LAB7 – Pointers
Exercise-1

Write a program to do followings.

1. Define variables below.
   - A float number (F)
   - Two pointers (P1 and P2) to float
2. Ask user to enter value of F.
3. Assign address of F to P1 pointer.
4. Display data values of F and P1.
5. Dynamically allocate a float memory, and assign its address to P2 pointer.
6. Copy data content of P1 into data content of P2.
7. Display data value of P2.
8. By using "%x" as format specifier, display the followings.
   - Memory address of F.
   - Value of P1 (an address).
   - Value of P2 (an address).

Exercise-2

Write a **MAIN** program to do followings.

1. Define a pointer (P) to int.
2. Define an int (N).
3. Ask user to enter N.
4. Dynamically allocate an integer array with N elements, assign its address to the P pointer.
5. Initialize elements of P array with random data.
6. Call the function whose prototype is:

```
float CalculateAvg (int * Array, int Size);
```

   Pass the P array and N to function as arguments.
   Function should calculate and return the average.
7. Display the returned average on screen.

Exercise-3

Modify the main program in Exercise-2 to do followings.

1. Call the function whose prototype is:

```
void CalculateStats (int * Array, int Size,
                     float * avg, int * min, int * max);
```

   Call-by-reference method

2. Pass the following arguments to function:
   - Input arguments: P array and N size.
   - Output arguments: Memory addresses of average, minimum and maximum variables.
3. Display the output arguments on screen (average, minimum, maximum).

Exercise-4

Change the function prototype in Exercise-3 as follows:

```
void CalculateStats (int * Array,
                     int Size,
                     float & avg,
                     int & min,
                     int & max);
```

Call-by-alias method