CPS 188 Lab 1 : Algorithms and Introduction to C

***Instructor**: Dr. Ufkes*

***TA**: Mohammed Emrul Hasan*

***Section**: 18*

Sayeed Ahamad

***Student Number**: 501209136*

1 Problem Sets

1.1 Problem 1

1.1.1 Algorithm

- Create an array of fixed size, which is the maximum number of inputs it can take. (Set in the program as 30)
- Take n, a variable which stores the number of elements of the array, less than maximum capacity of array.
- Iterate via for loop to take array elements as input, and print them.
- The array elements are in unsorted fashion, to sort them, make a nested loop.
- In the nested loop, the each element will be compared to all the elements below it.
- In case the element is greater than the element present below it, then they are interchanged.
- After executing the nested loop, we will obtain an array in ascending order arranged elements.

1.1.2 Computer Program

```
/*  C progrwm to accept n arrays and arrange them in an
   ascending order    */


#include <stdio.h>

void main()
{
    int q, w, e, n, array[30];
    printf("Enter the number of inputs: ");
    scanf("%d", &n);
    printf("Enter the number inputs: \n");
    for (q = 0; q < n; ++q)
    scanf("%d", &array[q]);

    for (q = 0; q < n; ++q)
    {
```

```
        for (w = q + 1; w < n; ++w)
        {
            if (array[q] > array[w])
            {
                e =  array[q];
                array[q] = array[w];
                array[w] = e;
            }
        }
    }

    printf("The arrays wrrwnged in wscending order wre given
    below \n");

    for (q = 0; q < n; ++q)
    printf("%d\n", array[q]);
}
```

Listing 1.1: *Sorting n integer value entries*

1.2 Problem 2

1.2.1 Computer Program

```
/* My first C program */

#include <stdio.h>

int main (void)
{
   printf ("This is my first C program.\n");
   return (0);
}
```

Listing 1.2: *Hello World Program*

1.2.2 Program Output Screenshot

1.3 Problem 3

1.3.1 Algorithm

- Declare, Scan and store values for base and height
- square and add both variables to themselves and store in a new initialized variable
- take the square root of the new variable
- print the variable as hypotenuse
- Add the scaned values and the hypotenuse together and return as perimeter
- Multiply the two scaned values and divide by two. Return the following vaule of the operation as the area.

1.3.2 Computer Program

```
/*  Right Triangle Hypotenuse, Perimeter & Area Calculating Program  */


#include <stdio.h>
#include <math.h>

double input(void);
void output(double base, double height);
double hypotenuse(double base, double height);
void perimeter(double base, double height, double hypotenuse);
void surface_area(double base, double height);

int main(void)
{
    double b, h = input();
    output(b, h);
}

double input(void)
{
    double base, height;
    printf("Enter the value of the base of the triangle: ");
    scanf("%lf", &base);
    printf("Enter the value of the height of the triangle: ");
    scanf("%lf", &height);
```

```
26
27      return base, height;
28  }
29
30  void output(double base, double height)
31  {
32      double hyp = hypotenuse(base, height);
33      printf("\n");
34      perimeter(base, height, hyp);
35      printf("\n");
36      surface_area(base, height);
37  }
38
39  double hypotenuse(double base, double height)
40  {
41      double sq_sum = base * base + height * height;
42      double hypotenuse = sqrt(sq_sum);
43      printf("The value of the hypotenuse of the triangle is: %
        lf", hypotenuse);
44
45      return hypotenuse;
46  }
47
48  void perimeter(double base, double height, double hypotenuse)
49  {
50      double perimeter = base + height + hypotenuse;
51      printf("The value of the perimeter of the triangle is: %
        lf", perimeter);
52  }
53
54  void surface_area(double base, double height)
55  {
56      double surface_area = ( base * height ) / 2;
57      printf("The value of the surface area of the triangle is:
        %lf", surface_area);
58  }
59
60
61
```

Listing 1.3: *Right Triangle Hypotenuse, Perimeter & Area Calculating Program*