Rajarata University of Sri Lanka

COM 1407 Computer Programming

LECTURE 7 (PART 2) – PASSING ARRAYS TO FUNCTIONS, HEADER FILES

BY: PIYUMI HERATH

Objectives

At the end of this lesson the students should be able to

- Practice the declaration, parameter passing, prototyping and calling functions taking arrays as parameters.
- Practice creating user defined C header files.
- Apply taught concepts for writing programs.

Functions and Arrays

- As with ordinary variables and values, it is also possible to pass the value of an array element and even an entire array as an argument to a function. However, you cannot return arrays from functions.
- So, to take the square root of averages[i] and assign the result to a variable called sq_root_result, we can pass the array element to function as follows.

```
sq_root_result = squareRoot (averages[i]);
```

Functions and Arrays (Cont…)

- To pass an array to a function, it is only necessary to list the name of the array, without any subscripts, inside the call to the function.
- As an example, if you assume that gradeScores has been declared as an array containing 100 elements, the expression

```
minimum (gradeScores);
```

passes the entire 100 elements contained in the array gradeScores to the function called minimum.

Define functions taking Arrays as parameters

```
int minimum( int array[ ] )
{
  …
  return minValue;
}
```

```
int minimum ( int array[100] )
{
  …
  return minValue;
}
```

- Length of the array doesn't matter as far as the function is concerned because C performs no bounds checking for formal parameters.
- So you can give the array size as a separate parameter to the function

- In c the array name, passed to the function, itself is the address of first element of that array. For example if array name is arr then you can say that **arr** is equivalent to the **&arr[0]**.
- So any changes that you do on the array elements within the function are reflected in the original array too.

Example :

```
#include <stdio.h>
float calculateSum(float age[]); /*Function prototype*/

int main() {
        float result, age[] = {23.4, 55, 22.6, 3, 40.5, 18};
        result = calculateSum(age); /*age array is passed to calculateSum() function*/
        printf("Result = %.2f", result);
        return 0;
}
float calculateSum(float age[])   {/*Function definition*/
        float sum = 0.0;
        for (int i = 0; i < 6; ++i) {
                sum += age[i];
         }
        return sum;
}
```

Here you can define the method to take size of the array as a parameter too..
Like: float calculateSum(float age[], int size){}

So the loop could be modified as
for (i=0;i<size;i++)

Header files

Header Files

- A header file in C programming language is a file with .h extension which contains a set of common function declarations and macro definitions which can be shared across multiple program files.
- C language provides a set of in build header files which contains commonly used utility functions and macros.
- **Types of Header Files in C**
  - User defined header files.
  - In-built header files.
- #include Preprocessor Directives is used to include both system header files and user defined header files in C Program.

Header Files (Cont…)

- **Syntax to Include In-built Header File in C Program**

```
#include <Header_file_name>
```

- Above mentioned #include syntax is used to include in-built system header files.
- It searches given header file in a standard list of directories where all in-built header files are stored.
- To include in-built header file we use angular brackets. <>

Header Files (Cont…)

- **Syntax to Include User defined Header File in C Program**
- **`#include "Header_file_name"`**
- It searches given user defined header file in a current directories where current c program exists.
- To include user-defined header file we use double quotes.

Create Your Own Header File

- Type this Code

```
int add(int a,int b)
{
return(a+b);
}
```

- Here we write only function definition as you write in General C Program
- Save this file with .h extension. Lets assume we saved this file as myhead.h.
- Save myhead.h header file in the same folder/directory where your current program is in.
- Compile this file.

Create Your Own Header File (Cont…)

- To Include your new header file in a c program used #include preprocessor directive.

```
#include<stdio.h>
#include"myhead.h"

int main () {
    int num1 = 10, num2 = 10, sum;
    sum = add(num1, num2);
    printf("Sum = : %d", sum);
    return 0;
}
```

Thank You.

Next Lesson : Type Casting