

LAB # 11 Solution
(Relevant Lecture: #22-24)

1. Write a program that reads 8 integers into a 1-D array from the keyboard. Then print out the array elements. Then **swap** the first element and the seventh element (Refer to the example on Slide 6 in Lecture#22). Finally print out the array elements after the swapping.

Testing: if you input 1 3 5 7 9 2 4 6 from the keyboard, then the output on the screen should be:

The array elements before swapping: 1 3 5 7 9 2 4 6
The array elements after swapping: 4 3 5 7 9 2 1 6

Note: the output in the test cases is given as an example. You may use a different format, for example, each number may be output on a different line.

Solution:

```
#include <stdio.h>
#define array_size 8

void main(void)
{

    int num_array[array_size];
    int temp = 0;
    int i;

    for (i = 0; i < array_size; i++)
    {
        printf("please input an array element:\n");
        scanf_s("%d", &num_array[i]);
    }

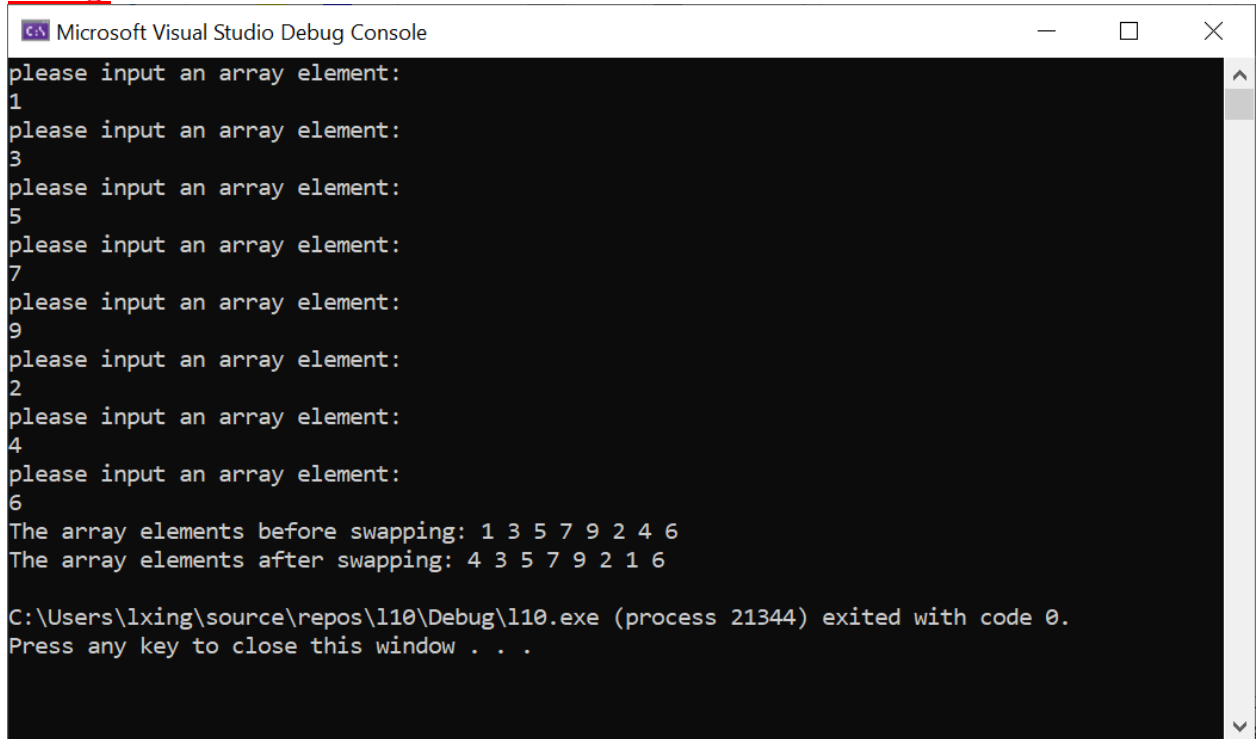
    printf("The array elements before swapping: ");
    for (i = 0; i < array_size; i++)
    {
        printf("%d ", num_array[i]);
    }
    printf("\n");

    /* swap the first element and the seventh element */
    temp = num_array[0];
    num_array[0] = num_array[6];
    num_array[6] = temp;

    printf("The array elements after swapping: ");
    for (i = 0; i < array_size; i++)
    {
        printf("%d ", num_array[i]);
    }
}
```

```
        printf("\n");  
    }  
}
```

Testing:



```
Microsoft Visual Studio Debug Console  
please input an array element:  
1  
please input an array element:  
3  
please input an array element:  
5  
please input an array element:  
7  
please input an array element:  
9  
please input an array element:  
2  
please input an array element:  
4  
please input an array element:  
6  
The array elements before swapping: 1 3 5 7 9 2 4 6  
The array elements after swapping: 4 3 5 7 9 2 1 6  
  
C:\Users\lxing\source\repos\l10\Debug\l10.exe (process 21344) exited with code 0.  
Press any key to close this window . . .
```

2. Modify the program in Exercise (4) on Slide 23 of Lecture #22 (refer to the corresponding solution file for the complete program) so that the program reads a 2-D array with 4 rows and 3 columns from the keyboard. Print out all the four elements of **the second column**, and print out the one element on **the third row and the third column** (one number per line).

Testing: if you input 1 2 3 4 5 6 7 8 9 10 11 12 from the keyboard, which corresponds to the 2-D array with the following 4 rows and 3 columns

1	2	3
4	5	6
7	8	9
10	11	12

Then the output on the screen should be:

```
The elements of the second column are:  
2  
5  
8  
11  
The element on the third row and the third column is:  
9
```

Solution:

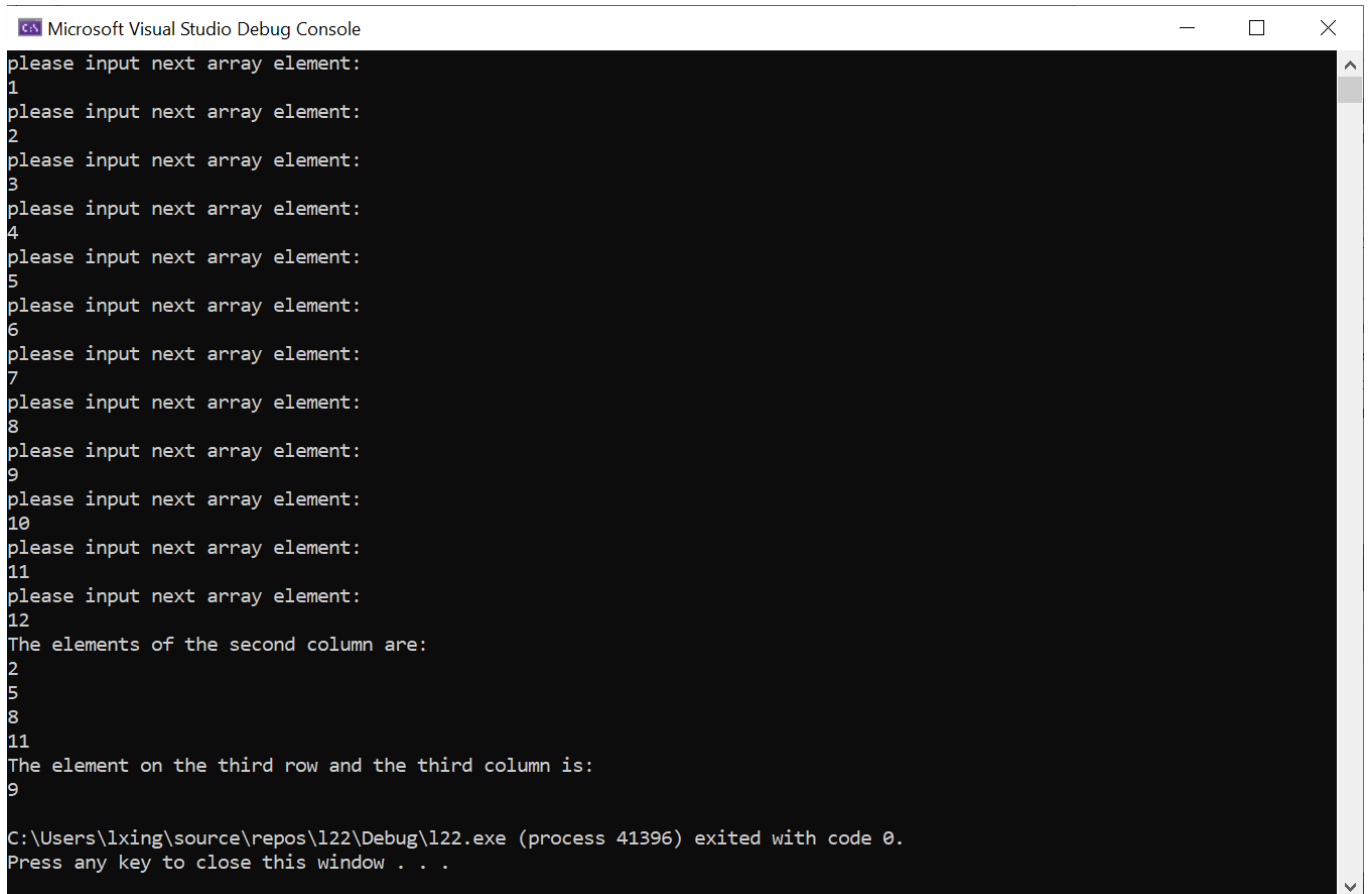
```
#include <stdio.h>

void main(void)
{
    int num_array[4][3];
    int i; /*i is row index*/
    int j; /*j is col index*/

    for (int i = 0; i < 4; i++)
        for (int j = 0; j < 3; j++)
        {
            printf("please input next array element:\n");
            scanf_s("%d", &num_array[i][j]);
        }

    printf("The elements of the second column are:\n");
    for (i = 0; i < 4; i++) /*i is row index*/
        printf("%d\n", num_array[i][1]);

    printf("The element on the third row and the third column is:\n");
    printf("%d\n", num_array[2][2]);
}
```



```
Microsoft Visual Studio Debug Console
please input next array element:
1
please input next array element:
2
please input next array element:
3
please input next array element:
4
please input next array element:
5
please input next array element:
6
please input next array element:
7
please input next array element:
8
please input next array element:
9
please input next array element:
10
please input next array element:
11
please input next array element:
12
The elements of the second column are:
2
5
8
11
The element on the third row and the third column is:
9

C:\Users\lxing\source\repos\l22\Debug\l22.exe (process 41396) exited with code 0.
Press any key to close this window . . .
```

3. Modify the array size to 10 in the **bubble sort** program discussed in *Lecture#24, Slide 15* (Refer to the corresponding solution file for the complete program) so that it can sort the 10 elements of the array into **non-decreasing** order (i.e., from the minimum value to maximum value).

Testing: run it using the 24 34 12 7 3 88 90 7 2 63 the output should be

2 3 7 7 12 24 34 63 88 90

Solution:

```
#include "stdio.h"
#define ARRAY_SIZE 10

void bubbleSort(int list[], int last);

void main(void)
{
    int myarray[ARRAY_SIZE];
    int i = 0;

    printf("Please input the array elements:\n");
    for (i = 0; i < ARRAY_SIZE; i++)
    {
        scanf_s("%d", &myarray[i]);
    }

    bubbleSort(myarray, ARRAY_SIZE - 1);

    printf("The array elements after sorting are:\n");
    for (i = 0; i < ARRAY_SIZE; i++)
    {
        printf("myarray[%d] is: %d\n", i, myarray[i]);
    }
}

void bubbleSort(int list[], int last)
{
    int current, walker, temp;
    for (current = 0; current < last; current++)
        for (walker = last; walker > current; walker--)
            if (list[walker] < list[walker - 1])
            {
                temp = list[walker];
                list[walker] = list[walker - 1];
                list[walker - 1] = temp;
            }
}
```

Testing:

```
Microsoft Visual Studio Debug Console
Please input the array elements:
24 34 12 7 3 88 90 7 2 63
The array elements after sorting are:
myarray[0] is: 2
myarray[1] is: 3
myarray[2] is: 7
myarray[3] is: 7
myarray[4] is: 12
myarray[5] is: 24
myarray[6] is: 34
myarray[7] is: 63
myarray[8] is: 88
myarray[9] is: 90

C:\Users\lxing\source\repos\l26\Debug\l26.exe (process 35360) exited with code 0.
Press any key to close this window . . .
```

4. Modify the **bubble sort** program in Exercise 3 so that it can sort the 10 elements of the array into **non-increasing** order (i.e., from the maximum value to minimum value).

Testing: run it using the 24 34 12 7 3 88 90 7 2 63 the output should be

99 88 63 34 24 12 7 7 3 2

Solution:

```
#include "stdio.h"
#define ARRAY_SIZE 10

void bubbleSort(int list[], int last);

void main(void)
{
    int myarray[ARRAY_SIZE];
    int i = 0;

    printf("Please input the array elements:\n");
    for (i = 0; i < ARRAY_SIZE; i++)
    {
        scanf_s("%d", &myarray[i]);
    }

    bubbleSort(myarray, ARRAY_SIZE - 1);

    printf("The array elements after sorting are:\n");
    for (i = 0; i < ARRAY_SIZE; i++)
    {
        printf("myarray[%d] is: %d\n", i, myarray[i]);
    }
}
```

```

void bubbleSort(int list[], int last)
{
    int current, walker, temp;
    for (current = 0; current < last; current++)
        for (walker = last; walker > current; walker--)
            if (list[walker] > list[walker - 1])
            {
                temp = list[walker];
                list[walker] = list[walker - 1];
                list[walker - 1] = temp;
            }
}

```

Testing:

```

Microsoft Visual Studio Debug Console
Please input the array elements:
24 34 12 7 3 88 90 7 2 63
The array elements after sorting are:
myarray[0] is: 90
myarray[1] is: 88
myarray[2] is: 63
myarray[3] is: 34
myarray[4] is: 24
myarray[5] is: 12
myarray[6] is: 7
myarray[7] is: 7
myarray[8] is: 3
myarray[9] is: 2

C:\Users\lxing\source\repos\l26\Debug\l26.exe (process 33880) exited with code 0.
Press any key to close this window . . .

```

5. Modify the **selection sort** program discussed in *Lecture#24, Slide 12* (Refer to the corresponding solution file for the complete program) so that it can sort the elements of the array into **non-increasing** order (i.e., from the maximum value to minimum value) and the sorted array elements are output to a file named `sortedArray.txt` using `fprintf()` (instead of printing the sorted array elements on the screen).

Testing: run the modified program using the input: 23 34 12 7 3 12 and check the file `sortedArray.txt` on your disk, which should contain

34 23 12 12 7 3

Review your **Lab#10 exercises** and **Lecture#20** about how to write output to a file using `fprintf()`.

Solution:

```

#include "stdio.h"
#define ARRAY_SIZE 6
void selectionSort(int list[], int last);

void main(void)
{
    int myarray[ARRAY_SIZE];
    int i = 0;

```

```

FILE* ptr;

printf("Please input the array elements:\n");
for (i = 0; i < ARRAY_SIZE; i++)
{
    scanf_s("%d", &myarray[i]);
}

selectionSort(myarray, ARRAY_SIZE - 1);

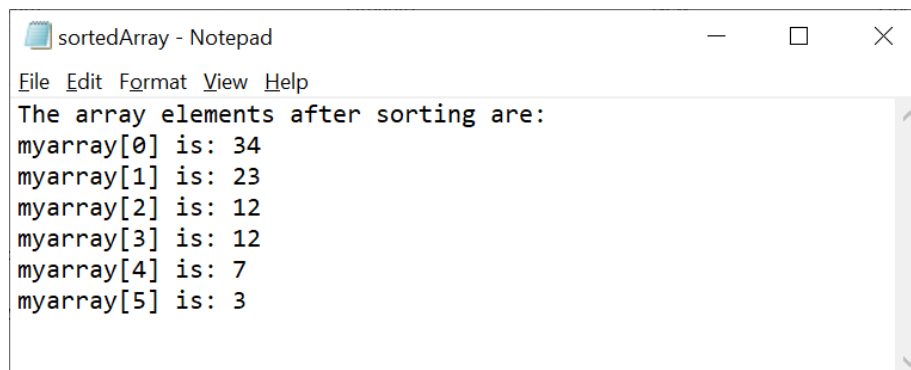
//output the sorted array elements to file
fopen_s(&ptr, "sortedArray.txt", "w");
fprintf(ptr, "The array elements after sorting are:\n"); //optional
for (i = 0; i < ARRAY_SIZE; i++)
{
    fprintf(ptr, "myarray[%d] is: %d\n", i, myarray[i]);
}
fclose(ptr);
}

void selectionSort(int list[], int last)
{
    int current, walker, temp, max;
    for (current = 0; current < last; current++)
    {
        max = current;
        for (walker = current + 1; walker <= last; walker++)
            if (list[walker] > list[max])
                max = walker;
        /*smallest selected: exchange with current element*/
        temp = list[current];
        list[current] = list[max];
        list[max] = temp;
    }
}

```

/*changes highlighted in green are optional */

Testing: check the file [sortedArray.txt](#) on your disk, it should contain



```

sortedArray - Notepad
File Edit Format View Help
The array elements after sorting are:
myarray[0] is: 34
myarray[1] is: 23
myarray[2] is: 12
myarray[3] is: 12
myarray[4] is: 7
myarray[5] is: 3

```