EIGHTH ANNUAL JUILFS CONTEST
SPRING 2016

Department of Computer and Information Science
University of Oregon
2016 May 21

contributors: Chris Brooks, Danny Khan, Robert Sappington, Chris Wilson

A. Close my Parentheses

(2 points)

Problem

John is the Chief Editor for the Daily Duck Newspaper Company. It is Johns job to make sure that everything published to the paper has no grammar or syntax errors. John, having a Computer Science background, wishes to automate errors he sees frequently so he can get back to studying more Computer Science.

Your job is to help John out by writing a program to close a string of characters with any unclosed parentheses. For example, if the string is "I really like candy (except only on weekends", your program will output a single parenthesis, ")". Can you help John out so that he has more time to study Computer Science?

Input

The first line of the input will be the number of test cases, **T**. **T** test cases follow. Each following test case will be a string **S** that will need to be fixed.

Output

For each test case, output one line containing "Case #x: " where x is the case number (starting from 1), followed by a string of closed parentheses, ")". If there are no parentheses to display, print "NONE" instead.

Limits

1 ≤ **T** ≤ 50.
1 ≤ **S.length** ≤ 100.

Sample Input:	Sample Output:
4	Case #1:)
(Hello World	Case #2:))
(Lets try and help (John	Case #3:)))
(SF(EW( ) (	Case #4:)
)#$^#FD(ssg(werwg)	

Note: for Case #4, there is only one ")" because the only parenthesis you have to close is after "FD". You do not count the leading closing parenthesis because it is not closing anything.

B. TOO MANY THINGS!
(2 points)

Once you start to collect things, your house will fill up with them very quickly. They are able to replicate themselves in a very strange way. Each day, there is a new generation of things. Thing1 want to count the items when their sum are only even amounts. Thing1 also wants to know how many items it counted total. This is even worse than the famous Fibonacci growth rate!

Their total numbers can be expressed as a sequence

0, 1, 1, 2, 3, 5, 8, 13, 21, 34….

$F_1 = 0$
$F_2 = 1$
$F_n = F_{n-1} + F_{n-2}$

Your job is to write a program that will calculate a series of N values. The input will be an integer n ($0<n<20$), being the number of test cases. Following this will be n integer values, each on a separate line. Each integer will represent a number of k ($0\leq k<34$). For each number k, the program should write down the corresponding F[k], add the sum of all the days that have even values. For example k = 4, $F_1 = 0$, $F_2 = 1$, $F_3 = 1$ $F_4 = 2$, only two are even numbers, $F_1 + F_4 = 0 + 2 = 2$, so output is 2.

Sample Input	Sample Output
5 1 4 7 9 10	0 2 10 10 44

C. Search the Building (3 points)

Liam Neeson's daughter has once again been kidnapped. From the information he has gathered, Neeson is able to conclude that his daughter is being held in one of the buildings in a location nearby. Since he knows that the building she is being held in will likely have many guards, he uses a remote controlled drone equipped with an infra-red camera to record the number of people in each building. In order to make sense of this data, Neeson employs the help of the students of the University of Oregon to create a program that can help him find buildings with suspiciously large numbers of people for him to investigate.

The program will need to be able to read from standard input. The first line will be two numbers indicating the number of rows and columns in the grid of buildings and all subsequent rows will consist of a number indicating the number of people in the building and a letter indicating whether it is a residential or office building. The rows and columns are numbered 0, 1, 2, …, N-1 and 0, 1, 2, …, M-1 respectively. There is a building located at every point in the grid. The list of buildings is row by row, that is the first M lines are row 0, the next M lines are row 1, etc. The program will need to return the coordinates of all buildings with numbers of people above the average for their building type followed by the number of people in the building.

Sample input:

```
4 3
2 R
5 R
86 O
52 O
3 R
45 O
15 R
41 O
1 R
38 O
3 R
4 R
```

Sample Output:

```
Residential buildings to investigate:
0 1 5
2 0 15
Office buildings to investigate:
0 2 86
```

D. Knox on fox in socks in box

(4 points)

Knox has a box and Fox has a box. They each have a box for their socks. But …

The boxes have to be in a different place since they cannot share the space, else their socks get all mixed up.

So, they need a program which will take a pair of rectangles on the plane, and say if they share any space ("overlap"). A rectangle is described by a pair of points (x1, y1) and (x2, y2). These are the upper left and lower right corners of the rectangle. Two rectangles will be described by 4 such pairs, and the program needs to determine if this pair overlap.

The input will be a number n, $1 \leq n \leq 20$, followed by n pairs of rectangles. Each rectangle will be on one line, so there will be 2n lines following the first line. A line for a rectangle will be 4 numbers

```
x1 y1 x2 y2
```

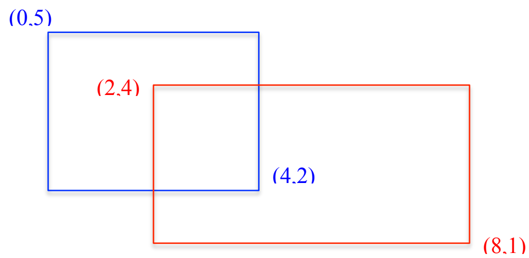
describing the two points.

For example, the input

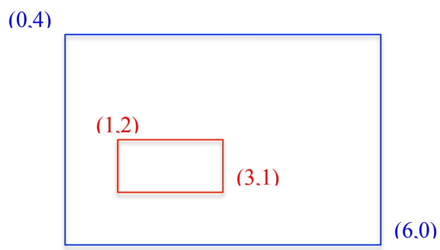
```
3
0 5 4 2
2 4 8 1
0 4 6 0
1 2 3 1
3 4 7 2
1 2 6 0
```

describes the three pairs of rectangles pictured below. The desired output is

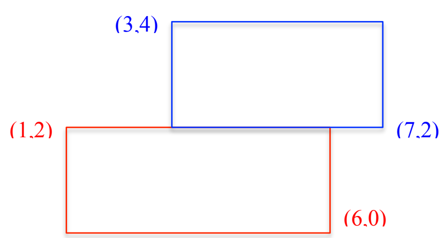
```
OVERLAP
OVERLAP
NO OVERLAP
```



(0,5)
(2,4)
(4,2)
(8,1)



(0,4)
(1,2)
(3,1)
(6,0)



(3,4)
(1,2)
(7,2)
(6,0)

E. Duck Cafe

(5 points)

Problem

Chris is a starving college student living on a tight budget. He decides to go to the Duck Cafe for lunch. The Duck Cafe is a little different than other Cafes; All of their items are the same, but have different costs depending on the quality of the item. Note that the Duck Cafe will not sell two or more items of the same quality. Chris wants the best meal he can get on his budget that will satisfy his hunger.

With the *exact* amount of Chris's **B** budget, help him find the number of combinations from a list of **N** Duck Cafe items that total the number of **I** items Chris can eat. Chris does not want to waste money buying more food than he can eat, but he also does not want to go hungry by not buying enough food.

Input

The first line of the input will be the number of test cases, **T**. **T** test cases follow.

The first line of each test case contains three integers: the number **B** of Chris's budget in dollars, the number of **I** items Chis can eat, and the number of **N** items the Cafe sells.

The second line of each test case contains **N** space-separated integers of the price in dollars.

Output

For each test case, output one line containing "Case #x: " where x is the case number (starting from 1), followed by a single integer of the number of combinations Chris can buy of **I** items with **B** budget.

Limits

$1 \leq \mathbf{T} \leq 100.$
$1 \leq \mathbf{B} \leq 100.$
$1 \leq \mathbf{I} \leq 5.$
$1 \leq \mathbf{N} \leq 50.$

Sample Input:

```
3
50 2 4
10 40 20 30
30 1 3
10 30 15
100 4 5
50 75 25 20 5
```

Sample Output:

```
Case #1: 2
Case #2: 1
Case #3: 1
```