

Sample Problems

PLEASE READ THE INSTRUCTIONS ON THIS PAGE CAREFULLY.

GENERAL RULES:

1. Your program should accept two command line arguments. The first command line argument to your program will be the name of the input file. The second command line argument to your program will be the name of the output file. Output should follow the specifications given in each problem.
2. All programs will be re-compiled using gcc(GNU Compiler Collection) to testing with the data.
3. Non-standard libraries cannot be used in your solutions. No header files other than `stdio.h`, `string.h`, `stdlib.h`, and `math.h` are allowed. No system calls are allowed. Function names also cannot use names of system calls.
4. Programming style is not considered in this contest. You are free to code in whatever style you prefer. Documentation is not required.
5. The allowed programming language is C. You should submit a single C File containing your entire code. Multiple file submission for a single problem in a single attempt is not allowed.

PROBLEM 1: Last number

The expression $N!$, read as "N factorial," denotes the product of the first N positive integers, where N is nonnegative. So, for example,

$N \ N!$

0 1

1 1

2 2

3 6

4 24

5 120

10 3628800

For this problem, you are to write a program that can compute the last non-zero digit of any factorial for ($0 \leq N \leq 10000$). For example, if your program is asked to compute the last nonzero digit of $5!$, your program should produce "2" because $5! = 120$, and 2 is the last nonzero digit of 120.

Input Format

Input to the program is a series of nonnegative integers not exceeding 10000, each on its own line with no other letters, digits or spaces. For each integer N , you should read the value and compute the last nonzero digit of $N!$.

Output Format

For each integer input, the program should print exactly one line of output. Each line of output should contain the last non-zero digit of $N!$.

Sample Input

1

2

26

125

3125

9999

Sample Output

1

2

4

8

2

8

Problem 2: Rotating Words

You are required to rotate a word a certain amount. For example, to rotate the word "Computer" by 1 results in "rCompute". Rotating it two more times gives you "terCompu".

Input Format

The first line contains integer t , the number of test cases. Each test case begins with a line contains the word (which will not have more than 15 letters). The next line contains an integer n , which will be less than the length of the word. You must rotate the word n times.

Output Format

The output will be the rotated word.

Sample Input

```
2
Computer
3
Program
1
```

Sample Output

```
terCompu
mProgra
```

PROBLEM 3: Mirrored Palindrome

A regular palindrome is a string of numbers or letters that is the same forward as backward. For example, the string "ABCDEDCBA" is a palindrome because it is the same when the string is read from left to right as when the string is read from right to left. A mirrored string is a string for which when each of the elements of the string is changed to its reverse (if it has a reverse) and the string is read backwards the result is the same as the original string.

For example, the string "3AIAE" is a mirrored string because "A" and "I" are their own reverses, and "3" and "E" are each others' reverses.

A mirrored palindrome is a string that meets the criteria of a regular palindrome and the criteria of a mirrored string. The string "ATOYOTA" is a mirrored palindrome because if the string is read backwards, the string is the same as the original and because if each of the characters is replaced by its reverse and the result is read backwards, the result is the same as the original string. Of course, "A", "T", "O", and "Y" are all their own reverses.

A list of all valid characters and their reverses is as follows.

Char Rev Char Rev Char Rev

A A M M Y Y

B N Z 5

C O O 1 1

D P 2 S

E 3 Q 3 E

F R 4

G S 2 5 Z

H H T T 6

I I U U 7

J L V V 8 8

K W W 9

L J X X

* Note that 0 (zero) and O (the letter) are considered the same character and therefore ONLY the letter "O" is a valid character.

Input Format

Input consists of strings (one per line) each of which will consist of one to twenty valid characters. There will be no invalid characters in any of the strings. Your program should read to the end of file.

Output Format

For each input string, you should print the string starting in column 1 immediately followed by exactly one of the following strings.

STRING

CRITERIA

" -- is not a palindrome."

if the string is not a palindrome
and is not a mirrored string

" -- is a palindrome."

if the string is a palindrome and
is not a mirrored string

" -- is a mirrored string."

if the string is not a palindrome
and is a mirrored string

" -- is a mirrored palindrome."

if the string is a palindrome and
is a mirrored string

Note that the output line is to include the -'s and spacing exactly as shown in the table above and demonstrated in the Sample Output below. Note a new line at the end of the input as well as the output files.

Sample Input

NOTAPALINDROME

ISAPALINILAPASI

2A3MEAS

ATOYOTA

Sample Output

NOTAPALINDROME -- is not a palindrome

ISAPALINILAPASI -- is a palindrome

2A3MEAS -- is a mirrored string

ATOYOTA -- is a mirrored palindrome

Problem 4: Triangles

```
    5
   2 7
  1 8 1
 8 2 3 2
5 2 4 8 9
1 2 3 4 4 4
```

The triangle game consists of a triangle of numbers and a counter, which begins the game at the top of the triangle. The aim of the game is to move from the top to the bottom and to get the highest total score. Your score is increased by each number you pass through. So your score starts at zero, then you add on the number at the top of the triangle, then you add all the other numbers you pass through on your way to the bottom.

There are 2 ways to move, you can move the counter diagonally down and left or diagonally down and right. E.g. starting at the top, on number 5, (your score begins at 5) you can move down-left to 2 OR down-right to 7.

You are to write a program that will output the maximum total score that is attainable on a given triangle.

Input Format

The first line contains the number of test cases.

Each test case consists details of triangle.

The first line of each test case is N, the number of rows in the triangle.

The following N lines contain the values of the triangle positions.

i.e.

Line 2 contains the number in the top (1st) row

Line 3 contains the 2 numbers in the 2nd row

Line 4 contains the 3 numbers in the 3rd row

...

Line N+1 contains the N numbers in the Nth row

(2 ≤ N ≤ 100)

The value at any position will be a non-negative integer.

Output Format

The output is a single integer: the maximum score attainable using the given triangle.

This score will not exceed 32767.

Sample Input:

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

6
5
2 7
1 8 1
8 2 3 2
5 2 4 8 9
1 2 3 4 4 4

Sample Output:

10
35