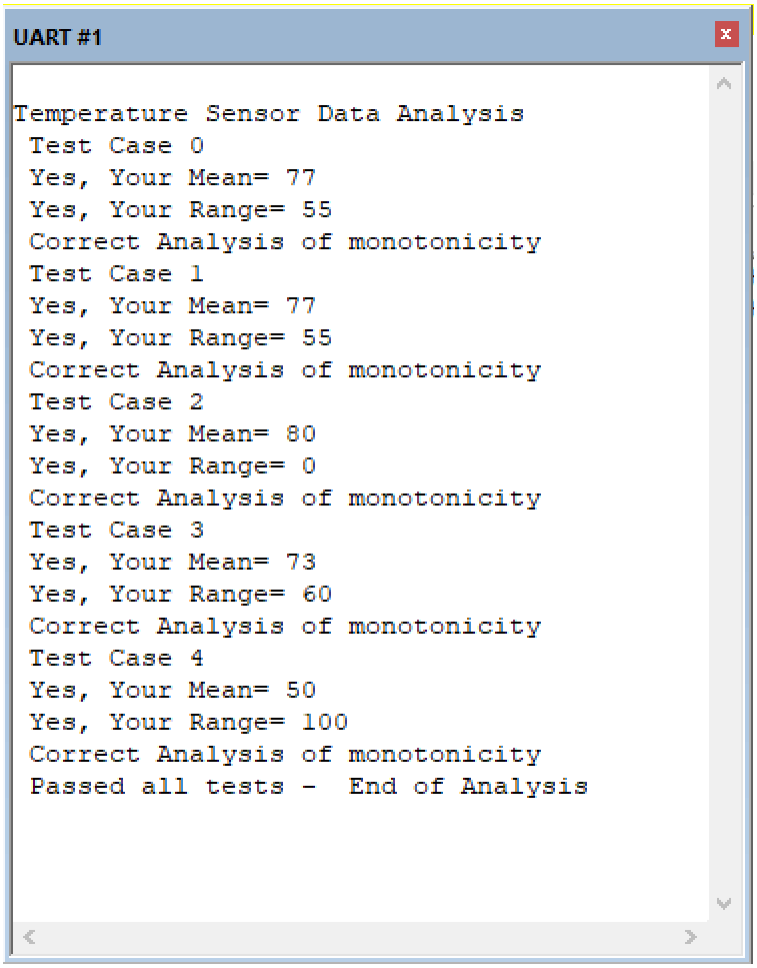

// ****************** Lab2.c ***************

// Program written by: Ali Mansoorshahi

// Date Created: 2/6/2017

// Last Modified: 2/6/2018

// Brief description of the Lab;

// This version is for the combined EE319K (Valvano) EE312 (Gligoric) sections

// An embedded system is capturing temperature data from a

// sensor and performing analysis on the captured data.

// The controller part of the system is periodically capturing size

// readings of the temperature sensor. Your task is to write three

// analysis routines to help the controller perform its function

```
//   The three analysis subroutines are:
//    1. Calculate the mean of the temperature readings
//       rounded down to the nearest integer
//    2. Calculate the range of the temperature readings,
//       defined as the difference between the largest and smallest reading
//    3. Check if the captured readings are a non-increasing montonic series
//       This simply means that the readings are sorted in non-increasing order.
//       We do not say "increasing" because it is possible for consecutive values
//       to be the same, hence the term "non-increasing". The controller performs
//       some remedial operation and the desired effect of the operation is to
//       lower the the temperature of the sensed system. This routine helps
//       verify whether this has indeed happened
#include "Lab2.h"
#define True 1
#define False 0

// Return the computed Mean
// Readings is an array of length N
// N is the length of the array
uint8_t Find_Mean(uint8_t Readings[],uint32_t N){
// Replace ths following line with your solution
        int32_t sum = 0;
        for(int i = 0; i < N; i++)
        {
                sum += Readings[i];
        }

  return(sum / N);
}
```

```
// Return the computed Range
// Readings is an array of length N
// N is the length of the array
uint8_t Find_Range(uint8_t Readings[],uint32_t N){
// Replace ths following line with your solution
        int32_t max = -200000000;
        int32_t min = 200000000;

        for(int i = 0; i < N; i++)
        {
                if(Readings[i] > max)
                {
                        max = Readings[i];
                }
                if(Readings[i] < min)
                {
                        min = Readings[i];
                }
        }

  return(max - min);
}

// Return True of False based on whether the readings
// a non-increasing montonic series
// Readings is an array of length N
// N is the length of the array
uint8_t IsMonotonic(uint8_t Readings[],uint32_t N){
```

```
// Replace ths following line with your solution


        for(int i = 0; i< N-1; i++)
        {
                if(Readings[i] < Readings[i+1])
                {
                        return False;
                }
        }


  return True;
}




//Testcase 0:
// Scores[N] = {80,75,73,72,90,95,65,54,89,45,60,75,72,78,90,94,85,100,54,98,75};
// Range=55 Mean=77 IsMonotonic=False
//Testcase 1:
// Scores[N] = {100,98,95,94,90,90,89,85,80,78,75,75,75,73,72,72,65,60,54,54,45};
// Range=55 Mean=77 IsMonotonic=True
//Testcase 2:
// Scores[N] = {80,80,80,80,80,80,80,80,80,80,80,80,80,80,80,80,80,80,80,80};
// Mean=80 Range=0 IsMonotonic=True
//Testcase 3:
// Scores[N] = {100,80,40,100,80,40,100,80,40,100,80,40,100,80,40,100,80,40,100,80,40};
// Mean=73 Range=60 IsMonotonic=False
//Testcase 4:
// Scores[N] = {100,95,90,85,80,75,70,65,60,55,50,45,40,35,30,25,20,15,10,5,0};
```

```
// Range=100 Mean=50 IsMonotonic=True
```