Introduction to Programming
Practice Problems-3
Arrays

If you have reached here after completing the previous practice problems, then it means that you are doing great :-) Congratulations !!! Let us make some programs using 1D-Arrays as we discussed in our class. You can start from the following basic Array Code.

```
#include <stdio.h>

int main() {
   int i;

   /* An array is an ordered sequence of primitive data (int, float)*/
   /* type_name array_name [array_size] */
   /* Start counting from 0 */
   /* array_size is an integral constant */
   /* warning: Array bound checking is not done in C*/

   float values[15];

   /* In C we count from 0 */
   for (i=0; i < 10; i++)
      values[i] = i * i;

   /* Print the array */
   for (i=0; i < 10; i++)
      printf ("values[%d] --> %g \n",i,  values[i]);
}
```

The above code declares an array named *values* with 15 elements of $float$ type. Each element is given the value equal to square of its index and later the array is printed on to the screen. Here comes the another set of practice problems. All the best.

1. Find the maximum and minimum value in an array of size n. Initialize each array element randomly between 0 and 100. hint: use srand() and rand() functions present in "stdlib.h"

2. Write a program to convert a decimal number to equivalent binary number.

3. Write a program to compute CGPA as discussed in our class.

4. Write a program to search an element in an array using linear search. Also, print the number of comparisons required.

5. Write a program to search an element in an array using binary search. Also, print the number of comparisons required.

6. Sort n numbers using selection sorting algorithm.

7. Write a program to find whether a string is pallindrome or not.

8. Write a C program that reads an integer n and stores the first n Fibonacci numbers in an array.

9. Write a C program that reads an integer n and uses an array to efficiently find out the first n prime numbers.

10. Read in an integer n, read in n integers and print the integer with the highest frequency.

11. Read in an integer n, read in n numbers and find out the mean, median and mode.

12. Read in two names and compare them and print them in lexicographic (dictionary) order.

13. Read in an integer n, read in n names and print the last name when compared in lexicographic order.