CS1010 Programming Methodology

Week 3: Algorithms and C Basics

> No matter how one may think himself accomplished, when he sets out to learn a new language, science, or the bicycle, he has entered a new realm as truly as if he were a child newly born into the world. ~ Frances Willard, *How I Learned to Ride the Bicycle*

To students:

The success of discussion sessions hinges very much on your (1) **PREPARATION** beforehand, (2) **ACTIVE PARTICIPATION** in class, and (3) after-class **REVISION**. Unless otherwise instructed, you do not need to submit your work for grading.

Please cooperate with your DL to work towards a fruitful and enriching learning experience.

Due to time constraint, sometimes not all the questions in the Discussion Sheet are discussed in class. Your DL has the discretion to choose the questions to discuss (or you may request your DL to discuss certain questions). Please do go through those left-over questions after class.

Also, why limit yourself only to the exercises here? You may find exercises from other sources for your own practice. The more you practice, the better you will do in this module.

I. Algorithms

Design an algorithm (in pseudo-code) for each of the following problems. Your discussion leader may ask you to present your answers.

The algorithms here may require three control structures: sequence, selection and repetition. You may assume that the input data have been read.

1. Check whether a given integer *M* appears in a list of *N* integers.

2. Given a list of *N* integers, find out how many of them are negative.

3. *N* white and black balls are arranged in a row. The example below shows a case of *N*=7.

   ● ● ○ ● ○ ● ○

   Of course, we shall choose an appropriate notation to represent our objects (balls): **B** for black ball and **W** for white ball. So the above may be presented as BBWBWBW.

   The task is to determine the least number of ‘swaps’ you need to shift all the white balls to the left of all the black balls, subject to the condition that you may only swap two neighboring balls. For our example above, the least number of swaps you need is 9. (Work it out yourself! Try out other examples as well.)

   Write an algorithm to compute the answer. For this problem, a good algorithm needs not even carry out the swapping.

II. C Basics and Programming Environment

In this section, we will go through the following topics which you should already know by now, just to make sure that you have cleared your doubts:

- Logging into UNIX system
- Using basic UNIX commands
- Using an editor (vim) to write/modify programs
- Using a compiler (gcc) to compile programs

You are encouraged to try questions 4 and 5 at home before coming for the discussion session.

4. The directory

```
~cs1010/discussion/
```

contains the following programs[1]:

Week3_Q5.c, Week3_Q7.c, Week3_Q8.c

To copy a program (e.g., Week3_Q5.c) into your own UNIX account, login to sunfire and issue this command:

```
cp ~cs1010/discussion/Week3_Q5.c .
```

Note the **dot** at the end of the command. It denotes the *current directory* which is your destination directory to download the copy of Week3_Q5.c. The dot must be present, or the cp command will not work.

Related question: if you want to create a directory called "discussion" in your own UNIX account and download Week3_Q5.c into it, what are the correct steps?

5. After you have copied **Week3_Q5.c** into your UNIX account, compile and run it.

Do you observe that the UNIX prompt appears on the same line right after the output of the program? How do you amend the program so that the UNIX prompt appears in the next line? Use the **vim** editor to amend your program, then compile and run it again.

[1] The source codes in the reference book use "void main(void)" which gives a warning message when they are compiled with **gcc** in UNIX. To avoid that, we will stick to another style "**int main(void)**".

6. Refer to question 5 above. If you compile Week3_Q5.c using the following command:

```
gcc -Wall Week3_Q5.c
```

the executable file that the compiler generates has the default filename **a.out**. This will overwrite any existing **a.out** file in that directory.

You may rename **a.out** to another name (say, Week3_Q5) using the **mv** command:

```
mv a.out Week3_Q5
```

This is tedious if you have to do this for every program. An alternative is to use a compiler option to compile a C source code into an executable file with a filename of your choice. Do you know how to do this?

7. This is **a very common mistake** made by students.

Download the program Week3_Q7.c as follows:

```
cp ~cs1010/discussion/Week3_Q7.c .
```

Compile the program as follows:

```
gcc -Wall Week3_Q7.c
```

What happens and why? How do you resolve it?

8. The following programs work but are very badly written. Explain why and how you would improve them.

(a) Program to compute the volume of a cone.

```
#include <stdio.h>
int main(void) {
double a,s,d,f;
a=3.14159; scanf("%lf %lf",&s,&d);
f=1.0/3.0*a*s*s*d;
printf("%.2f\n",f);
return 0;}
```

The above program can be downloaded as follows:

```
cp ~cs1010/discussion/Week3_Q8a.c .
```

(b)

```
#include <stdio.h>

int main(void)
{
    // declare the int variables num1, num2 and num3
    float num1,           num2, num3;

// ask user to enter two values into num1 and num2
       printf("Enter two real numbers: ");
    scanf("%f %f", &num1, &num2);

num3 = 0.0; // initialise num3 to 0.0
// divide num1 by num2, then multiply the result by num2
// and then assign the result to num3
        num3 = (num1/num2) * num2;
     printf("num3 = %f\n", num3); // display value in num3

return 0; }
```

The above program can be downloaded as follows:

```
cp ~cs1010/discussion/Week3_Q8b.c .
```

9. Compile your program Week3_Q8b.c and test it with the following inputs:

```
123.1 2.0
```

(a) What is the output of the program? Is it expected?

(b) For the purpose of checking, we can insert a printf() statement in the program to print out some intermediate result. An example is shown below:

```
printf("num1 = %f\n", num1);
```

Where to insert this statement in your program? What is the purpose? Is the output expected?

10. Download the program Week3_Q10.c from cs1010 account. Debug it until it produces the following output.

```
If you have successfully debugged all
of the programs in this lesson, you
are ready to move on to Week 4.
```

Hint: go through the error messages generated by the compiler and try to decipher them. Edit the program to correct all the errors. In addition, make your program more readable by adding appropriate indentation.