



# Novice Programming Problems

Brought to you by APP

# Instructions

All programs will accept input through standard input and output to standard out only. Output must match the desired format exactly. This means there should be no debug statements that are printed and that the format must match exactly as specified. Additionally, input will be given to you exactly as expected. If the problem says input will be an integer, you do not have to check to see if a character is entered. You can solve any problem using any combination of the languages Java, C#, and/or C++. You may work in a team of up to 3 people or compete by yourself. If you need clarification on a problem you can submit a request for clarification directly through PC<sup>2</sup>. The team who solves the most problems is the winner. In the case of a tie, the team who solved the last problem first will be the winner.

# Reverse a Word

Being able to effectively use strings is very important. Write a program that accepts user input through standard input and reverses it. This program will continue getting input until “done” is entered.

## Sample Input

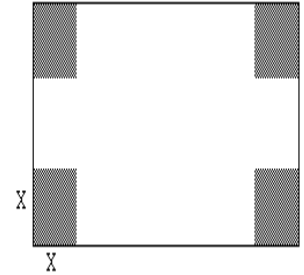
```
Hello World  
This will be reversed  
More examples  
1234567  
hahahaha  
done
```

## Sample Output

```
dlroW olleH  
desrever eb lliw sihT  
selpmaxe eroM  
7654321  
ahahahah
```

# Largest Box

Write a program that will continually accept integers until -1 is entered. All other numbers that are entered will be positive numbers representing the length of one side of a square sheet of cardboard. For each number entered, the program should output the integer value of the length that should be cut out of the corners of the sheet that would result in the largest volume of the box once folded, followed by the word done.



## Sample Input

```
3721
8
-1
```

## Sample Output

```
620
1
done
```

# Prime Numbers

Working with prime numbers. Write a program that accepts a list of long numbers and is terminated with “done”. This program should output either “prime” if the number is a prime number, or “not prime” if the number is not prime. A prime number is a number is a natural number greater than 1 that has no positive divisors other than 1 and itself.

## Sample Input

```
21
17
899809363
5973452647
1
49979693
4398042316799
done
```

## Sample Output

```
not prime
prime
prime
not prime
not prime
prime
prime
```

# Lexicographic List

Given an arbitrary fragment of text, write a program that produces a list of all words occurring in this text in lexicographic (dictionary) order and indicating the frequency of the occurrence of each word. Input text may also include punctuation marks (.,;:-!?"") and whitespace characters.

## Sample Input

The quick brown fox jumps over the lazy dog.

## Sample Output

brown (1)  
dog (1)  
fox (1)  
jumps (1)  
lazy (1)  
over (1)  
quick (1)  
the (2)

# Check ISBN Number

ISBN 10 numbers last digit is a check digit. The check digit determines whether or not the ISBN number is valid or not. Write a program that tests n number of ISBN numbers and determines whether or not they are valid. The first input number that represents the number of test cases. The program will then accept that many possible ISBN numbers in the form  $x_0x_1x_2x_3x_4x_5x_6x_7x_8x_9$  where  $x_9$  is the check digit. Your program will output whether or not the ISBN number is valid or not. The check digit is checked with the formula

$$x_9 = 11 - (10x_0 + 9x_1 + 8x_2 + 7x_3 + 6x_4 + 5x_5 + 4x_6 + 3x_7 + 2x_8) \bmod 11$$

\*If  $x_9 = 10$  it is replaced with an X instead.

## Sample Input

```
4
0205126693
542345754X
000000457X
1234567890
```

## Sample Output

```
valid
invalid
valid
invalid
```

# Simple Encryption

You have been tasked with encrypting messages for a secret underground society. They will provide you with a numerical key representing the interval in which the message is to be split, followed by the message itself. After the message has been split into segments you must recombine them to form a new message by appending the first character of each segment, proceeding to the next until all characters have been appended. Spaces should be output as '\*' to further encrypt the message.

## Sample Input

### Ex 1:

4 Hello World how are you

### Ex 2:

2 I like 2 program

## Sample Output

### Ex 1:

Horhay\*loroIWdweulo\*\*\*

### Ex 2:

Ilk\*\*rga\*ie2porm



# CC Food

The CC Commons is an eatery on campus in which each student may place an order for their meal. With numerous amounts of students eating at the CC Commons they need to keep track of what they foods have been ordered, how long it takes them to cook each item, and the quantity that can be made of each food item at a time(for example you can cook more than one order of fries at a time). Note that two separate food items can be made at the same time i.e. fries and a burger.

Write a program that first accepts the number of different food items the CC Commons has. Next enter in those food items followed by the time it takes to cook them and the quantity that can be cooked at the same time. Then enter in the number of orders placed. Enter in each of the food items followed by the quantity. Separate each order by entering 00 at the end of it. When the last 00 is entered the output of the total cook time of all orders should be shown.

## Sample Input

```
3
burger 7 3
fries 4 5
salad 5.5 1
2
burger 3
fries 1
00
salad 1
burger 1
fries 1
00
```

## Sample Output

```
14
```