acm International Collegiate Programming Contest 2009 IBM event sponsor

Maratona de Programação 2009

Seletiva CIn-UFPE

23 de Maio de 2009

(Este caderno possui 6 problemas; as páginas estão numeradas de 1 a 7, não contando esta página de rosto)

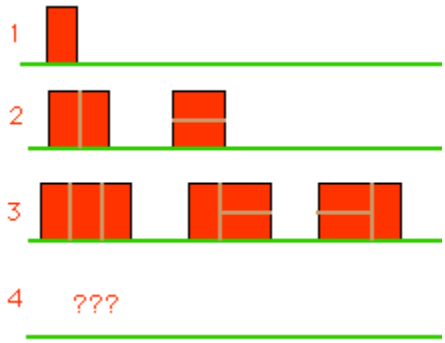
Observações Importantes:

1) O problema deve ser enviado com leitura/escrita na entrada/saída padrão.
2) O sistema de submissão de problemas se encontra em **www.cin.ufpe.br/~time-ufpe/contest** .
   Para submeter:
   2.1) Logue com o seu id cadastrado, entre em **submit**
   2.2) No campo problema coloque a letra referente ao problema a ser submetido
   2.3) Escolha a linguagem
   2.4) Submeta o problema.
   2.5) Aguarde o resultado da sua submissão, na página usada para submeter o problema.
3) O resultado de todos pode ser acompanhado pelo Scoreboard, encontrado na página.
4) Durante o 60 minutos finais o placar não é mais atualizado e durante os 15 minutos finais os participantes não recebem mais o resultado das suas submissões.
5) Não é permitido o acesso à nenhuma página durante até o final da competição.

DIVIRTAM-SE !

A - Brick Wall Patterns

In how many ways can you tile a 2x*n* wall by 2x1 tiles? Here is a sample tiling of a 2x1 , 2x2 and 2x3 walls.



How many ways can you find for a wall of length 4? And, for a wall of length 5?

Input

Your program receives a sequence of positive integers, one per line, each representing the length of a wall. The maximum value for the wall is length 50. The input terminates with a 0.

Output

For each wall length given in the input, your program must output the corresponding number of different ways for such a wall in a separate line.

Sample Input

```
1
2
3
0
```

Sample Output

```
1
2
3
```

B - Crazy tea party

N participants of *crazy tea party* sit around the table. Each minute one pair of neighbors can change their places. Find the minimum time (in minutes) required for all participants to sit in reverse order (so that left neighbors would become right, and right - left).

Input

The first line is the amount of tests. Each next line contains one integer n (1 <= n <= 32767) - the amount of crazy tea participants.

Output

For each number n of participants to crazy tea party print on the standard output, on a separate line, the minimum time required for all participants to sit in reverse order.

Sample Input

```
3
4
5
6
```

Sample Output

```
2
4
6
```

C - Subsequence

A sequence of *N* positive integers (10 < *N* < 100 000), each of them less than or equal 10000, and a positive integer *S* (*S* < 100 000 000) are given. Write a program to find the minimal length of the subsequence of consecutive elements of the sequence, the sum of which is greater than or equal to *S*.

Input

Many test cases will be given. For each test case the program has to read the numbers *N* and *S*, separated by an interval, from the first line. The numbers of the sequence are given in the second line of the test case, separated by intervals. The input will finish with the end of file.

Output

For each the case the program has to print the result on separate line of the output file. **If there isn't such a subsequence, print 0 on a line by itself.**

Sample Input

```
10 15
5 1 3 5 10 7 4 9 2 8
5 11
1 2 3 4 5
```

Sample Output

```
2
3
```

D - Find the Telephone

In some places is common to remember a phone number associating its digits to letters. In this way the expression **MY LOVE** means **69 5683**. Of course there are some problems, because some phone numbers can not form a word or a phrase and the digits **1** and **0** are not associated to any letter.
Your task is to read an expression and find the corresponding phone number based on the table below. An expression is composed by the capital letters (**A-Z**), hyphens (**-**) and the numbers **1** and **0**.

Letters	Number
ABC	2
DEF	3
GHI	4
JKL	5
MNO	6
PQRS	7
TUV	8
WXYZ	9

Input

The input consists of a set of expressions. Each expression is in a line by itself and has **C** characters, where **1 ≤ C ≤ 30**. The input is terminated by enf of file (EOF).

Output

For each expression you should print the corresponding phone number.

Sample Input

```
1-HOME-SWEET-HOME
MY-MISERABLE-JOB
```

Sample Output

```
1-4663-79338-4663
69-647372253-562
```

E - Shopaholic

Lindsay is a shopaholic. Whenever there is a discount of the kind where you can buy three items and only pay for two, she goes completely mad and feels a need to buy all items in the store. You have given up on curing her for this disease, but try to limit its effect on her wallet.

You have realized that the stores coming with these offers are quite selective when it comes to which items you get for free; it is always the cheapest ones. As an example, when your friend comes to the counter with seven items, costing 400, 350, 300, 250, 200, 150, and 100 dollars, she will have to pay 1500 dollars. In this case she got a discount of 250 dollars. You realize that if she goes to the counter three times, she might get a bigger discount. E.g. if she goes with the items that costs 400, 300 and 250, she will get a discount of 250 the first round. The next round she brings the item that costs 150 giving no extra discount, but the third round she takes the last items that costs 350, 200 and 100 giving a discount of an additional 100 dollars, adding up to a total discount of 350.
Your job is to find the maximum discount Lindsay can get.

Input

The first line of input gives the number of test scenarios, $1 \leq t \leq 20$. Each scenario consists of two lines of input. The first gives the number of items Lindsay is buying, $1 \leq n \leq 20000$. The next line gives the prices of these items, $1 \leq p_i \leq 20000$.

Output

For each scenario, output one line giving the maximum discount Lindsay can get by selectively choosing which items she brings to the counter at the same time.

Sample Input

```
1
6
400 100 200 350 300 250
```

Sample Output

```
400
```

F - Shoemaker's Problem

Shoemaker has N jobs (orders from customers) which he must make. Shoemaker can work on only one job in each day. For each i_{th} job, it is known the integer T_i ($1<=T_i<=1000$), the time in days it takes the shoemaker to finish the job. For each day of delay before starting to work for the i_{th} job, shoemaker must pay a fine of S_i ($1<=S_i<=10000$) cents. Your task is to help the shoemaker, writing a programm to find the sequence of jobs with minimal total fine.

Input

The input begins with a single positive integer on a line by itself indicating the number of the cases following, each of them as described below. This line is followed by a blank line, and there is also a blank line between two consecutive inputs.

First line of input contains an integer N ($1<=N<=1000$). The next N lines each contain two numbers: the time and fine of each task in order.

Output

For each test case, the output must follow the description below. The outputs of two consecutive cases will be separated by a blank line.

You programm should print the sequence of jobs with minimal fine. Each job should be represented by its number in input. All integers should be placed on only one output line and separated by one space. **If multiple solutions are possible, print the first lexicographically.**

Sample Input

```
1

4
3 4
1 1000
2 2
5 5
```

Sample Output

```
2 1 3 4
```