Exercise 5

布置时间：2009.12.3

Elearning 上提交时间：2009.12.13

1. （**字符串操作**）编写一个函数来比较两个 C 字符串的相等性。如果字符串相等，函数返回 true；否则返回 false。编写另一个函数来对两个 C 字符串进行 Ignore Case 比较。如果字符串大小写无关情况下相等（如”abc”和”ABC”相等），函数返回 true；否则返回 false。书写程序测试这两个函数。

2. （**main 函数**）学习命令行执行程序。

编译以前作业中的程序，找到其可执行文件（如 question_1.exe，可以根据程序名自行确定）所在的目录，如” E: \程序设计\作业 5\debug”。 根据下面方法执行程序：

打开开始菜单
➔点击’运行’
➔输入’cmd’
➔输入’E:’
➔输入’cd E: \程序设计\作业 5\debug’
➔输入 question_1 执行程序
➔检查运行结果。

完成下面程序，并通过命令行执行程序（如 question_1，可以根据程序名自行确定）。通过命令行执行程序时需要列出带有 3-4 个命令行选项的情况（如输入“question_1 –a –b –c”）。

```
//本程序需要打印出除了程序名外的每一个参数
int main （int argc, char *argv[] ) {
        …
}
```

3. （**main 函数**）通过命令行执行程序（如 question_1，可以根据题目自行确定）。如果输入“question_1 -max”，则用户需要输入一串整数，输出结果是这一串数字中的最大值；如果输入“question_1 -min”，则用户需要输入一串数字，输出结果是这一串数字中的最小值；如果输入“question_1 -avg”，则用户需要输入一串数字，输出结果是这一串数字中的平均值（整数即可）；如果不是上述输入，则需要报错。

4. （**main** 函数）上面程序不能处理”question_1 – max”等输入。请修改程序，使之能够处理“-”和后面选项之间的多余的空格。注意：question_1 –m ax 与 question_1 –max 的效果不应该相同；前者为输入错误，需要报错。

5. （**main** 函数）请修改上面的程序，使之能够处理多个命令行选项。如 question_1 –max –min 则可以打印出最大值和最小值（即用户需要输入一串数字，按照选项顺序分别打印出最大值、最小值、平均值等）。注意：本题计算方法可能较为复杂。

6. **（函数，Absolute C++中文版 115 页编程练习 1）**Write a program that converts from 24-hour notation to 12-hour notation. For example, it should convert 14:25 to 2:25 P.M. The input is given as two integers. There should be at least three functions: one for input, one to do the conversion, and one for output. Record the A.M./P.M. information as a value of type `char`, ’A’ for A.M. and ’P’ for P.M. Thus, the function for doing the conversions will have a call-by-reference formal parameter of type `char` to record whether it is A.M. or P.M. (The function will have other parameters as well.) Include a loop that lets the user repeat this computation for new input values again and again until the user says he or she wants to end the program.

7. **（函数，Absolute C++中文版 115 页编程练习 2）**The area of an arbitrary triangle can be computed using the formula

$$area = \sqrt{s(s-a)(s-b)(s-c)}$$

where a, b, and c are the lengths of the sides, and s is the semiperimeter $s=(a+b+c)/2$. Write a `void` function that uses five parameters: three value parameters that provide the lengths of the edges, and two reference parameters that compute the area and perimeter (*not the semiperimeter*). Make your function robust. Note that not all combinations of a, b, and c produce a triangle. Your function should produce correct results for legal data and reasonable results for illegal combinations.

8. **（函数，Absolute C++中文版 115 页编程练习 3）**Write a program that tells what coins to give out for any amount of change from 1 cent to 99 cents. For example, if the amount is 86 cents, the output would be something like the following:

```
86 cents can be given as 3 quarter(s) 1 dime(s) and 1 penny(pennies)
```

Use coin denominations of 25 cents (quarters), 10 cents (dimes), and 1 cent (pennies). Do not use nickel and half-dollar coins. Your program will use the following function (among others):

```
void computeCoin(int coinValue, int& number, int& amountLeft);
//Precondition: 0 < coinValue < 100; 0 <= amountLeft < 100.
//Postcondition: number has been set equal to the maximum number
//of coins of denomination coinValue cents that can be obtained
//from amountLeft cents. amountLeft has been decreased by the
//value of the coins, that is, decreased by number*coinValue.
```

For example, suppose the value of the variable amountLeft is 86. Then, after the following call, the value of number will be 3 and the value of amountLeft will be 11 (because if you take three quarters from 86 cents, that leaves 11 cents):

```
computeCoins(25, number, amountLeft);
```

Include a loop that lets the user repeat this computation for new input values until the user says he or she wants to end the program. (*Hint:* Use integer division and the % operator to implement this function.)

9. （数组，**Absolute C++**中文版 **150** 页编程练习 **1**）Write a program that reads in the average monthly rainfall for a city for each month of the year and then reads in the actual monthly rainfall for each of the previous 12 months. The program then prints out a nicely formatted table showing the rainfall for each of the previous 12 months as well as how much above or below average the rainfall was for each month. The average monthly rainfall is given for the months January, February, and so forth, in order. To obtain the actual rainfall for the previous 12 months, the program first asks what the current month is and then asks for the rainfall figures for the previous 12 months. The output should correctly label the months.

   There are a variety of ways to deal with the month names. One straightforward method is to code the months as integers and then do a conversion before doing the output. A large switch statement is acceptable in an output function. The month input can be handled in any manner you wish, as long as it is relatively easy and pleasant for the user.

   After you have completed the above program, produce an enhanced version that also outputs a graph showing the average rainfall and the actual rainfall for each of the previous 12 months. The graph should be similar to the one shown in Display 5.4, except that there should be two bar graphs for each month and they should be labeled as the average rainfall and the rainfall for the most recent month. Your program should ask the user whether she or he wants to see the table or the bar graph, and then should display whichever format is requested. Include a loop that allows the user to see either format as often as the user wishes until the user requests that the program end.

10. （数组，**Absolute C++**中文版 **150** 页编程练习 **2**）Write a function called deleteRepeats that has a partially filled array of characters as a formal parameter and that deletes all repeated letters from the array. Since a partially filled array requires two arguments, the function will actually have two formal parameters: an array parameter and a formal parameter of type int that gives the number of array positions used. When a letter is deleted, the remaining letters are moved forward to fill in the gap. This will create empty positions at the end of the array so that less of the array is used. Since the formal parameter is a partially filled array, a second formal parameter of type int will tell how many array positions are filled. This second formal parameter will be a call-byreference parameter and will be changed to show how much of the array is used after the repeated letters are deleted. For example, consider the following code:

    - char a[10];
    - a[0] = ’a’;
    - a[1] = ’b’;
    - a[2] = ’a’;
    - a[3] = ’c’;
    - int size = 4;
    - deleteRepeats(a, size);

After this code is executed, the value of a[0] is ’a’, the value of a[1] is ’b’, the value of a[2] is ’c’, and the value of size is 3. (The value of a[3] is no longer of any concern, since the partially filled array no longer uses this indexed variable.) You may assume that the partially filled array contains only lowercase letters. Embed your function in a suitable test program.

11. （数组，**Absolute C++中文版 150 页编程练习 3**）The standard deviation of a list of numbers is a measure of how much the numbers deviate from the average. If the standard deviation is small, the numbers are clustered close to the average. If the standard deviation is large, the numbers are scattered far from the average. The standard deviation, S, of a list of N numbers xi is defined as follows:

$$S = \sqrt{\frac{\sum_{i=1}^{N}(xi - \bar{x})^2}{N}}$$

Where $\bar{x}$ is the average of the numbers x1, x2, . . . Define a function that takes a partially filled array of numbers as its argument and returns the standard deviation of the numbers in the partially filled array. Since a partially filled array requires two arguments, the function will actually have two formal parameters: an array parameter and a formal parameter of type int that gives the number of array positions used. The numbers in the array will be of type double. Embed your function in a suitable test program.

12. （数组，**Absolute C++中文版 150 页编程练习 4**）Write a program that reads in an array of type int. You may assume that there are fewer than 50 entries in the array. Your program determines how many entries are used. The output is to be a two-column list. The first column is a list of the distinct array elements; the second column is the count of the number of occurrences of each element. The list should be sorted on entries in the first column, largest to smallest. For the array values: (-12 3 -12 4 1 1 -12 1 -1 1 2 3 4 2 3 -12), the output should be

- N Count
- 4 2
- 3 3
- 2 2
- 1 4
- -1 1
- -12 4

13. （数组，**Absolute C++中文版 150 页编程练习 5**）An array can be used to store large integers one digit at a time. For example, the integer 1234 could be stored in the array a by setting a[0] to 1, a[1] to 2, a[2] to 3, and a[3] to 4. However, for this exercise you might find it more useful to store the digits backward, that is, place 4 in a[0], 3 in a[1], 2 in a[2], and 1 in a[3]. In this exercise you will write a program that reads in two positive

integers that are 20 or fewer digits in length and then outputs the sum of the two numbers. Your program will read the digits as values of type char so that the number 1234 is read as the four characters ’1’, ’2’, ’3’, and ’4’. After they are read into the program, the characters are changed to values of type int. The digits will be read into a partially filled array, and you might find it useful to reverse the order of the elements in the array after the array is filled with data from the keyboard. (Whether or not you reverse the order of the elements in the array is up to you. It can be done either way, and each way has its advantages and disadvantages.) Your program will perform the addition by implementing the usual paper-and-pencil addition algorithm. The result of the addition is stored in an array of size 20 and the result is then written to the screen. If the result of the addition is an integer with more than the maximum number of digits (that is, more than 20 digits), then your program should issue a message saying that it has encountered “integer overflow.” You should be able to change the maximum length of the integers by changing only one globally defined constant. Include a loop that allows the user to continue to do more additions until the user says the program should end.

14. （**域和生命期**）分析改错题

1. 在下面的代码例子中，请指出不同的域。ix 的哪些声明是错误的？为什么？

```
int ix = 1024;
int ix();
void func( int ix, int iy ) {
    int ix = 255;
    if ( int ix = 0 ) {
        int ix = 79;
        {
        int ix = 89;
        }
    }
    else {
            int ix = 99;
    }
}
```

2. 在下面的代码例子中，用到ix 和iy 的地方分别引用了哪些声明？

```
int ix = 1024;

void func( int ix, int iy ) {
    ix = 100;
    for( int iy = 0 ; iy < 400; iy += 100 ) {
            iy += 100;
        ix = 300;
    }
    iy = 400;
}
```

3. 指出下列语句哪些是声明，哪些是定义，为什么？

(a) extern int ix = 1024;

(b) int iy;

(c) extern void reset(void *p) { /* ... */ }

(d) extern const int *pi;

(e) void print(const matrix &);

4. 在下列声明和定义中，哪些应被放到头文件中，哪些应被放到程序文本文件中，为什么？

(a) int var;

(b) inline bool is_equal(const SmallInt &, const SmallInt &) { }

(c) void putValues(int *arr, int size);

(d) const double pi = 3.1416;

(e) extern int total = 255;