

UEE1302(1066) F12: Introduction to Computers and Programming

Lab 9: Array Standard Class String



What you will learn from Lab 9

In this laboratory, you will understand how to use C++ *class strings* and learn the usage of file I/O.

TASK 9-1 : TWO-DIMENSIONAL ARRAY

- ✓ Please predict the result of program lab9-1, and then execute this program to compare the result on screen with what you predict.

```
//File: lab9-1.cpp
#include <iostream>
#include <iomanip>
using namespace std;

const int numrows = 3;
const int numcols = 4;
void showarray(int [][]);
int main()
{
    int val[numrows][numcols]={ {1, 2, 3, 4},
                                  {5, 6, 7, 8},
                                  {9, 10, 11,12} };
    cout << "Display the multiplied elements";
    showarray(val);
    return 0;
}
void showarray(int array[][]){
    for(int row=0; row<numrows; row++){
        for(int col=0; col<numcols; col++){
            cout << setw(5)<< array[row][col] << " ";
        }
        cout << endl;
    }
}
```

Hint: change void showarray(int [][]) to void showarray(int [][numcols])

TASK 9-2 : C-STRING

- ✓ Please predict the result of program lab8-6, and then execute this program to compare the result on screen with what you predict.

```
//File: lab9-2.cpp
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
    char stringA[20] = {'C','+', '+', '\0'};
    char stringB[20] = "Programming";

    cout << "length of " << stringA << " is " << strlen(stringA) << endl;
    cout << "length of " << stringB << " is " << strlen(stringB) << endl;

    char stringC[20];

    // Copy stringA to stringC
    strcpy(stringC, stringA);
    cout << stringC << endl;

    // Concatenate " " and stringB to stringC
    strcat(stringC, " ");
    strcat(stringC, stringB);
    cout << stringC << endl;

    // Clear StringC
    strcpy(stringC, "");

    //compare characters of two strings
    cout << "Enter a new word :";
    cin.getline (stringC, 20);
    if(strcmp(stringC, stringB)!=0)
        cout << "the two strings are same!" << endl;
    else
        cout << "the two strings are different @" << endl;

    return 0;
}
```

TASK 9-3: CCTYPE

- ✓ Problem lab9-3 introduces the usage of cctype to manipulate c-strings and characters. Note that `#include <cctype>` is required to be included.
- ✓ Problem lab9-3 is used to analyze an input sentence. It computes the number of alphabetical letters, digits and spaces in a sentence and shows the result on screen. Moreover, the program for such problem changes letters to the upper case and displays the new sentence. You can try to input different kinds of sentences to test the problem. The expected result for the program is shown as follows.

```
> ./lab9-3
Enter a sentence:
Hsinchu is in 1st place in city ranking
New sentence is:
HSINCHU IS IN 1ST PLACE IN CITY RANKING
Number of alphas is 31
Number of digits is 1
Number of spaces is 7
```

- ✓ However, the following example is the execution of lab9-3 and different from the expected.

```
> ./lab9-3
Enter a sentence:
Hsinchu is in 1st place in city ranking
New sentence is:
HSINCHU
Number of alphas is 9
Number of digits is 1
Number of spaces is 0
```

- ✓ Please try to find out the errors and fix them in the given C/C++ program lab9-3

```
//File: lab9-3.cpp
#include <iostream>
#include <cstring>
#include <cctype>
using namespace std;
int main()
{
    char str[80];
    cout << "Enter a sentence: " << endl;
    cin >> str;

    int numalpha = 0;
    int numspace = 0;
    int numdigit = 0;
```

```
int len = 80;

for (int idx=0;idx<len;idx++)
{
    char c = str[idx];
    if (isalpha(c))
    {
        str[idx] = toupper(c);
        ++numalpha;
    }
    else if (isdigit(c))
    {
        ++numdigit;
    }
    else if (isspace(c))
    {
        ++numspace;
    }
}

cout << "New sentence is: " << endl;
cout << str << endl;
cout << "Number of alphas is " << numalpha << endl;
cout << "Number of digits is " << numdigit << endl;
cout << "Number of spaces is " << numspace << endl;

return 0;
```

Hint: `cin >> str` \Rightarrow `cin.getline(str,80)`

TASK 9-4 : CLASS STRING

- ✓ Problem lab9-4 introduces the usage of class `string`. `#include <string>` is required to be included.
- ✓ Problem lab9-4 asks the user's name and shows the welcome information. The expected result of the program lab9-4 is shown as follows.

```
> ./lab9-4
Enter your name: Ben
Hello, Ben!!
Welcome to my house!!
```

- ✓ However, the programmer misses something and shows a different result as follows.

```
> ./lab9-4
Enter your name: Ben
Hello, Ben!!Welcome to my house!!
```

- ✓ Please fix the error(s) to show the correct results. Note that you need to concatenate all welcome words into one string.

```
//File: lab9-4.cpp
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string heading = "Hello";
    string ending("Welcome to my house!!");
    string name;

    cout << "Enter your name: ";
    cin >> name;

    string sentence = heading + ", " + name + "!!" + ending;
    cout << sentence << endl;

    return 0;
}
```

TASK9-5 : CLASS STRING – MEMBER FUNCTION

- ✓ In this task, some member functions built in the <string> library are introduced. Note that you need to use dot operator (.) to access the member functions of the string variable.
- ✓ Problem lab9-5 asks user to enter a word and tries to find out if there is a sub-string "nctu" in the user-input string.

```
//File: lab9-5.cpp
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string str;
    cout << "Enter a sentence:" << endl;
    getline(cin, str);
}
```

```
int pos = str.find("nctu");
if (pos == string::npos)
{
    cout << "nctu is not found !!" << endl;
}
else
{
    cout << "nctu is found at pos: " << pos << endl;
}
cout << "Substring from str[2] to str[4] is " << str.substr(2,3) << endl;
return 0;
}
```

- Please try different strings to test the program.

```
> ./lab9-5
Enter a sentence:
I am a student in nctu
nctu is found as pos: 18
Substring from str[2] to str[4] is am
> ./lab9-5
Enter a sentence:
studentinnctu
nctu is found as pos: 9
Substring from str[2] to str[4] is ude
> ./lab9-5
Enter a sentence:
nctustudent
nctu is found as pos: 0
Substring from str[2] to str[4] is tus
> ./lab9-5
Enter a sentence:
nthu
nctu is not found !!
Substring from str[2] to str[4] is hu
```

- `str.find(str1)` returns index of the first occurrence of `str1` in `str`
- `string::npos`. `npos` indicates the end of the string. `npos = -1`.
- `str.substr(pos,length)` returns the sub-string of `str` from index `pos` to index `pos+length`

TASK 9-6: EXERCISES

- ✓ Please write a program to ask user to enter a sentence, count the number of words in this sentence and display words one-by-one on screen. Note that each word can be only composed of [0~9] and [a~z]. The words are separated by ' '(space). For implementing the program for this problem, you can use the member function `find(str1,pos)`, `find_first_not_of(str1,pos)`, `substr(pos1, len)` and `length()` of class string.

```
> ./ex9-1
Enter a sentence:
I am a student from NCTU
There are 6 words in your sentence.
They are "I", "am", "a", "student", "from" and "NCTU"
> ./ex9-1
Enter a sentence:
I like the band 1976
There are 5 words in your sentence.
They are "I", "like", "the", "band" and "1976"
>
```

Hint: You can use two for loops to finish the job. The first loop is to identify the number of spaces in the sentence. The second loop is to print each word. Note that there is an "and" between the last two words.

Hint:

1. `str.find(str1,pos)` returns index of the first occurrence of `str1` in `str` starting from `pos`
2. `str.find_first_not_of(str1,pos)` returns index of the first occurrence not of `str1` in `str` starting from `pos`
3. `str.substr(pos,len)` returns the sub-string of `str` from index `pos` to index `pos+len`
4. `str.length()` returns the length of `str`

- ✓ Please complete a program which can generate `matrix1` and `matrix2` randomly with specific number of rows and columns. Note that the number of rows and columns are less than 10 and the range of the elements in matrix is (0, 10]. The `matrix3` is the result which is `matrix1` product `matrix2` composed by sorting the elements for each row.

When multiplying matrices, the elements of the rows in the first matrix are multiplied with corresponding columns in the second matrix. One may compute each entry in the third matrix one at a time.

$$\begin{aligned}
 A &= \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1m} \\ A_{21} & A_{22} & \cdots & A_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nm} \end{pmatrix} & B &= \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1p} \\ B_{21} & B_{22} & \cdots & B_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ B_{m1} & B_{m2} & \cdots & B_{mp} \end{pmatrix} \\
 AB &= \begin{pmatrix} AB_{11} & AB_{12} & \cdots & AB_{1p} \\ AB_{21} & AB_{22} & \cdots & AB_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ AB_{n1} & AB_{n2} & \cdots & AB_{np} \end{pmatrix} & AB_{ij} &= \sum_{k=1}^m A_{ik} B_{kj}
 \end{aligned}$$

8/10


```
#include <iostream>
#include <iomanip>
#include <time.h>
using namespace std;
const int Maxsize=10;
int numrow,numcol;
void showarray(int ,int ,int [][][Maxsize]);
void random(int [][][Maxsize],int [][][Maxsize]);
void mul (int [][][Maxsize],int [][][Maxsize],int [][][Maxsize]);
int main()
{
    srand(time(NULL));
    cout << "Please enter rows and cols" << endl;
    cin>>numrow>>numcol;
    int matrix1[Maxsize][Maxsize],matrix2[Maxsize][Maxsize];
    random(matrix1,matrix2);
    cout << "Matrix1" << endl;
    showarray(numrow,numcol,matrix1);
    cout << "Matrix2" << endl;
    showarray(numcol,numrow,matrix2);
    int matrix3[Maxsize][Maxsize]={0};
    mul(matrix1,matrix2,matrix3);
    cout << "Matrix3" << endl;
    showarray(numrow,numrow,matrix3);
}
```

Exercise after LAB

1. Lottery

Please write a program that simulates a lottery. The program should have an array of 5 integers named `winningDigits`, with a randomly generated number in the range of 0 through 9 for each element in the array. The program should ask the user to enter 5 digits and should store them in a second integer array named `player`. The program must compare the corresponding elements in the two arrays and count how many digits match. For example, the following shows the `winningDigits` array and the `Player` array with sample numbers stored in each. There are two matching digits, elements 2 and 4.

winningDigits	7	4	9	1	3
Player	4	2	9	7	3

Once the user has entered a set of numbers, the program should display the winning digits and the player's digits and tell how many digits matched.

2. Digit sums of Squares and Cubes

If you add up all the digits in 468, you get $4+6+8 = 18$. The square and cube of 468 are 219024 and 102503232, respectively. Interestingly, if you add up the digits of the square or cube, you get 18 again. Are there other integers that share this property? Write a program that lists all positive integers k less than 1000 such that the three numbers k , k^2 , and k^3 have digits that add up to the same number.