2nd Mid-term Exam

C COMPUTER PROGRAMMING LABORATORY（I）

November 23, 2023

Exam rules

- Only Dev-C++ can be used for the exam.
- If your codes cannot be compiled by Dev-C++, it is considered as syntax error.
- Please write all your codes as a c source file named after your student ID.
  example: M113040076_1.c、M113040076_2.c…
- When submitting the assignment, upload all six questions together to the university's assignment submission area, without the need to compress them.
- **In a single question, the score is either full credit or zero.**
- Please check whether your codes can be compiled and output the desired result before submitting to National Sun Yat-sen Cyber University.
- No reason for late submission.
- It is forbidden to search for information during the exam.
- **The cheaters will get zero point.**

a. (20 points)

Description

Hoz has invented a system called HMail, which allows sending text messages from a mobile phone to oneself ten years ago. However, he forgot one crucial detail: smartphones didn't exist ten years ago!

At that time, mobile phones had traditional nine-key keypads, causing this system to require a specific format for messages to be transmitted correctly. The traditional nine-key keypad had the following English combinations for digits 2 to 9 in order:

"abc", "def", "ghi", "jkl", "mno", "pqrs", "tuv", "wxyz"

For example, if you wanted to type the word "hoz," you would press the 'h' key twice, the 'o' key three times, and the 'z' key four times. Therefore, in the HMail system, you would input "h2o3z4" to send the message correctly.

Requirement

Please use continuous input.

Please use **fgets()** to read the SMS string.

The print format must be like the picture below.

Notice

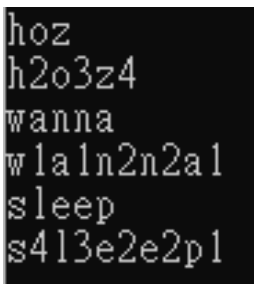
Output the entered content in the format that can be sent through HMail, and each output should be followed by a newline ("\n").

Input at most 100 messages, and each message should be separated by a newline ("\n").

The input message length will not exceed 100 lowercase English letters, and the message content will consist only of lowercase English letters without numbers or spaces.

Hint

while(fgets(arr, length of array, stdin) != NULL



Test Data

(1).

hoz

wanna

sleep

(2).

apple

banana

cat

dog

b. (20 points)

Description

Hoz is a math teacher at National Sun Yat-sen Elementary School. He often stays up late to grade students' assignments, sacrificing his quality of life. These mathematical problems could be solved by writing a simple program to help with scoring. Can you, the intelligent one, assist Hoz?

Please use continuous input to read the data, verify if the entered mathematical expressions are correct, and keep track of the total number of correct answers.

Requirement

Use continuous input to read mathematical expressions.

The print format must be like the picture below.

Notice

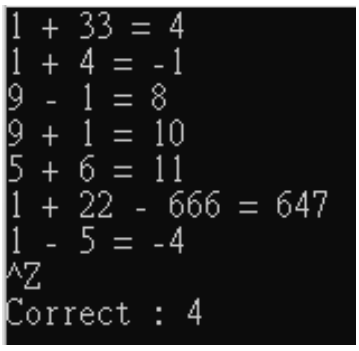
The arithmetic operations are limited to addition and subtraction.

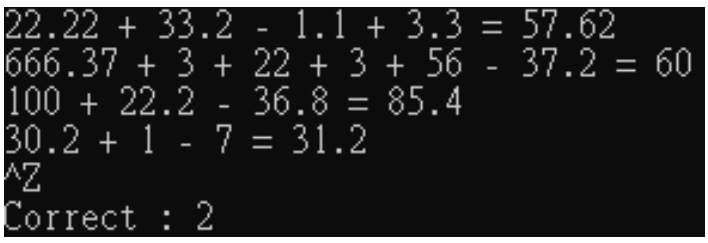
There is a space between numbers and symbols.

Each line does not exceed 1024 characters.

Hint

while(fgets(arr, length of array, stdin) != NULL)





Test Data

(1).

1 + 33 = 4

1 + 4 = -1

9 - 1 = 8

9 + 1 = 10

5 + 6 = 11

1 + 22 - 666 = 647

1 - 5 = -4

(2).

22.22 + 33.2 - 1.1 + 3.3 = 57.62

666.37 + 3 + 22 + 3 + 56 - 37.2 = 60

100 + 22.2 - 36.8 = 85.4

30.2 + 1 - 7 = 31.2

c. (20 points)

Description

Write a program to remove leading and trailing spaces from a sentence, and eliminate consecutive spaces between words in a normal English sentence. In a well-formed English sentence, there should be only one space between words, and no spaces at the beginning or end.

Requirement

Please use continuous input.

The print format must be like the picture below.

Notice

Each line of input contains at most 128 characters.

Please refer to the example input and output for details.

No need to consider punctuation(標點符號) issues; the test data will not include punctuation.

Please review the content of the description carefully once again.

Hint

while(fgets(arr, length of array, stdin) != NULL)

```
    I  want   to buy   iPhone15   Pro
I want to buy iPhone15 Pro
  Let   it   go
Let it go
```

```
  Only need    the   light when it   is burning low
Only need the light when it is burning low
Only       miss   the sun       when it starts to          snow
Only miss the sun when it starts to snow
```

Test Data

(1).

I want to buy iPhone15 Pro

Let it go

(2.)

Only need the light when it is burning low

Only miss the sun when it starts to snow

d. (20 points)

Description

Given an unspecified input, if the input can be interpreted as an integer, please output it as an integer. Otherwise, attempt to remove illegal characters and output the corrected input as an integer. Please analyze or correct the ambiguous input content (excluding control characters and space characters such as \t \n \v \f \r) and output an integer accordingly.

Requirement

Please use continuous input.

The print format must be like the picture below.

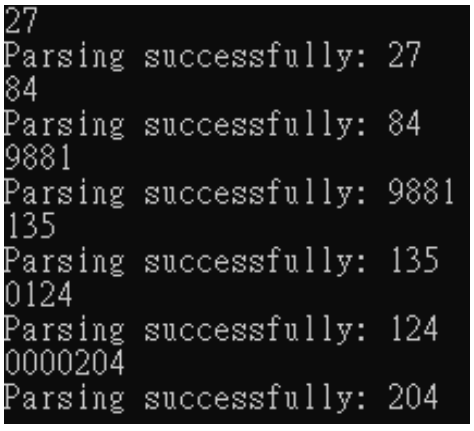
Notice

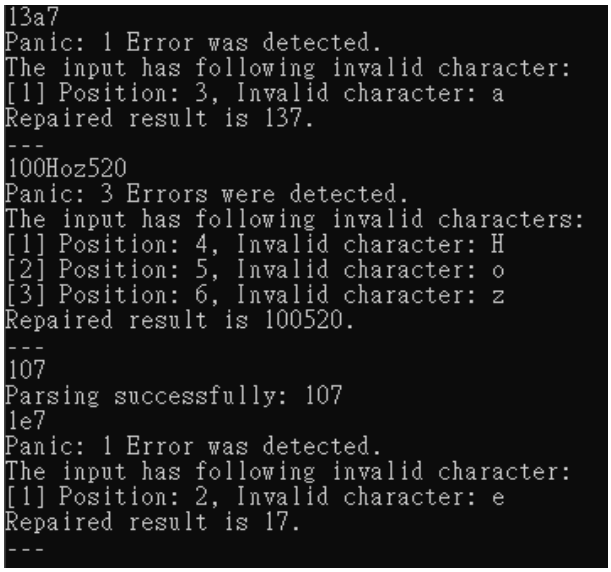
The input contains no more than 40 characters.

Please refer to the example input and output for details.

Hint

while(fgets(arr, length of array, stdin) != NULL)





Test Data

(1).

27

84

9881

135

0124

0000204

(2).

13a7

100Hoz520

107

1e7

e. (20 points)

Description

Let the user to input an array of integers until reaching End-of-file. The task is to sort all prime numbers in the array in descending order based on their relative positions, without affecting the positions of other elements.

Requirement

Please use continuous input.

The print format must be like the picture below.

Notice

The array's length will not exceed 1024 integers.

When the input is “0”, output "Finish!\n" and terminate the program.

Each number will be in the range of 0 to 10000.

Hint

```
Input Array: 1 2 3 4 5 6 7 8 9 10
^Z
Prime sorted Array: 1 7 5 4 3 6 2 8 9 10

Input Array: 2 4 6 8 3 5 7 11 13 17 19
^Z
Prime sorted Array: 19 4 6 8 17 13 11 7 5 3 2

Input Array: 10 8 6 4 2 0
^Z
Prime sorted Array: 10 8 6 4 2 0
```

Test Data

(1).

1 2 3 4 5 6 7 8 9 10

2 4 6 8 3 5 7 11 13 17 19

10 8 6 4 2 0

(2).

9431 9433 9437 9439 9461 9463 9467 9473 9479 9491 9497 9511 9521 9533 9539 9547 9551 9587 9601 9613 9619 9623 9629 9631 9643 9649 9661 9677 9679 9689 9697 9719 9721 9733 9739 9743 9749 9767 9769 9781 9787 9791 9803 9811 9817 9829 9833 9839 9851 9857 9859 9871 9883 9887 9901 9907 9923 9929 9931 9941 9949 9967 9973

Output should be 9973 9967 9949 9941 9931 9929 9923 9907 9901 9887 9883 9871 9859 9857 9851 9839 9833 9829 9817 9811 9803 9791 9787 9781 9769 9767 9749 9743 9739 9733 9721 9719 9697 9689 9679 9677 9661 9649 9643 9631 9629 9623 9619 9613 9601 9587 9551 9547 9539 9533 9521 9511 9497 9491 9479 9473 9467 9463 9461 9439 9437 9433 9431

7624 5468 237 1892 456 873 9824 352 7692 345 6781 2345 9012 876 2394 567 981 2345 672 891 5623 8912 3456 8923 124 567 2309 482 9034 1245 653 2389 4567 189 2390 567 234 981 238 4893 1098 7654 2398 456 123 897 453 7692 3456 981 234 657 8923 124 567 2301 345 8923 765 982 345 9810 567 8923 234 897 453 6789 1098 567 8923 105 678 235 678 123 789 456 238 109 234 5678 876 2394 8923 876 1234 5678 123 678 453 8923 1098 345 678 1230 876 567 893 432 1098 5678 2301

Output should be 7624 5468 237 1892 456 873 9824 352 7692 345 8923 2345 9012 876 2394 567 981 2345 672 891 8923 8912 3456 8923 124 567 8923 482 9034 1245 8923 8923 8923 189 2390 567 234 981 238 4893 1098 7654 2398 456 123 897 453 7692 3456 981 234 657 6781 124 567 2301 345 5623 765 982 345 9810 567 4567 234 897 453 6789 1098 567 2389 105 678 235 678 123 789 456 238 2309 234 5678 876 2394 653 876 1234 5678 123 678 453 109 1098 345 678 1230 876 567 893 432 1098 5678 2301

f. (20 points)

Description

Read an unsorted array of integers from standard input, where the values are input until the end-of-file is reached. The array's length will not exceed 1024 integers. First, you need to sort the array and then use binary search to find the value in the sorted array that is closest to 10,000. If there are two values equally close to 10,000, output the larger one.

Requirement

Please use continuous input.

The print format must be like the picture below.

Notice

This problem must be solved using binary search.

Read signed integers from standard input until End-of-file.

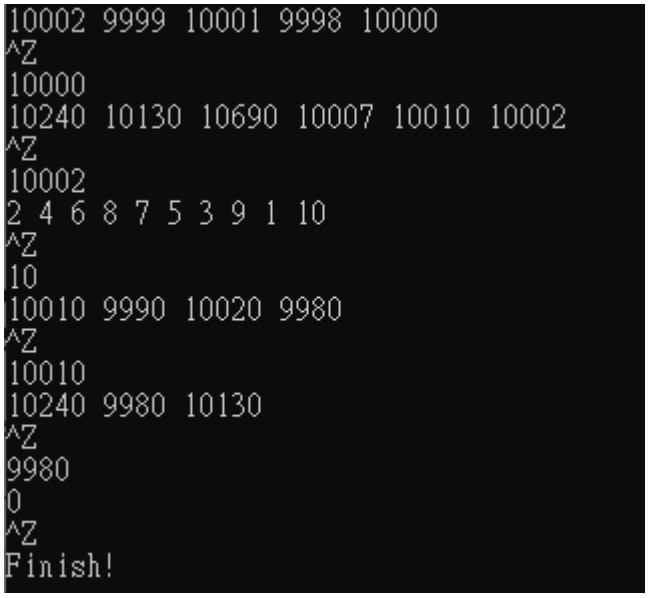
Output the value closest to 10000, followed by a newline. If there are two equally close values, output the larger one.

When the input is “0”, output "Finish!\n" and terminate the program.

The qsort() function is banned.

Hint

#include<string.h> memset(Array, ‘\0’, 1024)



Test Data

10002 9999 10001 9998 10000

10240 10130 10690 10007 10010 10002

2 4 6 8 7 5 3 9 1 10

10010 9990 10020 9980

10240 9980 10130

0