Every act of conscious learning requires the willingness to suffer an injury to one's self-esteem. That is why young children, before they are aware of their own self-importance, learn so easily. ~*Thomas Szasz*

CS1010 Programming Methodology

Week 7: One-Dimensional Arrays

To students:

Welcome back! We hope that you have made good use of the one-week recess to revise on past topics covered before the recess. Also, we hope that you have written a lot of programs on your own for your self-learning.

By now, you should know what is expected of you before, during and after discussion sessions.

Your DL may skip some of the questions here and give you some other questions instead.

Many programs for this discussion can be downloaded from cs1010 account. For example, to copy **Week7_Q1a.c**, you can type:

```
cp  ~cs1010/discussion/Week7_Q1a.c  .
```

Please be reminded that

- **Lab #3 deadline is this Saturday, 6pm**.
- Our **Term Test is next Tuesday 6:30pm!** Please look up IVLE workbin **Term Test** folder for more information.

I. Array Basics

1. **Spot the errors.**
   Spot the errors in the following programs. Are there any compilation or logic errors?

   (a) Program **Week7_Q1a.c**

Download source code from cs1010 account

```
#include <stdio.h>
int main(void)
{
    float[5] values;
    int i;

    for (i=1; i<=5; i++)
        values[i] = 2.5*i;

    for (i=1; i<=5; i++)
        printf("%f ", values[i]);
    printf("\n");

    return 0;
}
```

(b) Can you print all the elements in an array by replacing the second ‘for’ loop in the program above with just the following statement?

```
printf("%f\n", values); // to print ALL elements
```

(c) Program **Week7_Q1c.c**

```
#include <stdio.h>
float sumArray(float, int);
int main(void)
{
    float prices[6], total;
    prices = { 10.2, 5.3, 4.4, 6.8, 7.7, 9.5 };

    sumArray(prices[6], 6);
    printf("Total = %f\n", total);
    return 0;
}

float sumArray(float arr, int size)
{
    int i;
    float sum = 0.0;
    for (i=0; i<size; i++)
        sum += prices[i];
    return sum;
}
```

Download source code from cs1010 account

2. This is a true story. On 25 August 2010 we received an email from a lecturer about the following program which, when run, gives an infinite loop.

   Give a possible explanation. What is the moral of the story?

```
#include <stdio.h>
int main(void)
{
    double arr[] = { 1.1, 2.2, 3.3, 4.4 };
    int i;

    for (i=0; i<=4; i++)
    {
        printf("%d\n", i);
        arr[i] = 0;
    }
    return 0;
}
```

3. **Manual tracing.**

(a) Trace the program **Week7_Q3a.c** manually and write down the output.

```
#include <stdio.h>
#define LENGTH 5

void printArray(int [], int);
void process(int [], int);

int main(void)
{
    int numbers[] = {2, 1, 3, 0, 4};
    process(numbers, LENGTH);
    printArray(numbers, LENGTH);
    return 0;
}

void process(int arr[], int size)
{
    int i;
    for (i=0; i<size; i++)
    {
        arr[i] = arr[arr[i]];
    }
}

void printArray(int arr[], int size)
{
    int i;
    for (i=0; i<size; i++)
        printf("%d ", arr[i]);
    printf("\n");
}
```

Download source code from cs1010 account for verification after manual tracing

(b) You want to call **printArray()** somewhere in the program of **Week7_Q3a.c** for checking/debugging purpose. Where is a good place to put it?

II. Discussion: Array and Pointer

4. In Week 7 lecture, we mentioned the relationship between arrays and pointers briefly. Slide 19 reveals that an array name is the address of its first element. This is illustrated in slides 25 and 29 during lecture. Slide 26 shows an alternative syntax for array parameter in a function.

   The following program, **Week7_SumArray.c** is adapted from lecture notes.

```
#include <stdio.h>
int sumArray(int [], int);

int main(void)
{
   int foo[] = {44, 9, 17, 1, -4, 22};
   printf("sum is %d\n", sumArray(foo, 4));
   return 0;
}

// To sum up values in array 'arr' with 'size' elements
int sumArray(int arr[], int size)
{
   int i, total=0;

   for (i=0; i<size; i++)
      total += arr[i];

   return total;
}
```

Download source code from cs1010 account

   The following program, **Week7_SumArrayUsingPointer.c**, uses pointer syntax for the array parameter, as well as pointer operation in the **sumArray()** function. The **main()** function is omitted below as it is the same as the one in **Week7_SumArray.c**.

```
int sumArray(int *, int);

int main(void) { ... } // omitted for brevity

// To sum up values in array 'arr' with 'size' elements
int sumArray(int *arr, int size)
{
   int *p, total=0;

   for (p = arr; p < arr + size; p++)
      total += *p;

   return total;
}
```

Download source code from cs1010 account

III. Problem Solving with Arrays

5. An array is a collection of data. It is very common to ask the following two questions about a collection: (a) do all the data in the collection share a certain property? (b) does there exist one datum that has a certain property? The former is a *universal* question, and the later an *existential* question.

   For example, “are all the values in the array non-negative?” is a universal question; “is there one value in the array that is negative?” is an existential question. In this case, the two questions are actually the same, hence we can transform one into another.

   Write a function **nonNegative(int arr[], int size)** that returns 1 if all the elements in arr[0]… arr[size-1] are non-negative; or returns 0 otherwise.

6. Given an array of integers, write a function **isSorted(int arr[], size)** that returns 1 if the array **arr** is sorted in non-decreasing order, or returns 0 otherwise. For example, 3, 12, 15, 18 and -5, 8, 8, 10 are in non-decreasing order, but 4, 6, 9, 7, 12 is not.

   Do you see any similarity between this question and Q5? For this question, what is the property you have “abstracted” out to check?

Download skeleton **Week7_Q6.c** from cs1010 account