Indian National Olympiad in Informatics, 2007

Time: 3 hours *21 January, 2007*

Instructions

(a) You will have to return this question paper at the end of the examination with relevant parts filled out.

(b) There are two questions. You have to write working programs in Pascal, C or C++ to solve each of these questions.

- *Only your source code will be submitted for evaluation. Your program will be recompiled and run on the evaluation computer.*
- *Make sure your C/C++ programs compile with the GNU C compiler (`djgpp`, under Windows). Programs written using Turbo C++ may not compile and run in the evaluation environment and may hence lose marks.*

(c) At the end of each question, there is a space to indicate the location of the source code file for your solution. Please fill up this information without fail. Otherwise, your solution cannot be evaluated.

(d) All input for your programs will come from the keyboard. All output from your programs should be written to the screen.

(e) Please fill out your contact details on the reverse of this page as completely as you can. Use your serial number as reported in the ZIO results page as your roll number for INOI. Ask your centre supervisor if you do not know your roll number.

(f) At the beginning of each program, include a comment with your name, roll number and the problem number. For instance, if your name is Sachin Tendulkar, your roll number is 107 and the problem you are writing a C/C++ program for Problem 1, add a comment of the form

```
/*  Sachin Tendulkar, 107, Problem 1 */
```

at the top of your file.

This will help us if files accidentally get mixed up when being sent from the exam centres to the evaluation centre.

. . . P.T.O.

Contact details	
Roll Number:	
Name:	
Home Address:	PIN Code
Phone Number(s): (with STD code)	
Email address:	

Question 1   SMS Dictionary

Most mobile phones have their own private dictionary of words to allow users to type messages quicker. On a typical phone, each number key is assigned a subset of the alphabet $\{a, b, ..., z\}$: 2 is assigned the subset $\{a, b, c\}$, 3 is assigned $\{d, e, f\}$, 4 is assigned $\{g, h, i\}$, 5 is assigned $\{j, k, l\}$, 6 is assigned $\{m, n, o\}$, 7 is assigned $\{p, q, r, s\}$, 8 is assigned $\{t, u, v\}$ and 9 is assigned $\{w, x, y, z\}$.

When the user types a sequence of numbers, this sequence is mapped to all possible words that can be constructed from the key assignment. For instance, if the user types 66, this could correspond to any one of the letter sequences “mm”, “mn”, “mo”, “nm”, “nn”, “no”, “om”, “on” or “oo”. These letter sequences are looked up in the dictionary stored in the phone and all matches are reported. For instance, if the phone's dictionary contains only “on” and “no” from this set of sequences, the user will be offered a choice of “on” or “no” to insert in the message. Similarly, the input 4663 might be interpreted as either “$good$” or “$home$”. An input sequence may have a unique interpretation—for example, the only word in the dictionary matching the input 28 may be “at”. Other sequences may not match any word in the dictionary—for instance, 99999.

Your task is the following. Given the typical assignment from number keys to letters of the alphabet given above and given a dictionary of words, report the input sequence that matches the largest number of words in the dictionary. For example, if the dictionary consists of the words $\{at, on, good, no\}$ then the answer is 66, because 66 matches both “on” and “no” while no other input matches more than one word in the dictionary. On the other hand, with the dictionary $\{at, on, good, no, home, gone\}$, the answer is 4663, because 4663 matches three words, “$good$”, “$home$” and “$gone$” in the dictionary.

Input format

The first line of input is an integer M, the number of words in the dictionary. This is followed by M lines of input. Each line contain one word from the dictionary, where a word is sequence of characters from the lowercase alphabet $\{a, b, c, ..., z\}$.

Note: Each word in the dictionary is, in general, an arbitrary sequence of letters from $\{a, b, c, ..., z\}$. In particular, it is not assumed that the words stored in the dictionary are valid words in English or any other language.

Output format

A single line containing the input sequence that matches the maximum number of words in the dictionary. If multiple input sequences qualify for the maximum number of matches, it suffices to report any one.

Test data

For all inputs, $1 \leq M \leq 100000$. Each word in the dictionary is at most 8 characters long. In 50% of the inputs, $1 \leq M \leq 1000$.

Example

Here is the sample input and output corresponding to the example discussed above.

Sample input 1

```
4
at
on
good
no
```

Sample output 1

```
66
```

Sample input 2

```
6
at
on
good
no
home
gone
```

Sample output 2

```
4663
```

Note: Your program should not print anything other than what is specified in the output format. Please remove all diagnostic print statements before making your final submission. A program with extraneous output will be treated as incorrect!

Important

Indicate the location of the source code file for your solution to Question 1 in the box below. *If you fail to do this, your solution cannot be evaluated!*

Source file:

Question 2 The Middleman

Gundu the middleman wants to purchase grain from the farm owned by Ramu the farmer. Ramu's farm is divided into an $M \times N$ grid of fields. Gundu has estimated the profit from each field. Some of the fields will yield a loss because the quality of the grain from these fields is so poor that the cost of harvesting the grain overruns the price at which it can be sold. Moreover, Ramu has marked some of his fields as “Not for Sale” as he plans to use the seeds from these fields for his seed bank.

Ramu knows that Gundu is interested in purchasing grain only from the most profitable fields. So, he has imposed a restriction saying that Gundu must buy the grain from a square subgrid of the grid of fields. Moreover, this subgrid must not contain any field marked “Not for Sale”.

Here is a sample farm with a 6×6 grid of fields. Each field is labelled by its expected profit (a positive entry) or loss (a negative entry). The fields at locations (2,5) and (4,2) have been reserved by Ramu as “Not for Sale”. It turns out that any field that is available for sale has a nonzero expected profit or loss. This allows us to use the value 0 to identify the “Not for Sale” fields.

```
8 -3  4 -1  5 18
2  3  4 -3  0  7
6 -2  3  1  1  3
3  0  2 -1  3  1
2  2  2  2  2  2
1  3  1 -1  3  1
```

In this example, the maximum profit is obtained by picking all the fields in the square with (3,3) as the top left corner and (6,6) as the bottom right corner. The total profit for this subgrid is 25.

Your task is to write a program that reads a description of the farm and computes the maximum profit that Gundu can make.

Input format

The first line contains two integer M and N. This is followed by M lines each of which contains N entries, separated by a blank. Each of these entries is an integer in the range $-1000, -999, \ldots, -1, 0, 1, \ldots 999, 1000$.

Output format

A single integer indicating the maximum profit that the Gundu can make.

Test data

For all inputs, $1 \leq M, N \leq 200$.

Example

Here is the sample input and output corresponding to the example discussed above.

Sample input

```
6 6
8 -3  4 -1  5 18
2  3  4 -3  0  7
6 -2  3  1  1  3
3  0  2 -1  3  1
2  2  2  2  2  2
1  3  1 -1  3  1
```

Sample output

```
25
```

Note: Your program should not print anything other than what is specified in the output format. Please remove all diagnostic print statements before making your final submission. A program with extraneous output will be treated as incorrect!

Important

Indicate the location of the source code file for your solution to Question 2 in the box below. *If you fail to do this, your solution cannot be evaluated!*

Source file: