Programming – DT282/1 & DT228/1

Lab 8 – Tuesday, November 17th, 2015

Note: You are expected to finish all programmes in your own time if you do not get these done during the lab session. This is your own responsibility.

Arrays (part 3) and Pointers

Remember: Use Symbollic names in your programs. Do not hard-code.

Write separate programs to:

1. Show how to initialise two 3x4 arrays (2-Dimensional arrays with 3 rows and 4 columns in each) when they are declared. In your program, declare a 3rd 3x4 array. Multiply each corresponding element in the 1st and 2nd array and store this product in the corresponding element of the 3rd array. For example, array3[0][0] = array1[0][0] x array2[0][0], array3[0][1] = array1[0][1] x array2[0][1], etc..
2. Complete the following program so that it computes the sum of all the elements in the array:

```
int data[][] = { {3, 2, 5, 7, 4, 2},
                 {1, 4, 4, 8, 13, 1},
                 {9, 1, 0, 2,  , },
                 {0, 2, 6, 3, -1, -8}
               };

// declare the sum
int i, j;
..

// compute the sum
for ( i=0; i < ???; i++)
{
     for ( j=0; j < ???; j++)
     {
          // calculate the sum here
     }
}

// display the sum
printf( .. );
```

Be very careful with the indexes of the arrays - remember, array_name[0][0] is the element of the 1st row and 1st column.

3. Write a program that uses a 3x2 array. Your program must do the following:

a) Enter in values for each element in the array.
b) Calculate and display the sum of row 0, row 1, and row 2 separately.
c) Calculate and display the sum of column 0 and column 1 separately.
d) Find the highest number in the array and display it.

NOTE: I advise you to work the solution for this program on paper first. Do not hack code to solve this question.

4. What happens when you try and do the following:

```
// code
int var1;
int *ptr;

var1 = 1;
ptr = &var1;

..
..
printf("the address of var1 is %d", &var1); // the %d should be %p
printf("ptr contains %p", ptr); /* there is a missing & and should be
                                                              &ptr */
printf("*ptr contains %d", ptr); /* there is a missing * (indirection
                          operator) and therefore should be *ptr */
..
..

}
```

Complete the above program and observe the output.

5. What happens if you try to assign a pointer any address you pick randomly (e.g. F176BA2)?

a) Will the compiler allow you?
b) If so, print the contents of that address and see what happens. Which delimiter should you use?
c) Write a short program to declare an integer and a pointer. Make the pointer point to the integer. Input an integer value into the int variable. Using the indirection operator, output the contents of address stored in the pointer.
d) Now, increment the pointer (e.g. ptr++) and output the contents of the new address stored in the pointer. What do you see?
e) Can you assign any value to a random address (e.g. using the indirection operator, can you assign the value 10 into the address FB6546)? Try it.