

Lab #03

Date: 2014/03/05 – Michael Hsu

goo.gl/M5TDGL

M5TDGL

About Homework 1

About Homework 2

PD14-02b BONUS

Array Declaration

```
int iarr[5] = {0, 1, 2, 3, 4};  
double darr[5] = {0.0, 0.1, 0.2, 0.3, 0.4};  
char carr[5] = {'A', 'B', 'C', 'D', 'E'};  
bool barr[5] = {false, true, false, true, false};
```

Array Initialization

```
int iarr[100] = {0};  
double darr[100] = {0.0};  
bool barr[100] = {false};
```

```
int iarr[5] = {0, 1}; // {0, 1, 0, 0, 0}
```

```
int iarr[3] = {1, 2, 3};  
int iarr[] = {1, 2, 3};  
// Equal
```

Array Access

```
#include <iostream>
using namespace std;

int main()
{
    const int LENGTH = 10; // constant to store array length
    int iarr[LENGTH] = {0}; // all elements is 0

    for(int i = 0; i < LENGTH; i++)
        iarr[i] = i;

    for(int i = 0; i < LENGTH; i++)
        cout << iarr[i] << " ";

    return 0;
}
```

0 1 2 3 4 5 6 7 8 9

Copy Arrays

```
#include <iostream>
using namespace std;
```

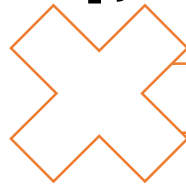
```
int main()
{
```

```
    const int LENGTH = 5;
```

```
    int arr1[LENGTH] = {1, 2, 3, 4, 5};
```

```
    int arr2[LENGTH];
```

```
    arr2 = arr1;
```



you cannot write array1 = array2

```
    for(int i = 0; i < LENGTH; i++)
        arr2[i] = arr1[i];
```

```
    return 0;
```

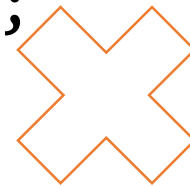
```
}
```

Two Dimension Array

We may initialize a two-dimensional array as follows:

```
int score[2][3] = {{1, 2, 3}, {4, 5, 6}};  
int score[2][3] = {1, 2, 3, 4, 5, 6};  
int score[][3] = {1, 2, 3, 4, 5, 6};  
// they are equal
```

```
int score[2][] = {1, 2, 3, 4, 5, 6};  
int score[][] = {1, 2, 3, 4, 5, 6};  
// error
```



Two Dimension Array Access

```
#include <iostream>
using namespace std;
```

```
int main()
{
```

```
    int score[2][3] = {1, 2, 3, 4, 5, 6};
```

```
    for (int i = 0; i < 2; i++)
```

```
    {
```

```
        for (int j = 0; j < 3; j++)
```

```
            cout << score[i][j] << " ";
```

```
        cout << endl;
```

```
    }
```

```
    return 0;
```

```
}
```

score[0]	score[0][0] 1	score[0][1] 2	score[0][2] 3
score[1]	score[1][0] 4	score[1][1] 5	score[1][2] 6

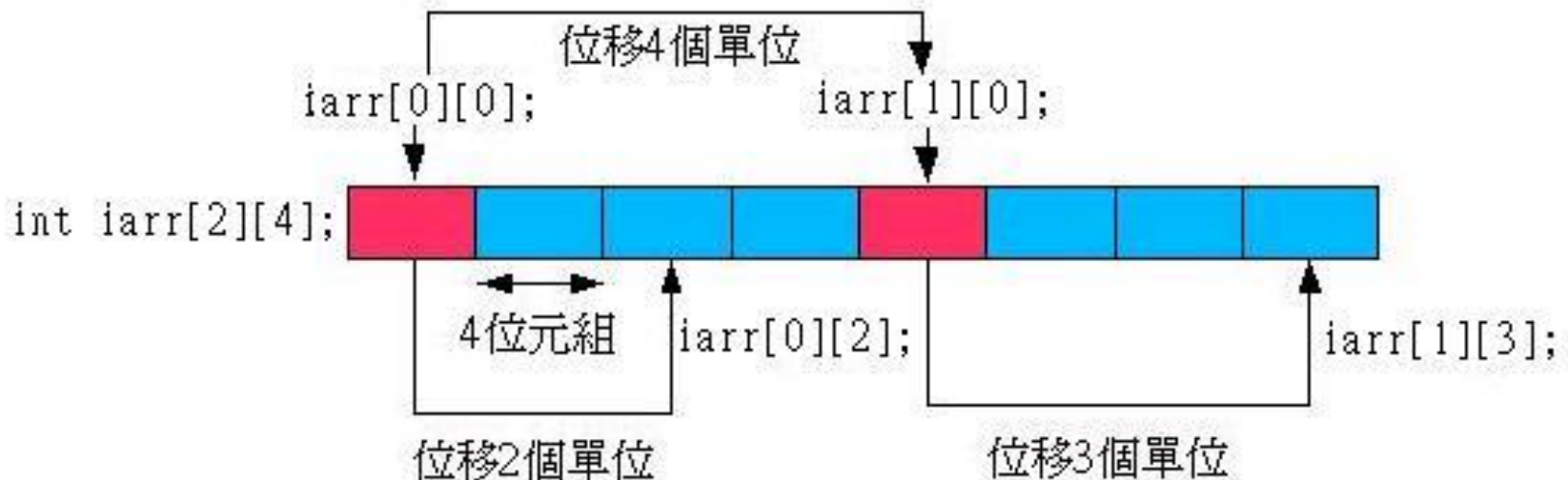
```
1 2 3
```

```
4 5 6
```

Two Dimension Array

Two-dimensional arrays are not actually rows and columns. A two-dimensional array is actually several one-dimensional arrays.

score[0]	score[0][0] 1	score[0][1] 2	score[0][2] 3
score[1]	score[1][0] 4	score[1][1] 5	score[1][2] 6



Lab Work: LAB14-03a Display Polynomial (1/2)

Description

Write a program that repeatedly reads the coefficient-exponent pairs, arranged in descending order by exponent and display the polynomial.

Input

The input contains several lines of integer values. In each line, there are an even number of values. Values are separated by white spaces. Every two numbers will be a pair of coefficient-exponent component. The first value of each line will be number n of pairs. The first value $c_i \in \mathbb{Z} \cap [-1000, 1000]$ in the component indicates the coefficient. The second value $e_i \in \mathbb{Z} \cap [0, 1000]$ in the component indicates the exponent. And the number of coefficient-exponent components $i \in \mathbb{Z} \cap (0, 1000]$. Then each line will form a polynomial.

For example, a line of testing data contains

$n \ c_1 \ e_1 \ c_2 \ e_2 \ c_3 \ e_3 \ c_4 \ e_4$

Lab Work: LAB14-03a Display Polynomial (2/2)

Output

You have to output the polynomial in descending order. If the coefficient or exponent value is 1, do not print for it.

Sample input

1 15 6

3 6 0 2 3 -1 1

3 -6 3 1 4 7 1

Sample output

15x⁶

2x³-x+6

x⁴-6x³+7x

C strings as character arrays

```
#include <iostream>
using namespace std;

int main()
{
    char array[10];

    cout << "Enter something: ";
    cin >> array;
    cout << array << endl;

    if(array[4]== 'e')
        cout << "get e !";
    return 0;
}
```

```
Enter something: abcde
abcde
get e !
```

Convert string to integer, atoi()

```
#include <iostream>
#include <stdlib.h> /* atoi */
using namespace std;

int main()
{
    char array[10]= "5565";

    cout << atoi(array)+1;

    return 0;
}
```

5566

C strings as character arrays

You can find more information about string from following website:

<http://www.cplusplus.com/reference/cstdlib/atoi/>

Lab Work: LAB14-03b Polynomial Calculator (1/2)

Description

Write a program that repeatedly reads coefficient-exponent pairs, calculates two polynomial and display the final result of polynomial.

Input

Almost the same inputs of 'LAB14-03a'. The input contains several lines of integer values. In each line, we add first polynomial to second polynomial. The first value of each polynomial will be number n of pairs. Values are separated by white spaces.

For example, a line of testing data contains

$$n \ c_1 \ e_1 \ c_2 \ e_2 \ c_3 \ e_3 \ c_4 \ e_4 + n \ c_5 \ e_5 \ c_6 \ e_6$$

Output

You have to output three polynomial in descending order. Each polynomial are separated by one white spaces. If the coefficient or exponent value is 1, do not print for it.

Sample input

1 15 6 + 1 5 6

3 2 3 -1 1 6 0 + 3 1 4 -6 3 7 1

Sample output

15x⁶ 5x⁶ 20x⁶

2x³-x+6 x⁴-6x³+7x x⁴-4x³+6x+6

Q&A