UNIVERSITY OF MASSACHUSETTS
DARTMOUTH

ECE160: Foundations of Computer Engineering I

Lecture #23 – Arrays (3)

Instructor: Dr. Liudong Xing
SENG-213C, lxing@umassd.edu
ECE Dept.

Administrative Issues (4/5)

- Lab#10 due 5pm, Wednesday, April 5.
- Last day to withdraw from a class is **Friday, April 7**
- Today's topics
  - Arrays (Finish L#22)
  - Arrays & Functions (Then L#23)

Review of Lectures #22

- How to exchange contents of two array elements?
  - Through a temporary variable
- Multi-dimensional arrays
  - Before use, a multi-D array has to be defined and declared `int myarray[3][2];`
  - Three ways to initialize a multi-D array
    - At the definition time `int myarray [3][2] = {1,2,3,4,5,6};`
    - Inputting values from the keyboard
    - Assigning values

```
for(int i=0; i< 3; i++)    /*row index*/
    for (int j=0; j<2; j++) /*col index*/
        scanf("%d", &myarray[i][j]);
```

```
for(int i=0; i< 3; i++)    /*row index*/
    for (int j=0; j<2; j++) /*col index*/
        myarray[i][j]=2*i+j;
```

Arrays and Functions

- You can pass an individual array element to a function or you can pass a whole array to a function.

- There is a BIG DIFFERENCE though……

Functions: Review Lecture#14-17

Agenda

- Pass an individual array element to a function
- Pass an entire array to a function

Passing an Individual Element

- Treat a single array element like a simple variable!
  - Pass by values: pass the values of the element without having it changed in the function
  - Pass by reference: change the value of the array element in the function

Parameter passing: Review Lecture#16

Example (1)

```
#include "stdio.h"

void add(int number);

void main(void)
{
    int myarray[5] = {1,2,9,3,6};
    add(myarray[2]);
    printf("The value of myarray[2] is: %d\n", myarray[2]);
}

void add(int number)
{
    number = number + 100;
}
```

What is the output of the program?

Example (2)

```
#include "stdio.h"

void add(int *number);

void main(void)
{
    int myarray[5] = {1,2,9,3,6};
    add(&myarray[2]);
    printf("The value of myarray[2] is: %d\n", myarray[2]);
}

void add(int *number)
{
    *number = *number + 100;
}
```

What is the output of the program?

Agenda

- Pass individual array elements to functions
- Pass entire array to functions

Passing The Entire Array

- Passing by values won't work WELL!
  - If an array containing 3000 elements was passed by values to a function, another 3000 locations must be allocated in the function area.

- C passes the address of the array to the function!

Array Name

- C considers the name of an array as a primary expression whose value is the address of the first element in the array

- Example: assume

```
int myarray[5];
```

  - Both myarray and &myarray[0] indicate the address of the first element of the array

Passing the Entire Array

- In the calling function, use the array name as the input parameter passed to the called function

- In the called function (specifically, in the function header and function declaration), declare the parameter as an array

Example (3)

```
#include "stdio.h"

void add(int arr[]);

void main(void)
{
    int myarray[5]= {1,2,9,3,6};
   add(myarray); /* Pass the whole array to a function */
    printf("The value of myarray[2] is: %d\n", myarray[2]);
}

void add(int arr[])
{
    arr[2] = arr[2] + 100;
}
```

A variable with brackets [] in function prototype and header indicate the parameter is an array!

What is the output of the program?

Note!

- In the function prototype and header, it is not necessary to specify the number of elements in the array.
- Because the array is defined elsewhere, all that is important is that the compiler knows it is an array!

```
void add(int arr[]); //prototype
```

```
void add(int arr[]) //header
{
        arr[2] = arr[2] + 100;
}
```

Exercise (1)

- Write a program that prints the sum of the elements of an array. The array contains 10 integers which are entered from the keyboard.
  1. Enter array elements from the keyboard
  2. Compute the sum
  3. Output the sum on the screen

- Implement it without using a function.
- Implement it using a function that computes the sum of the ten array elements.

Exercise (1) – Solution 1 without using function

```
/* WITHOUT A FUNCTION */
#include "stdio.h"

void main(void)
{
   int myarray[10];
   int sum = 0;
   int i;

   /*enter 10 array elements from keyboard and
     calculate the sum*/

                    ???

   printf("The sum is %d\n", sum);
}
```

Exercise (1) – Solution 2 using a function that computes the sum

```c
/* WITH A FUNCTION */
#include "stdio.h"

int sum(int a[]); //function prototype

void main(void)
{
    int myarray[10];
    int mysum = 0;
    int i;
    for (i = 0; i < 10; i++)
    {
        scanf_s("%d", &myarray[i]);
    }
    //call the function here
                                ???
    printf("The sum is %d\n", mysum);
}

//function definition here
                                ???
```

Exercise (2)

- Write a program that replaces each element of an array with its square. The array contains 10 integers which are entered from the keyboard.

- Requirements: to do the square operation, use a function to which you pass the WHOLE ARRAY.

```c
#include "stdio.h"
#define arr_size 10
void square(int a[]); //function prototype
void main(void)
{
    int myarray[arr_size];
    int i;

    for (i = 0; i < arr_size; i++)
    {
         scanf_s("%d", &myarray[i]);
    }

    //function call here
                                        ???
    printf("The squared array elements are:\n");

                                        ???
}

//function definition here
                                        ???
```

Exercise (3)

- True or false:

a) ____In a function call, using an array name not followed by brackets as a parameter passes the entire array to the called function by copying each element into the array region of the called function

b) ____The name of an array, not followed by brackets, indicates the address of the first element of the array

Summary of Lectures #23

- We can pass an individual array elements to a function like any other variables as long as the array element type matches the function parameter type!
  - Pass by values
  - Pass by references

- To pass the whole array to a function, we pass the address of the array (via array name), i.e., pass by references!

Things To Do

- Review lecture notes
- Run programs in the exercises

Next Topics

- Arrays (Cont'd): sorting problems