ENGG*1410: "Introductory Programming for Engineers", Assignment #9 "Pointers"

Prof. Shawki Areibi
School of Engineering, University of Guelph
Fall 2021

Start Date: Week #9, Due Date: Week #10 (Friday, 5:00 PM) in Dropbox

1. Compile and run the following C program below. Next, instead of using the variable today to print the date, use datePtr to print the date instead.

```
// Program to illustrate structure pointers

#include <stdio.h>

int main(void)
{

     struct date
     {
       int month;
       int day;
       int year;
     };

     struct date today, *datePtr;

     today.month = 9;
     today.day = 25;
     today.year = 2021;

     printf ("Today's date is %i/%i/%2.1i.\n",
              today.month,today.day,today.year % 100);

     return 0;
}
```

2. The concept of pointers to structures and structure containing pointers are very powerful ones in C, for they enable you to create sophisticated data structures, such as linked lists, doubly linked lists, and trees.
   (a) Compile the following program and execute it.
   (b) Determine the output of the program executed.

(c) Modify the program such that you declare 10 variables of type struct entry and create a linked list.

```
// Demo of the concept of Linked Lists

#include <stdio.h>

struct entry
{
    int value;
    struct entry *PtrNext;
};

int main (void)
{

   struct entry n1, n2, n3;
   int          i;

   n1.value = 100;
   n2.value = 200;
   n3.value = 300;

   n1.PtrNext = &n2;
   n2.PtrNext = &n3;

   i = n1.PtrNext->value;
   printf("current value: %i ",i);

   printf(".. next value: %i\n",n2.PtrNext->value);

   return 0;
}
```

3. Write a C program to swap two numbers using pointers and functions. The main() function will allow the user to enter two numbers and print their values before they get swapped. It will call a function by reference (i.e sending the address of the values to be swapped) and then print their values after they are swapped.

4. Write a C program to copy one array elements to another array using pointers. Follow the instructions below:

   (a) Input size and elements in first array and store it in some variable say *size* and *source_array.*

   (b) Declare another array say *dest_array* to store copy of *source_array.*

   (c) Declare appropriate pointers to the two arrays that you declared.

   (d) Print both arrays before and after copying.

5. Write a C program to find reverse of a given string using a loop. For example if the user enters ”Hello” reversing this string would produce “olleh”. Follow the instructions below:

   (a) Input a string from the user and store it in some variable say *str.*

(b) Declare another array that will store the reverse of the string, say *char reverse*

$$SIZE$$

.

(c) Find the length of the string and store it in some variable say *len*.

(d) Initialize two variables that will keep track of the original and revers string.

(e) Use a loop to copy current character from original string to reverse string.

6. Write a C program to return multiple values from a function. Follow the instructions below:

   (a) Create a function called getMinMax() that receives 4 parameters (an array, the size of the array, a pointer called min, another pointer called max)

   (b) In main() let the user enter the size of the array and enter the elements in the array.

   (c) Let main() call getMinMax.

   (d) The function getMinMax() will return the minimum value of the array received and the maximum value in the array to the calling function.