EE 2331 Data Structures and Algorithms, Semester B, 2009/10

Tutorial 1: Pointers

Week 1 (14th January, 2010)

The tasks of tutorial exercises are divided into three levels. Level 1 is the basic tasks. You should have enough knowledge to complete them after attending the lecture. Level 2 is the advanced tasks. You should able to tackle them after revision. Level 3 is the challenge tasks which may be out of the syllabus and is optional to answer. I expect you to complete at least task A in the tutorial.

Outcomes of this tutorial

1. Able to print the address of pointers
2. Able to determine the size of pointers and integers
3. Able to manipulate integer pointers
4. Able to print the address of array elements
5. Able to manipulate arrays using pointers
6. Able to understand the difference between pass-by-value and pass-by-reference.

This tutorial provides practice with basic elements of pointers, addresses, values, and arrays in C.

Task A (Level 1): Integer pointers

Complete the task_a() function to print the address of, and value stored in, each of the variables. Use the format string "%d" CANNOT print the addresses. Guess how to print them. You should see addresses that look something like this: "0xbff2a00c". The initial characters "0x" tell you that hexadecimal notation is being used; the remainder of the digits give the address itself

```
void task_a() {
        int *p, a, b;
        a = 5;
        p = &a;
        …
}
```

Expected Output Format (the addresses of yours may be different):

```
Please enter your action ( 1) Task A, 2) Task B, 3) Task C ): 1
The address of p        = 0xbff2a00c
The value of p          = 0xbff2a008
The value pointed by p = 5

The address of a        = 0xbff2a008
The value of a          = 5

The address of b        = 0xbff2a004
```

Observation: Draw a memory diagram showing the location of each of the variables in your program. Can you determine the size of integer and pointer to integer using the result in task a? Demonstrate your work to your tutor.

Hints: you may not able to read the output of your program as the console will be closed automatically upon the completion of the program. Move the cursor to the last instruction of your program (i.e. "return 0;") and then press F9 to insert a breakpoint. Now run your program again and you will see that your program stops at that instruction. You are now able to read the outputs. Press the "play" button in Visual Studio to continue running the remaining part of the program.

Breakpoint is a very useful technique for debugging the programs. You may find more related debugging techniques by exploring Visual Studio.

Task B (Level 1): Arrays and pointers

Complete the task_b() function to print the address of, and value stored in the array using pointer *p*.

```
void task_b() {
        int *p, a[] = {6, 2, 4};
        p = a;
        while (p < &(a[3])) {
                ...
                p++;
        }
}
```

Expected Output Format:

```
Please enter your action ( 1) Task A, 2) Task B, 3) Task C ): 2
The address of p        = 0xbf976aec
The value of p          = 0xbf976ae0
The value pointed by p = 6

The address of p        = 0xbf976aec
The value of p          = 0xbf976ae4
The value pointed by p = 2

The address of p        = 0xbf976aec
The value of p          = 0xbf976ae8
The value pointed by p = 4
```

Observation: Are the elements of array allocated in the sequential order? Is there any empty space between them? Can you further conclude the size of integer using the result in task a and b? Is there any better way to determine the accurate size of integer and pointers?

Task C (Level 2): Swap Function

Task C suggests another implementation for reversing a character array. Write a swap() function that accepts the addresses of two array elements and interchange the content of these two elements.

Observation: Can you modify the reverse() function to use array index [] notation instead of pointer notation while keeping the swap() function remains unchanged? Why do we pass the addresses instead of passing the value of the elements to the function?