Final Exam

C COMPUTER PROGRAMMING LABORATORY（I）

December 21, 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

Please identify the largest number present in the input article and print it.

An article containing only positive numbers.

Requirement

Please use continuous input.

Only the use of fgets() is allowed, while others such as scanf() and getchar() are prohibited.

Please write a getword() function to handle each word, and getword() must be written as an independent function and cannot be written within the main program.

The print format must be like the picture below.

Notice

Output the largest number found in the article.

Please take note of the potential overflow issue.

Hint

(1) getword()

(2) atoi()

(3) isspace()

Input

輸入

```
There are 100 apples and 800 oranges.
There are 1000 students.
There are 1000900 students.
There are 1000 students.
```

Output

輸出

```
1000900

```

Test Data

There are 100 apples and 800 oranges.

There are 1000 students.

There are 1000900 students.

There are 1000 students.

b. (20 points)

Description

Please use a data structure to record students' scores in Chinese, English, Mathematics, Physics, and Social Studies.

Additionally, implement a function to sort students based on their total scores.

Requirement

It must use a structure to record students' scores.

The print format must be like the picture below.

Notice

The input sequence includes scores for Chinese, English, Mathematics, Physics, and Social Studies, separated by spaces. Scores may be missing.

There will be at most ten students, when encountering EOF, print the total sorted scores result.

If a score is missing, please fill it in with 0 points.(eg. Test Data 2)

Input

輸入

```
80 20 30 40 50
30 40 70 70 30
20 30 30 100 100

```

Output

輸出

```
20 30 30 100 100
30 40 70 70 30
80 20 30 40 50

```

Test Data

Input:

(1)

80 20 30 40 50

30 40 70 70 30

20 30 30 100 100

(2)

80 20

30

100 100 100

Output:

(1)

20 30 30 100 100

30 40 70 70 30

80 20 30 40 50

(2)

100 100 100 0 0

80 20 0 0 0

30 0 0 0 0

c. (20 points)

Description

Given an English article, please refer to the following guidelines: Words with the same alphabetical order (regardless of letter case) are considered the same word.

For instance, THAI, thai, Thai are all considered one word. Inputs consist solely of words, without punctuation.

Words are separated by space characters (ASCII 0x20 (32)).

Count the total occurrences of unique words.

Input an English article consisting of characters limited to A~Z(ASCII 65~90） and a~z(ASCII 97~122).

Requirement

The print format must be like the picture below.

Notice

Each word's length does not exceed 1000 characters.

The number of words does not exceed 1000.

Hint

(1) #include<ctype.h> toupper()、tolower()

Input

輸入

```
apple juice Banana Juice Good GOOD Good
```

Output

輸出

```
Number of terms in the dictionary is 4.

```

Test Data

Input:

(1) apple juice Banana Juice Good GOOD Good

(2) Wakanda Forever wakanda wakanda FOREVER

Output:

(1) Number of terms in the dictionary is 4.

(2) Number of terms in the dictionary is 2.

d. (20 points)

Description

Please build a tiny database, which is using double character pointer variable to store the information. There are two kinds of commands, the syntaxes are described in the following statements.

1. INSERT <Data Length> <Data>
2. OUTPUT

We will give you an integer at the beginning to stand for the number of data rows; INSERT will store the <Data> into your tiny database, the data length will be indicated at <Data Length> field. If you encounter the OUTPUT command, you should output all the data in reverse sequence and end this program.

ptr **OK

*ptr **OK**

*ptr[] **NO**

ptr[] **NO**

ptr[][] **NO**

Requirement

Using array is prohibited.

The print format must be like the picture below.

void *malloc(size_t size)

Notice

Please note that you need to use (double) pointer variables instead of using pointer array or 2-D array to solve this problem; any form of array will be strictly prohibited during formal exam.

Input

輸入

```
3
INSERT 3
abc
INSERT 5
abcde
INSERT 6
abcdef
OUTPUT

```

Output

輸出

```
abcdef
abcde
abc

```

Test Data

3

INSERT 3

abc

INSERT 5

abcde

INSERT 6

abcdef

OUTPUT

e. (20 points)

Description

The task you provided involves selecting 7 characters from the set S={1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,G} to form a string of length 7. Each character can be chosen at most 4 times. The objective is to determine whether this string is in a "ready" state. If it is, then list each tile it is waiting for, following the order of elements in set S. If the string is not in a ready state or if there are format errors (such as a string length not equal to 7, presence of elements not in set S, a single tile appearing more than 4 times, etc.), output -1.

A "ready" state is defined as the condition where adding one more tile would complete a set of 8 tiles, forming a winning hand. An 8-tile complete set consists of a pair (2 identical characters) and two sets of "triplets." For English letters, a "triplet" is three identical letters (e.g., AAA, BBB, FFF), while for Arabic numerals, it can be three identical numbers (e.g., 111, 222, 888) or three consecutive numbers (e.g., 123, 345, 789). It's important to note that 9 and 1 are not considered consecutive, meaning combinations like 912 and 891 are not valid "triplets."

Requirement

Please use continuous input.

Only the use of fgets() is allowed, while others such as scanf() and getchar() are prohibited.

The print format must be like the picture below.

Notice

If the input string is correctly formatted (length 7, composed of elements from set S, and each character appearing at most 4 times) and is in a ready state, output "Ready for" followed by all the target tiles the input string is waiting for (in the order of elements in set S, separated by spaces). Otherwise, output -1.

Hint

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

```
12345AABD
-1

CD33
-1

123AA78
Ready for 6 9

8AAABBB
Ready for 8

1234GGG
Ready for 1 4

2333BBB
Ready for 1 2 4

4445666
Ready for 3 4 5 6 7

68CCCDD
Ready for 7

2223456
Ready for 1 3 4 6 7

35AABBB
Ready for 4

0800449
-1

1222223
-1

2333345
Ready for 1 2 4 5

123BBDD
Ready for B D
```

Test Data

12345AABD

CD33

123AA78

8AAABBB

1234GGG

2333BBB

4445666

68CCCDD

2223456

35AABBB

0800449

1222223

2333345

123BBDD

f. (20 points)

Description

Customize an encryption method by repeating the original password twice and then inserting a few characters from the front.

Let the user to input an encrypted message string and print out the original password and its length.

Requirement

Please use continuous input.

Only the use of fgets() is allowed, while others such as scanf() and getchar() are prohibited.

The print format must be like the picture below.

Notice

Each row contains a string with a length not exceeding 1000 representing the encrypted message.

The characters in the string can only be uppercase letters 'A"Z', lowercase letters 'a"z', and numbers '0'~'9'.

You can assume that the input messages are all valid.

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

Hint

```
aaa

a
1

abcdefghijklmnopqrstuvwxyzthispasswordwordthispasswordword

thispasswordword
16

abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz

abcdefghijklmnopqrstuvwxyz
26

tmtthetmttmtntmtotmtntmtotmtn

tmtotmtn
8

abcdefghijklmnopqrstuvwxyz123456789123456789

123456789
9

a1b2c3d4e5f6x24y25z26x24y25z26

x24y25z26
9

hozandhowardhozandhowardsupershysupershy

supershy
8

0
Finish!
```

Test Data

aaa

abcdefghijklmnopqrstuvwxyzthispasswordwordthispasswordword

abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz

tmtthetmttmtntmtotmtntmtotmtn

abcdefghijklmnopqrstuvwxyz123456789123456789

a1b2c3d4e5f6x24y25z26x24y25z26

hozandhowardhozandhowardsupershysupershy