Code Wars 2006 :: Programming

Passwords

Some computer programs try to prevent easily guessable passwords being used, by rejecting those that do not meet some simple rules. One simple rule is to reject passwords which contain any sequence of letters immediately followed by the same sequence. For example:

APPLE	Rejected	('P' is repeated)
CUCUMBER	Rejected	('CU' is repeated)
ONION	Accepted	('ON' is not immediately repeated)
APRICOT	Accepted	

Write a program which inputs a password (between 1 and 10 upper case letters) and then prints 'Rejected' if the password is rejected, or 'Accepted' otherwise. Your program should then terminate.

Sample output

```
Password: BANANA
Rejected
```

Painting

A new university campus is coming up at Siruseri. At the centre of this campus is a piece of sculpture designed by a well known modern artist of Siruseri. The sculpture is made up of a set of concrete pillars of square cross-section of varying heights, placed next to each other.

The sculpture occupies a rectangular area of length M feet and width N feet. M×N pillars, each of cross section 1 foot by 1 foot, are placed in this area. These pillars of varying height. A description of such an sculpture can be given as an M×N array of numbers. The number at position (i,j) indicates the height of the pillar placed at position (i,j). For example, here is a description of a sculpture where M=2 and N=3.

```
3 2 3
2 1 1
```

The artist wants the entire sculpture to be painted in canary yellow. Notice that when two pillars are placed next to each other, a part of the surface of each of these pillars is no longer exposed to the outside and need not be painted. In the example above, only 9 square feet of the pillar at position (1,1) need to be painted. As a standalone pillar it has 13 square feet of paintable area (1 square foot on the top, and 3 square foot on each of its four faces). But, in the current arrangement, it is placed adjacent to two pillars which hide 2 square feet each of its exposed area. So, only 9 square feet need to be painted. If you add up the paintable areas for all the 6 pillars, you can see that a total of 34 square feet need to be painted.

Your task is to determine the total area that is to be painted given a description of the sculpture.

Input format
The first line of the input contains two integers, M and N, indicating the length and width of the sculpture in feet. This is followed by M rows of N positive numbers each. The jth number on line i+1 denotes height of the pillar at position (i,j).

Output format
A single positive integer indicating the total area to be painted.

Test data
You may assume $1 \leq M,N \leq 1000$.

Sample input

```
2 3
3 2 3
2 1 1
```

Sample output

```
34
```

Count on Cantor (75 points)

One of the famous proofs of modern mathematics is George Cantor's demonstration that the set of rational numbers is enumerable. The proof works by using an explicit enumeration of rational numbers as shown in the diagram below. In the diagram, the first term is 1/1, the second term is 1/2, the third term is 2/1, the fourth term is 3/1, the fifth term is 2/2, and so on.

```
1/1  1/2  1/3  1/4  1/5  ...
2/1  2/2  2/3  2/4
3/1  3/2  3/3
4/1  4/2
5/1
```

Input and Output

Write a program that will read a number in the range from 1 to 10^7 and will print the corresponding term in Cantor's enumeration as given below. The input list contains a single number per line and will be terminated by end-of-file.

Sample input

```
3
14
7
```

Sample output

```
TERM 3 IS 2/1
TERM 14 IS 2/4
TERM 7 IS 1/4
```

Strategic Defense Initiative (100 points)

"Commander! Commander! Please wake up commander!''
"... mmmph. What time is it?''
"4:07 am, Commander. The following message just arrived on the emergency zeta priority classified scrambler, marked your eyes only.''

You grudgingly take the letter, rub the sleep from your eyes, fleetingly wish that the 'Backer closed at an earlier hour, and start to read.

> Dear StarWars SDI Commander,
> Bad news, buddy. Crazy Boris had a bit too much vodka last night and when he woke up this morning, instead of the snooze button on his alarm clock, he...well, let me put it this way: we've got tons of nuclear missiles flying this way. Unfortunately, all that we have is a chart of the altitudes at which the missiles are flying, arranged by the order of arrivals. Go for it, buddy.
> Good luck.
> Secretary of Defense
> P.S. Hilly and Bill say hi.

To make things worse, you remember that SDI has a fatal flaw due to the budget cuts. When SDI sends out missiles to intercept the targets, every missile has to fly higher than the previous one. In other words, once you hit a target, the next target can only be among the ones that are flying at higher altitudes than the one you just hit. For example, if the missiles are flying toward you at heights of 1, 6, 2, 3, and 5 (arriving in that order), you can try to intercept the first two, but then you won't be able to get the ones flying at 2, 3, 5 because they are lower than 6. Your job is to hit as many targets as possible. So you have to quickly write a program to find the best sequence of targets that the flawed SDI program is going to destroy.

Russian war tactics are fairly strange; their generals are sticklers for mathematical precision. Their missiles will always be fired in a sequence such that there will only be one solution to the problem posed above.

Input and Output
Each input to your program will consist of a integer N, denoting the no of missiles launched followed by sequence of N integer altitudes, each on a separated by a space. Output from your program should contain the total number of targets you can hit, followed by the altitudes of those targets, in the order of their arrivals.

Sample Input

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

Sample Output

```
Max hits: 4
1 2 3 5
```

Sum It Up

Write a program that finds the largest contiguous sum of an array of numbers given as input to your program. The program should first accept the number of elements in the array as the input. Then it should accept the elements of the array. The output should be the largest contiguous sum.

The input to the program will be as follows; each on a separate line:

<number of elements in the array>
<element 0>
<element 1>
.
.
<element n-1>

The output should be in the following format; each on separate line:

<The largest contiguous sum>

Sample Input

8
100
-200
-300
40
400
-90
999
234

So here the number of elements in the array is 8. The elements are the next 8 numbers given as input. So here it is clear that the largest contiguous sum is the sum of the elements 40, 400, -90, 999, 234.

Sample Ouput

1583

Alpha-Sets

Write a program to accept a finite set of alphabets and all its nonempty subsets in lexicographic order. The set of alphabetic symbols will have at most eight characters. Note that the number of such non-empty subsets is equal to $(2^n - 1)$, where n is the number of elements in the given set.

For example, given a set {a,b,c}, the list of its non-empty subsets is:

{a}, {b}, {c}, {a,b}, {a,c}, {b,c}, {a,b,c}

Input Format

The input will consist of two lines.

First line indicating the number of elements in the input set say N.

Second will have a string of N non-blank alphabetical characters forming a set.

Note that the characters themselves will be given in alphabetical order.

Output Format

The output must list each of the required subsets on a seperate line (i.e., the no. of lines in the output must be equal to the no. of non-empty subsets of the given set). The characters in each line, also the subsets themselves should be in alphabetical order.

Sample Input

```
3
abc
```

Sample Output

```
a
ab
abc
ac
b
bc
c
```

Factors and Factorials

The factorial of a number N (written N!) is defined as the product of all the integers from 1 to N. It is often defined recursively as follows:

$1! = 1$
$N! = N * (N - 1)!$

Factorials grow very rapidly: 5! = 120, 10! = 3,628,800. Hence, they become difficult to calculate.

Write a program that will read in a number N (2 <= N <= 65000) and print out the last non zero digit of its factorial.

Input will consist of the number N.
Output will consist of the single digit, which is the last non zero digit of N!.

Sample Input / Output

(5,2)
(53,2)
(500,4)
(1000,2)
(5000,2)
(1234,8)
(10000,8)
(65000,6)
(1,1)
(10,8)

© 2006 Code Warriors | www.code-warriors.org
This version by Ankur Banerjee | www.ankurb.info