

Introduction to Computers and Programming LAB-Quiz2

2016/12/14 Time: 2 hrs

※ Please create a new folder and name it as your Student ID. Inside the folder, your file format will be Q_1.c, Q_2.c, etc. (There will be score deduction for wrong file name).

※ Any C library is forbidden, except for <stdio.h>, <stdlib.h> and <string.h>

※ No Internet. No discussions. Be honest with your own works.

※ The class is for C language, so do not use C++.

※ If your program cannot be compiled, you will get zero point for the question.

※ Follow the input/output format in examples.

1. Alphabet-ordered linked list

Implement a string linked list in alphabetical order. When a user enters a string (length < 100, and in only lower case a ~ z), insert the string into the linked list in alphabetical order. After the insertion, show the list from the first to the last. **Your program should be able to accept strings continuously; moreover, it must be implemented by using linked lists, any declaration of static 2-D character array is forbidden.**

Hint:

1. *strcmp(str1, str2)* has three kinds of return value: <0, 0 and >0.

2. Dictionary order is the alphabetical order that usually used in dictionaries.

For example, a -> aa -> abandon -> abort -> abortion -> ache -> -> b -> c ->....

```
Enter a string : a
The list content: a

Enter a string : aa
The list content: a -> aa

Enter a string : aaa
The list content: a -> aa -> aaa

Enter a string : abc
The list content: a -> aa -> aaa -> abc

Enter a string : z
The list content: a -> aa -> aaa -> abc -> z

Enter a string : x
The list content: a -> aa -> aaa -> abc -> x -> z

Enter a string : xray
The list content: a -> aa -> aaa -> abc -> x -> xray -> z

Enter a string : zoo
The list content: a -> aa -> aaa -> abc -> x -> xray -> z -> zoo

Enter a string : abandon
The list content: a -> aa -> aaa -> abandon -> abc -> x -> xray -> z -> zoo
```

2. Dynamic String

Dynamic memory allocation allows us to allocate a memory block at the execution time, and because the allocated memory block is contiguous, we can use it as an array. However, dealing dynamic memory should be handled carefully. Carelessness can cause memory leak, or even crash a program. We intend to implement a new data structure called Dynamic String as well as the related supportive functions for it. **(Your program should not crash in any situation!!)**

About the Structure of **Dynamic String (DynaStr)**

capacity: the maximum number of elements that data can store

length: length of string

data: the pointer to the dynamic allocated memory spaces

About the behaves of **supposed functions**

Please refer to TA's template for the implement detail of each function.

Note:

You **have to** follow TA's template and finish these six functions.

You should **NOT** store '\0' character.

You **cannot** declare new **global variable**, **static variable** or **modify main function** **otherwise you will get 0 point.** (but you can add new functions and local variables)

Examples:

```
-----
| 1. Create a Dynamic String.
| 2. Set the value of the Dynamic String.
| 3. Append a C-String into the Dynamic String.
| 4. Show the information of the Dynamic String.
| 5. Clear the Dynamic String.
| 6. Delete the Dynamic String.
| 7. Exit.
|-----
What are you going to do? -> 1
What are you going to do? -> 2
Please input string to set: abc
What are you going to do? -> 4
abc(length:3, capacity:3)
What are you going to do? -> 3
Please input string to append: 123
What are you going to do? -> 4
abc123(length:6, capacity:6)
What are you going to do? -> 5
What are you going to do? -> 4
(length:0, capacity:6)
What are you going to do? -> 2
Please input string to set: a
What are you going to do? -> 4
a(length:1, capacity:6)
What are you going to do? -> 6
```

```

What are you going to do? -> 3
Please create a Dynamic String first!!
What are you going to do? -> 1
What are you going to do? -> 3
Please input string to append: hello
What are you going to do? -> 4
hello(length:5, capacity:5)
What are you going to do? -> 7

```

3. List of winners

Design a program that can build a table of records for the people who buy the game tickets. It has the four function below:

Functions:

- (1) **assign**: When you type **assign**, the program needs to go into the assign mode that can repeatedly add new profile, until input an empty line.
- (2) **show**: When you type **show**, the program needs to show the table of the profiles.
- (3) **sort order_kind**: When you type **sort**, the program needs to show the table of the profiles in the order of *order_kind*. There are 2 *order_kind*:
 - (a) name: sort the profile by the name in ascending alphabetical order.
 - (b) prize: sort the profile from the largest amount of money to the smallest.
 - **sort** function just shows a sorted table of profiles in the order requested, but it does NOT modify the original profile table.
- (4) **clear**: When you type **clear**, the program needs to clear the table of the profiles.

Description:

This game is similar to a lottery game. First, we input the total prize, then the winning numbers of the game (at least 3 numbers and at most 9 numbers). The numbers are in the range 1~109. Then you need to type **assign** to switch to the assign mode and input the names, tickets, and the e-mail addresses of people who buy the tickets. Save the information in records.

Record format:

<Name>	<Tickets>	<E-mail>
Alice	1 55 61	Alice11@citizen.com
Bob	69 11 100	bob11@foreigner.com

Table output format:

Name	Tickets	Id	Citizenship	Prize
Alice	1 55 61	Alice11	Citizen	1000
Bob	69 11 100	bob11	Foreigner	800

The attribute **Tickets** contain the numbers he/she bets on.

The attribute **ID** is extracted from **E-mail** (before '@').

The attribute **Citizenship** is learned from **E-mail**, after '@' and before '.com'. It has two types, citizen or foreigner.

And **Prize** is calculated according to the rules below:

1. The prize money for each number is the total prize divided by the number of betting numbers.

(Prize per number = Total Prize/Number of betting numbers)

2. One matched number for one prize.
3. If the person is a citizen, he/she will get the full prize he/she won.
If the person is a foreigner, he/she will get 80% of the prize he/she won.

Notes:

1. You need to deal with the space between the inputs.
2. Everyone can buy only one tickets, which means the Name in the profile table cannot be repeated!
3. The program must can assign new profile at any moment after input total prize and winning numbers.

Example1:

```
Enter Total Prize:15000
Enter the winning numbers:1 2 84 12 36
assign
Alice 1 55 87 28 109 Alicell@citizen.com
Bob 2 66 84 10 36 Bob22@citizen.com
Dandy 6 8 12 106 55 Dandy44@foreigner.com
Cat 2 66 84 10 36 Cat33@foreigner.com

show
Name      Tickets      Id      Civil      Prize
Alice     1      55      87      28      109      Alicell      citizen      3000.000000
Bob       2      66      84      10      36      Bob22       citizen      9000.000000
Dandy     6      8      12      106     55      Dandy44     foreigner    2400.000000
Cat       2      66      84      10      36      Cat33       foreigner    7200.000000

sort prize
Name      Tickets      Id      Civil      Prize
Bob       2      66      84      10      36      Bob22       citizen      9000.000000
Cat       2      66      84      10      36      Cat33       foreigner    7200.000000
Alice     1      55      87      28      109      Alicell     citizen      3000.000000
Dandy     6      8      12      106     55      Dandy44     foreigner    2400.000000

sort name
Name      Tickets      Id      Civil      Prize
Alice     1      55      87      28      109      Alicell     citizen      3000.000000
Bob       2      66      84      10      36      Bob22       citizen      9000.000000
Cat       2      66      84      10      36      Cat33       foreigner    7200.000000
Dandy     6      8      12      106     55      Dandy44     foreigner    2400.000000
```

```
clear
```

```
show
Name      Tickets      Id      Civil      Prize
```

Example2:

```
Enter Total Prize:15000
Enter the winning numbers:1 2 84 12 36
assign
Alice 1 55 87 28 109 Alicell@citizen.com
Bob 2 66 84 10 36 Bob22@citizen.com
Alice 1 55 87 28 109 Alicell@citizen.com
The name was already in the list
Dandy 6 8 12 106 55 Dandy44@foreigner.com
Alice 1 55 87 28 109 Alicell@citizen.com
The name was already in the list
Bob 2 66 84 10 36 Bob22@citizen.com
The name was already in the list
Cat 2 66 84 10 36 Cat33@foreigner.com

show
Name      Tickets      Id      Civil      Prize
Alice     1      55      87      28      109      Alicell    citizen    3000.000000
Bob       2      66      84      10      36      Bob22     citizen    9000.000000
Dandy    6       8      12     106     55      Dandy44   foreigner  2400.000000
Cat      2      66      84      10      36      Cat33     foreigner  7200.000000
```

Exmpale3:

```
Enter Total Prize:-2
the wining prize need to be not less than 0
Enter Total Prize:1000
Enter the winning numbers:2
the number of the balls is wrong
Enter Total Prize:1000
Enter the winning numbers:1 2 3 4 5 6 7 8 9 10
the number of the balls is wrong
```

4. Longest natural number counting sequence

Write a program that works as a counter that meets the following criteria:

1. For any given counter, its maximum and minimum values are specified, e.g. max=5, min=1, as natural numbers, i.e. positive integers. A starting number is also specified.
2. Any counter is a both-way counter, i.e. it can count up or down, and it can only increase or decrease by 1.
3. When the counter reaches the maximum, the next number can be either reset to the minimum, or the number prior to the maximum.

By contrast, when it reaches the minimum, the next number can be either the next number larger than the minimum, or reset to the maximum.

E.g. Min=1, Max=4, Start=2

The following counting sequences are legal:

(a) 2,3,4,3,2,1

(b) 2,3,4,1,2,3

(c) 2,3,4,3,2,3,2,1,4,1,2,1,2,3

(d) 2,1,4,3,2,1,2,3,4,1,2,3,2,1

Your job is to create the longest legal counting sequence from a series of given numbers.

Input: Max, Min, Start (Max, Min, Start ≥ 1 , min \leq max) A series of numbers terminated with 0
Output: The longest legal counting sequence and its length . (If there are more than one longest sequence, You need to print all of them in any order.)

Examples:

Max = 5, Min = 2, Start = 2

Numbers available are **3 4 4 4 5 0**

The longest sequence should be **2 3 4 5 4** and **2 5 4 3 4** (You have to print both of them)

Because:

Next number of 2 is 3	Next number of 3 is 4	Next number of 4 is 5	Previous number of 5 is 4
2	3	4	5

Previous number of 2 is 5	Previous number of 5 is 4	Previous number of 4 is 3	Next number of 3 is 4
2	5	4	3

```
Please input max number: 20
Please input min number: 5
Please input start number: 6
Please input available numbers: 20 7 5 0
There are 1 longest Sequences(length 3):
6 5 20
```

```
Please input max number: 15
Please input min number: 1
Please input start number: 10
Please input available numbers: 1 12 2 6 11 1 3 6 8 14 9 0
There are 2 longest Sequences(length 3):
10 11 12
10 9 8
```