a number  $N$ , you are asked to count the number of integers between  $A$  and  $B$  inclusive which are relatively prime to  $N$ .

Two integers are said to be co-prime or relatively prime if they have no common positive divisors other than 1 or, equivalently, if their greatest common divisor is 1. The number 1 is relatively prime to every integer.

### Input Specification

The first line on input contains  $T$  ( $0 < T \leq 100$ ) the number of test cases, each of the next  $T$  lines contains three integers  $A$ ,  $B$ ,  $N$  where ( $1 \leq A \leq B \leq 10^{15}$ ) and ( $1 \leq N \leq 10^9$ ).

### Output Specification

For each test case, print the number of integers between  $A$  and  $B$  inclusive which are relatively prime to  $N$ . Follow the output format below.

### Sample Input

```
2
1 10 2
3 15 5
```

### Sample Output

```
Case #1: 5
Case #2: 10
```

In the first test case, the five integers in range  $[1,10]$  which are relatively prime to 2 are  $\{1,3,5,7,9\}$ .

Program:	wimbledon.(c cpp java)
Input:	wimbledon.in
Balloon Color:	yellow

## [C] Wimbledon

I'm flying back to Cairo tomorrow, and do you know what is sad about that? Well, in addition to leaving beautiful Lebanon? Yep, I'm going to miss the 2011 Wimbledon final match which is going to happen while I'm 20,000 feet above sea level!

But hey, wait! I may have a shot here! At Wimbledon, matches are played only at day light, so maybe if players were not ready to give up easily on each point, then the match will extend until sunset, and the match would be suspended and resumed on Monday just for my convenience! So what I need to know is how long on average a game between two players will last, and to help me feel better you should write a computer program to determine that!

The result of a tennis match is determined by the number of "sets" each player wins, the first player to win three sets wins the match. Accordingly, all possible match results are 3-0, 3-1 and 3-2. The result of each set is determined by the number of "games" in the set each player wins, the first player to win six or more games with two or more game difference from the opponent wins the set, however if the result of a set (including the last set) leveled at 6-6 then a tie-break game is played to determine the set winner, Accordingly all possible set results are 6-0, 6-1, 6-2, 6-3, 6-4, 7-5 and 7-6 using a tie breaker game!

During each game (including tie-break), one player has the serve, this means that this player must start the play for all points of the game by hitting the ball, serving the ball is considered a big advantage. Assume that the serve starts at the first player and then alternates after each game till the end.

Given the probability of winning a game on serve for each player against his opponent, and assuming that each game lasts for five minutes, calculate the expected duration of the match.

## Input Specification

The first line of input contains an integer  $T$ , the number of test cases. The first line of each test case contains the first and last names of the first player, followed by an integer ( $0 \leq A \leq 100$ ) where  $A/100$  is the probability that the first player wins a game on his serve against the second player. The second line contains the first and last names of the second player, followed by an integer ( $0 \leq B \leq 100$ ) where  $B/100$  is the probability that the second player wins a game on his serve against the first player.

## Output Specification

For each test case, print the case number followed by the expected duration of the match in minutes rounded to six decimal digits.

## Sample Input

```
2
Rafael Nadal 50
Roger Federer 50
Pete Sampras 100
Hamza Darwish 0
```

## Sample Output

```
Case #1: 199.281006
Case #2: 90.000000
```

In the second test case, I don't stand a chance against Pete! He always wins all games on his serve (probability 100/100) and he also always wins all games on my serve (as I win with probability 0/100), so he always wins the match with three straight sets (6-0, 6-0, 6-0), a total of 18 games played, each lasting five minutes for a total of 90 minutes match.

Program:	sort.(c cpp java)
Input:	sort.in
Balloon Color:	pink

## [D] Manhattan Sort

Yet another sorting problem! In this one, you're given a sequence  $S$  of  $N$  distinct integers and are asked to sort it with minimum cost using only one operation:

*The Manhattan swap!*

Let  $S_i$  and  $S_j$  be two elements of the sequence at positions  $i$  and  $j$  respectively, applying the Manhattan swap operation to  $S_i$  and  $S_j$  swaps both elements with a cost of  $|i-j|$ . For example, given the sequence  $\{9,5,3\}$ , we can sort the sequence with a single Manhattan swap operation by swapping the first and last elements for a total cost of 2 (absolute difference between positions of 9 and 3).

### Input Specification

The first line of input contains an integer  $T$ , the number of test cases. Each test case consists of 2 lines. The first line consists of a single integer ( $1 \leq N \leq 30$ ), the length of the sequence  $S$ . The second line contains  $N$  space separated integers representing the elements of  $S$ . All sequence elements are distinct and fit in 32 bit signed integer.

### Output Specification

For each test case, output one line containing a single integer, the minimum cost of sorting the sequence using only the Manhattan swap operation.

### Sample Input

```
2
3
9 5 3
6
6 5 4 3 2 1
```

### Sample Output

```
Case #1: 2
Case #2: 9
```

## [E] Cipher

Program:	cipher.(c cpp java)
Input:	cipher.in
Balloon Color:	black

A spy is using a clever cipher technique to encode and decode messages. The technique is very efficient as it enables him to send and receive any message, no matter how long it is with only three numbers!

After many great efforts, The Department of National Security managed to figure out how the technique works:

1. First of all, the only characters used are the space and lower case English letters. Each character is assigned an integer called the character volume, a space character has volume 1, 'a' has volume 2, 'b' has volume 3 and so on until 'z' which has volume 27. The volume of the whole message  $V$  is the sum of volumes of characters in the message.
2. A message consists of a number of words  $W$ , a word is a consecutive sequence of lower case letters, messages have no leading or trailing spaces and there is only one space between consecutive words.
3. For a certain  $V$  and  $W$ , let  $S$  be the lexicographically sorted set of messages that have volume  $V$  and consists of exactly  $W$  words. We can refer to a certain message using its one-based index  $I$  in  $S$ .

So when the spy wants to send a message  $M$ , he calculates its Volume  $V$  and the number of words it contains  $W$  and finds its index  $I$  in the corresponding set  $S$  (its index among all sorted messages of volume  $V$  with  $W$  words) and sends only three numbers along:  $V$ ,  $W$  and  $I$ !

The Department of National Security has done a great effort so far, yet they seek your help to decode messages sent by the spy! That is, given  $V$ ,  $W$  and  $I$  you must decrypt the spy's message or determine that there is no such one!



## Input Specification

The first line of input contains an integer ( $1 \leq T \leq 200$ ), the number of test cases. Each data set consists of a line with three integers the volume of the message ( $1 \leq V \leq 75$ ), the number of words ( $1 \leq W \leq 20$ ), and the index of message ( $1 \leq I \leq 10^{18}$ ).

## Output Specification

For each test case, print one line containing the decrypted message or "Corrupted!" if there is no valid message corresponding to the given inputs.

## Sample Input

```
3
7 2 3
2 1 2
50 2 39098532022
```

## Sample Output

```
Case #1: aa a
Case #2: Corrupted!
Case #3: big bang
```

In the first test case, the sorted set of messages of volume 7 which contain 2 words is {"a aa", "a c", "aa a", "b b", "c a"}, hence the third and required message is "aa a".

In the second test case, the sorted set of messages of volume 2 which contain 1 word is {"a"}, the set has only one message while index 2 is required, so something must have gone wrong!

Program:	division.(c cpp java)
Input:	division.in
Balloon Color:	purple

## [F] Big Division

A theoretical physicist friend is doing research about the "Answer to the Ultimate Question of Life, the Universe, and Everything", he thinks that it is not 42 as suggested by "The Hitchhiker's Guide to the Galaxy" science fiction comedy series; instead he thinks it is the result of dividing the products of two sequences of positive integers A and B!

The task of calculating the product of A and B followed by division turned out to be not as easy as it looks, specially with the sequences being long and the products getting too large very quickly! Even using a modern computer, a straight forward implementation for the calculations may take very long time to complete! And this is where we seek your help as a brilliant computer scientist!

### Input Specification

The first line of input contains an integer ( $1 \leq T \leq 200$ ), the number of test cases. T test cases follow, the first line of each test case contains two integers ( $1 \leq N, M \leq 110,000$ ), the lengths of sequences A and B respectively. Two lines follow, the first line contains N space separated integers ( $0 < A_0, A_1 \dots A_n \leq 1,000,000$ ), and the second line contains M space separated integers ( $0 < B_0, B_1 \dots B_m \leq 1,000,000$ ).

### Output Specification

For each test case, print one line containing the result of dividing the product of sequence A by the product of sequence B as a reduced fraction of the format "X / Y" (Notice the single space before and after the fraction sign). X and Y are guaranteed to fit in 32-bit signed integer. A reduced fraction is a fraction such that the greatest common divisor between the nominator and the denominator is 1.

### Sample Input

```
2
3 1
2 4 5
12
2 4
1 15
5 1 7 2
```

### Sample Output

```
Case #1: 10 / 3
Case #2: 3 / 14
```

Program:	calc.(c cpp java)
Input:	calc.in
Balloon Color:	orange

## [G] Calculator

The users feedback for your most beloved open-source operating system is gathered, and guess what the most required feature turned out to be? Yes it is finally what we all have been waiting for so long, extending the built-in calculator functionality! One of the suggested extensions is adding the capability of evaluating polynomials, and that's what you are thrilled to participate with!

In this problem you're given a polynomial entered by the user in the calculator and are asked to evaluate it for a certain value.

### Input Specification

The first line of input contains  $T$  ( $0 < T \leq 100$ ) the number of polynomials, each test case consists of two lines, the first line contains an integer ( $-1000 \leq X \leq 1000$ ), the value for the variable  $X$  for which the polynomial is evaluated.

The second line contains a polynomial  $P$  with integer coefficients.  $P$  is a sum of terms of the form  $CX^E$ , where the coefficient  $C$  and the exponent  $E$  satisfy the following conditions:

1.  $E$  is an integer satisfying ( $0 \leq E \leq 30$ ). If  $E$  is 0, then  $CX^E$  is expressed as  $C$ . If  $E$  is 1, then  $CX^E$  is expressed as  $CX$ , unless  $C$  is 1 or -1. In those instances,  $CX^E$  is expressed as  $X$  or  $-X$ .
2.  $C$  is an integer. If  $C$  is 1 or -1 and  $E$  is not 0 or 1, then the  $CX^E$  will appear as  $X^E$  or  $-X^E$ .
3. Only non-negative  $C$  values that are not part of the first term in the polynomial are preceded by  $+$ .
4. Exponents in consecutive terms are strictly decreasing.
5.  $C$  fits in a 32-bit signed integer.

## Output Specification

For each test case, print the value of polynomial evaluation. The result will fit in a 32-bit signed integer. Follow the output format below.

## Sample Input

```
2
-2
2X^2-5X+7
2
-3X^12+X
```

## Sample Output

```
Case #1: 25
Case #2: -12286
```

## [H] Crank

Program:	crank.(c cpp java)
Input:	crank.in
Balloon Color:	green

Chev Chelios had his heart stolen from him, by the boss of the most dangerous gang in the city. His heart has been replaced with an artificial chargeable one!

Chelios was on mission of locating the boss of the gang to get his heart back since then without luck! Now the artificial heart's battery lifetime is about to expire! Fortunately, he finally located the target but he needs your help to get there before his artificial heart stops beating!

Chelios decided to attack the boss building from the roof because all gates are heavily protected by gangsters. Chelios has the map of the gang block which shows the heights of all buildings within the block. The plan is that a helicopter will drop Chelios on the roof of one of the buildings on the boundary of the block during night. Then Chelios will get to the boss building by moving to adjacent buildings, vertically or horizontally. Chelios can only move to a building which has the same or smaller height as the current building (going up severely affects his damaged heart).

Given the gang building block map which shows the heights of all buildings in the block along with the boss building, write a program to help Chev Chelios determine the number of buildings on the boundary of the block he can be dropped by the helicopter at so that he would be able to reach the boss's building without climbing!

## Input Specification

The first line of input contains an integer  $T$ , the number of test cases.  $T$  test cases follow, the first line of each test case contains two integers ( $1 \leq R, C \leq 10$ ) the height and width of the building block. The second line contains two integers ( $1 \leq A \leq R$ ), and ( $1 \leq B \leq C$ ), the coordinates of the boss building on the map.  $R$  lines follows; each line consists of  $C$  space separated integers representing the heights of all buildings. A height  $H$  of a building satisfies ( $1 \leq H \leq 1000$ ).

## Output Specification

For each test case, print the number of buildings on the boundary of the block Chev Chelios can be dropped by the helicopter at so that he would be able to reach the boss's building without climbing up! Follow the output format below.

## Sample Input

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

## Sample Output

```
Case #1: 1
Case #2: 4
```

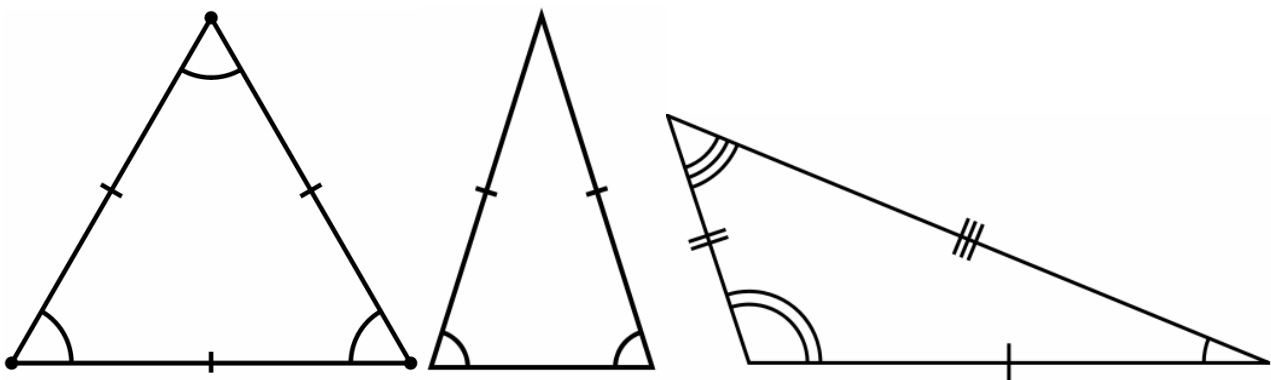
Program:	triangle.(c cpp java)
Input:	triangle.in
Balloon Color:	red

## [I] Triangle

A triangle is one of the basic shapes of geometry: a polygon with three corners or vertices and three sides or edges which are line segments.

Triangles can be classified according to the relative lengths of their sides:

- In an equilateral triangle all sides have the same length. An equilateral triangle is also a regular polygon with all angles measuring  $60^\circ$ .
- In an isosceles triangle, two sides are equal in length. An isosceles triangle also has two angles of the same measure; namely, the angles opposite to the two sides of the same length; this fact is the content of the Isosceles triangle theorem.
- In a scalene triangle, all sides are unequal. The three angles are also all different in measure. Some (but not all) scalene triangles are also right triangles.





### Input Specification

The first line of input contains an integer ( $1 \leq T \leq 100$ ), the number of test cases.  $T$  test data sets follow, each data set consists of 3 integers  $A$ ,  $B$  and  $C$ , where ( $1 \leq A, B, C \leq 1,000,000$ ) the triangle side lengths.

### Output Specification

For each test case, print "equilateral", "isosceles" or "scalene" describing the triangle type. If the input doesn't create a valid triangle output "invalid!". Follow the output format below.

### Sample Input

```
2
3 3 4
6 4 2
```

### Sample Output

```
Case #1: isosceles
Case #2: invalid!
```