Practical 1

A]

Program Title:

Find the length of the smallest subarray with sum greater than k.

Code:

```
#include <stdio.h>
#include <limits.h>

int minSubarrayLength(int* arr, int n, int k) {
   int minLen = INT_MAX;
   int start = 0, sum = 0;

   for (int end = 0; end < n; end++) {
      sum += arr[end];

      // Shrink the window while sum > k
      while (sum > k) {
         if (end - start + 1 < minLen) {
            minLen = end - start + 1;
         }
         sum -= arr[start];
         start++;
      }
   }

   // If no subarray found
   if (minLen == INT_MAX) {
      return 0;
   }

   return minLen;
}

int main() {
   int n, k;

   printf("Enter size of array: ");
```

```
    scanf("%d", &n);

    int arr[n];
    printf("Enter %d elements: ", n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    printf("Enter value of k: ");
    scanf("%d", &k);

    int result = minSubarrayLength(arr, n, k);
    if (result == 0)
        printf("No subarray found with sum greater than %d\n", k);
    else
        printf("Smallest subarray length with sum > %d is %d\n", k, result);

    return 0;
}
```

Output:

```
user@user-HP-EliteBook-840-G1:~/Documents/DSAcCollege$ ./Assigment1A
Enter size of array: 5
Enter 5 elements: 1
2
3
4
5
Enter value of k: 7
Smallest subarray length with sum > 7 is 2
```

B]

Program Title: Bitonic Array

Code:

```
#include <stdio.h>


int main() {
    int n;
    printf("Enter number of elements: ");
    scanf("%d", &n);


    int arr[n];
    printf("Enter %d elements: ", n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    int left[n], right[n];
    left[0] = 1;                                    //first always has length 1
    for (int i = 1; i < n; i++) {                   //left[] (increasing counts)
        if (arr[i] > arr[i-1])
            left[i] = left[i-1] + 1;
        else
            left[i] = 1;
    }
    right[n-1] = 1;                                         // right[] (decreasing counts)
    for (int i = n-2; i >= 0; i--) {
        if (arr[i] > arr[i+1])
            right[i] = right[i+1] + 1;
        else
            right[i] = 1;
    }

    int maxLength = 1;
    for (int i = 0; i < n; i++) {                 //Find maximum bitonic length
        int bitonicLength = left[i] + right[i] - 1;
        if (bitonicLength > maxLength)
            maxLength = bitonicLength;
    }
    printf("Length of longest bitonic subarray = %d\n", maxLength);

    return 0;
}
```

Output:

```
user@user-HP-EliteBook-840-G1:~/Documents/DSAcCollege$ ./Bitonic
Enter number of elements: 5
Enter 5 elements: 10 50 80 40 60
Length of longest bitonic subarray = 4
```