1. main.c

```
#include "header.h"
#include <stdio.h>

int main(){
   int questionNumber;
   printf("Enter Question Number ( 1 - 15 ):");
   scanf("%d",&questionNumber);
   switch (questionNumber)
   {
   case 1: {
      // Q1
      int size = 10;
      int arr1[size];
      int i, answer;
      printf("Enter %d numbers:\n", size);
      for(i = 0 ; i < size ; i++){
         scanf("%d",&arr1[i]);
      }printf("Given numbers are : \n");

      for(i = 0 ; i < size ; i++){
         printf("%d ",arr1[i]);
      }
      answer = SumOfSquares(arr1 , size);
      printf("\n");
      printf("The Sum of Squares of given %d numbers is %d\n",size ,answer);
      break;
   }
   case 2: {
      // Q2
      int size = 10;
```

```
    int arr2[size];
    printf("Enter %d numbers:\n", size);
    for(int i = 0 ; i < size ; i++){
        scanf("%d",&arr2[i]);
    }
    printf("Reverse of Given Array of Integers is : \n");
    ReverseOfArray(arr2 , size);
    printf("\n");
    break;
}
case 3: {
    // Q3
    int size = 50;
    float arr3[size];
    float target;
    printf("Enter %d numbers:\n", size);
    for(int i = 0 ; i < 50 ; i++){
        scanf("%f",&arr3[i]);
    }
    printf("Enter Target\n");
    scanf("%f",&target);
    int answer = FindingElement(arr3 , 50 , target);
    if(answer != 0){
        printf("Index of Given Target Number is %d\n",answer);
    }
    else{
        printf("Not Found\n");
    }
    break;
}
case 4: {
```

```
    // Q4
    int arr4[] = {60 , 700 , 80 , 900 , 10};
    int size = sizeof(arr4)/sizeof(arr4[0]);
    TriangularPattern(arr4 , size);
    break;
}
case 5: {
    // Q5
    int size;
    printf("Enter size of integer array in bytes : ");
    scanf("%d",&size);
    printf("Number of Elements in arrya is %d",NumberOfElements(size));
    break;
}
case 6: {
    // Q6
    int arr5[] = {70 , 80 , 90 , 100 , 110};
    printf("Original Array: ");
    for(int i = 0; i < 5; i++){
        printf("%d ",arr5[i]);
    }
    printf("\n");
    int size = sizeof(arr5)/sizeof(arr5[0]);
    int n;
    printf("Enter no of rotations : ");
    scanf("%d",&n);
    RightShift(arr5 , size , n);
    printf("Right Shift with %d rotations : ",n);
    for(int i = 0 ; i < size ; i++){
        printf("%d ",arr5[i]);
    }
```

```
    int arr6[] = {70 , 80 , 90 , 100 , 110};
    printf("\n");
    LeftShift(arr6 , size , n);
    printf("Left Shift with %d rotations : ",n);
    for(int i = 0 ; i < size ; i++){
        printf("%d ",arr6[i]);
    }
    printf("\n");
    break;
}
case 7: {
    // Q7
    int arr7[] = {40 , 50 , 50 , 50 , 60 , 70 , 80 , 90 , 60 , 100 , 10};
    printf("Original Array: ");
    for(int i = 0; i < 11; i++){
        printf("%d ",arr7[i]);
    }
    printf("\n");
    int size = sizeof(arr7)/sizeof(int);
    printf("Array after Deleting Duplicate Elements : ");
    DeleteDuplicate(arr7 , size);
    break;
}
case 8: {
    // Q8
    printf("Ten random numbers in [1,100]\n");
    RandomNoGenerator(1 , 100 , 10);
    break;
}
case 9: {
```

```
    // Q9
    int arr9[20];
    int size = 20;
    printf("Enter %d numbers:\n", size);
    for(int i = 0 ; i < 20 ; i++){
        scanf("%d",&arr9[i]);
    }
    printf("Given Numbers are : \n");
    for(int i = 0 ; i < 20 ; i++){
        printf("%d ",arr9[i]);
    }
    printf("\n");
    NoOfPosNumbers(arr9 , size);
    NoOfNegNumbers(arr9 , size);
    NoOfOddNumbers(arr9 , size);
    NoOfEvenNumbers(arr9, size);
    NoOfZeroes(arr9 , size);
    break;
}
case 10: {
    // Q10
    int arr10[] = {3 , 6 , 0 , 6 , 3};
    int size = sizeof(arr10)/sizeof(arr10[0]);
    printf("Given Numbers are : \n");
    for(int i = 0 ; i < size ; i++){
        printf("%d ",arr10[i]);
    }
    printf("\n");
    Palindrome(arr10 , size);
    break;
}
```

```
case 11: {
   // Q11
   int arr11[] = {10, 45 , 32 , 16 , 88};
   printf("Original Array: ");
   for(int i = 0; i < 5; i++){
      printf("%d ",arr11[i]);
   }
   printf("\n");
   int size = sizeof(arr11)/sizeof(arr11[0]);
   ReverseArray(arr11 , size);
   printf("Reverse of Given Array is : ");
   for(int i = 0 ; i < size ; i++){
      printf("%d ",arr11[i]);
   }
   break;
}
case 12: {
   // Q12
   int arr12[] = {1 , 2 , 3 , 4 , 5 , 6 , 11 , 8 , 12 , 10};
   printf("Original Array: ");
   for(int i = 0; i < 10; i++){
      printf("%d ",arr12[i]);
   }
   printf("\n");
   int size = sizeof(arr12)/sizeof(arr12[0]);
   int target;
   printf("Enter Target Number : \n");
   scanf("%d",&target);
   printf("Nearest Lesser : %d\n",NearestLesser(arr12 , size , target));
   printf("Nearest Greater : %d\n",NearestGreater(arr12 , size , target));
   break;
```

```
}
case 13: {
   // Q13
   int A[] = { 45, 50, 70, 85, 90};
   int B[] = { 30, 40, 60, 75, 80};
   int C[10];
   SortMixArray(A , B , C);
   printf("Sorted Mixed Array is : \n");
   for(int i = 0 ; i < 10 ; i++){
      printf("%d ",C[i]);
   }
   break;
}
case 14: {

   // Q14
   int arr14[1000];
   int newarr[1000];
   GenerateArray(arr14);
   printf("Random Generated Array is : ");
   for (int i = 0; i < 1000; i++) {
      printf("%d ", arr14[i]);
   }
   printf("\n");
   printf("Numbers which are divisible by 8 or 25 in the array are : \n");
   int size = NewArray(arr14 , newarr);
   for (int i = 0 ; i < size ; i++){
      printf("%d ", newarr[i]);
   }
   break;
}
```

```
    case 15: {
        // Q15
        int arr15[] = {1 , 2 , 3 , 4 , 5 , 6 , 7};
        printf("Original Array: ");
        for(int i = 0; i < 7; i++){
            printf("%d ",arr15[i]);
        }
        printf("\n");
        int size = sizeof(arr15)/sizeof(arr15[0]);
        int secondLargest = SecondLargest(arr15 , size);
        printf("Second Largest Element in the Given Array is : %d",secondLargest);
        break;
    }
    default:
        break;
    }
    return 0;
}
```

2. header.h

```
#ifndef HEADER_H
#define HEADER_H

int SumOfSquares(int arr[] , int size);
void ReverseOfArray(int arr[] , int size);
int FindingElement(float arr[] , int size , float target);
void TriangularPattern(int arr[] , int size);
int NumberOfElements(int size);
void RightShift(int arr[] , int size, int n);
void LeftShift(int arr[] , int size , int n);
void DeleteDuplicate(int arr[] , int size);
void RandomNoGenerator(int min , int max , int no);
void NoOfPosNumbers(int arr[] , int size);
void NoOfNegNumbers(int arr[] , int size);
void NoOfOddNumbers(int arr[] , int size);
void NoOfEvenNumbers(int arr[] , int size);
void NoOfZeroes(int arr[] , int size);
void Palindrome(int arr[] , int size);
void ReverseArray(int arr[] , int size);
int NearestLesser(int arr[] , int size , int target);
int NearestGreater(int arr[] , int size , int target);
void SortMixArray(int A[] , int B[] , int C[] );
void GenerateArray(int arr[]);
int NewArray(int arr[] , int newarr[]);
int SecondLargest(int arr[] , int size);
#endif // HEADER_H
```

3. logic.c

```
#include "header.h"
#include <stdio.h>
#include <limits.h>
#include <stdlib.h>
// Q1
int SumOfSquares(int arr[], int size){
   int i , sum = 0;
   for(i = 0 ; i < size ; i++){
      sum += (arr[i]*arr[i]);
   }
   return sum;
}
// Q2
void ReverseOfArray(int arr[] , int size){
   int i;
   for(i = size-1 ; i >= 0 ; i--){
      printf("%d ",arr[i]);
   }
   return;
}
// Q3
int FindingElement(float arr[] , int size , float target){
   int i;
   for(i = 0 ; i < size ; i++){
      if(arr[i] == target){
         return i;
      }
   }
   return 0;
}
```

```
// Q4
void TriangularPattern(int arr[] , int size){
   int i,j;
   for(i = 0 ; i < size ; i++){
      for(j = 0 ; j <= i ; j++){
         printf("%d ",arr[j]);
      }
      printf("\n");
   }
}
// Q5
int NumberOfElements(int size){
   int number = size/sizeof(int);
   return number;
}
// Q6
void RightShift(int arr[] , int size, int n){
   for(int i = 1 ; i <= n ; i++){
      int temp = arr[size-1];
      for(int j = size-1 ; j >= 0 ; j-- ){
         arr[j] = arr[j - 1];
      }
      arr[0] = temp;
   }
}
void LeftShift(int arr[] , int size , int n){
   for(int i = 1 ; i <= n ; i++){
      int temp = arr[0];
      for(int j = 0 ; j < size - 1 ; j++){
         arr[j] = arr[j + 1];
      }
```

```
        arr[size-1] = temp;
    }
}

// Q7
void DeleteDuplicate(int arr[] , int size){
    for(int i = 0 ; i < size ; i++){
        for(int j = i+1 ; j < size ; j++){
            if(arr[i] == arr[j]){
                for(int k = j ; k < size - 1 ; k++){
                    arr[k] = arr[k+1];
                }
                size--;
                j--;
            }
        }
    }
    for(int i = 0 ; i < size ; i++){
        printf("%d " , arr[i]);
    }
}

// Q8
void RandomNoGenerator(int min , int max , int no){
    int c, n;
    for (c = 1; c <= no; c++) {
        n = rand() % (max - min + 1) + min;
        printf("%d\n", n);
    }
}
```

```
// Q9
void NoOfPosNumbers(int arr[] , int size){
   int count = 0;
   for(int i = 0 ; i < size ; i++){
      if(arr[i] > 0){
         count++;
      }
   }
   printf("Number of positive integers in the array is %d\n",count);
}
void NoOfNegNumbers(int arr[] , int size){
   int count = 0;
   for(int i = 0 ; i < size ; i++){
      if(arr[i] < 0){
         count++;
      }
   }
   printf("Number of negative integers in the array is %d\n",count);
}
void NoOfOddNumbers(int arr[] , int size){
   int count = 0;
   for(int i = 0 ; i < size ; i++){
      if(arr[i] % 2 != 0){
         count++;
      }
   }
   printf("Number of odd integers in the array is %d\n",count);
}
void NoOfEvenNumbers(int arr[] , int size){
   int count = 0;
   for(int i = 0 ; i < size ; i++){
```

```
        if(arr[i] % 2 == 0){
            count++;
        }
    }
    printf("Number of even integers in the array is %d\n",count);
}
void NoOfZeroes(int arr[] , int size){
    int count = 0;
    for(int i = 0 ; i < size ; i++){
        if(arr[i] == 0){
            count++;
        }
    }
    printf("Number of zeroes in the array is %d\n",count);
}

// Q10

void Palindrome(int arr[] , int size){
    int flag = 0;
    for(int i = 0, j = size - 1; i < size/2; i++ , j--){
        if(arr[i] == arr[j]){
            flag = 0;
        }
        else{
            flag = 1;
            break;
        }
    }
    if(flag == 0){
        printf("Palindrome");
```

```
    }
    else{
        printf("Not Palindrome");
    }
}

// Q11
void ReverseArray(int arr[] , int size){
    int i,j;
    for(i = 0 , j = size - 1 ; i < j ; i++ , j--){
        int temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
    }
}

// Q12
int NearestLesser(int arr[] , int size , int target){
    int lesser = INT_MIN;
    for(int i = 0 ; i < size ; i++){
        if(arr[i] < target && arr[i] > lesser){
            lesser = arr[i];
        }
    }
    return lesser;
}

int NearestGreater(int arr[] , int size , int target){
    int greater = target;
    for(int i = 0 ; i < size ; i++){
        if(arr[i] > target){
```

```
            greater = arr[i];
        }
    }
    return greater;
}

// Q13
void SortMixArray(int A[] , int B[] , int C[] ){
    int i,j = 0 ,k = 0;
    for(i = 0 ; i < 10 ; i++){
        if(j < 5 && (k >= 5 || A[j] <= B[k])){
            C[i] = A[j];
            j++;
        }
        else{
            C[i] = B[k];
            k++;
        }
    }
}

// Q14
void GenerateArray(int arr[]){
    int i, n;
    for (i = 1; i < 1000; i++) {
        n = rand() % (1000 - 1 + 1) + 1;
        arr[i] = n;
    }
}
int NewArray(int arr[] , int newarr[]){
    int size = 0;
```

```
    for(int i = 0 ; i < 1000 ; i++){
        if((arr[i] % 8 == 0) || (arr[i] % 15 == 0)){
            newarr[size] = arr[i];
            size++;
        }
    }
    return size;
}

// Q15
int SecondLargest(int arr[] , int size){
    int max = INT_MIN;
    int smax = INT_MIN;
    for(int i = 0 ; i < size ; i++){
        if(arr[i] > max){
            smax = max;
            max = arr[i];
        }
        else if(smax < arr[i] && max != arr[i]){
            smax = arr[i];
        }
    }
    return smax;
}
```

Outputs Of Array Lab Assignment:

```
E:\COEP\DSA\Assignments\ConditionalStatementsLabAssignment>gcc -Wall main.c logic.c -o a

E:\COEP\DSA\Assignments\ConditionalStatementsLabAssignment>a
Enter Question Number ( 1 - 15 ):1
Enter 10 numbers:
1 2 3 4 5 6 7 8 9 10
Given numbers are :
1 2 3 4 5 6 7 8 9 10
The Sum of Squares of given 10 numbers is 385

E:\COEP\DSA\Assignments\ConditionalStatementsLabAssignment>a
Enter Question Number ( 1 - 15 ):2
Enter 10 numbers:
1 2 3 4 5 6 7 8 9 10
Reverse of Given Array of Integers is :
10 9 8 7 6 5 4 3 2 1

E:\COEP\DSA\Assignments\ConditionalStatementsLabAssignment>a
Enter Question Number ( 1 - 15 ):3
Enter 50 numbers:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
Enter Target
36
Index of Given Target Number is 35

E:\COEP\DSA\Assignments\ConditionalStatementsLabAssignment>a
Enter Question Number ( 1 - 15 ):4
60
60 700
60 700 80
60 700 80 900
60 700 80 900 10
```

```
Enter Question Number ( 1 - 15 ):5
Enter size of integer array in bytes : 32
Number of Elements in arrya is 8
E:\COEP\DSA\Assignments\ConditionalStatementsLabAssignment>a
Enter Question Number ( 1 - 15 ):6
Original Array: 70 80 90 100 110
Enter no of rotations : 3
Right Shift with 3 rotations : 90 100 110 70 80
Left Shift with 3 rotations : 100 110 70 80 90

E:\COEP\DSA\Assignments\ConditionalStatementsLabAssignment>a
Enter Question Number ( 1 - 15 ):7
Original Array: 40 50 50 50 60 70 80 90 60 100 10
Array after Deleting Duplicate Elements : 40 50 60 70 80 90 100 10
E:\COEP\DSA\Assignments\ConditionalStatementsLabAssignment>a
Enter Question Number ( 1 - 15 ):8
Ten random numbers in [1,100]
42
68
35
1
70
25
79
59
63
65

E:\COEP\DSA\Assignments\ConditionalStatementsLabAssignment>
```

```
E:\COEP\DSA\Assignments\ConditionalStatementsLabAssignment>a
Enter Question Number ( 1 - 15 ):9
Enter 20 numbers:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Given Numbers are :
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Number of positive integers in the array is 20
Number of negative integers in the array is 0
Number of odd integers in the array is 10
Number of even integers in the array is 10
Number of zeroes in the array is 0

E:\COEP\DSA\Assignments\ConditionalStatementsLabAssignment>a
Enter Question Number ( 1 - 15 ):10
Given Numbers are :
3 6 0 6 3
Palindrome
E:\COEP\DSA\Assignments\ConditionalStatementsLabAssignment>11
'11' is not recognized as an internal or external command,
```

```
E:\COEP\DSA\Assignments\ConditionalStatementsLabAssignment>a
Enter Question Number ( 1 - 15 ):9
Enter 20 numbers:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Given Numbers are :
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Number of positive integers in the array is 20
Number of negative integers in the array is 0
Number of odd integers in the array is 10
Number of even integers in the array is 10
Number of zeroes in the array is 0

E:\COEP\DSA\Assignments\ConditionalStatementsLabAssignment>a
Enter Question Number ( 1 - 15 ):10
Given Numbers are :
3 6 0 6 3
Palindrome
```

```
E:\COEP\DSA\Assignments\ConditionalStatementsLabAssignment>a
Enter Question Number ( 1 - 15 ):12
Original Array: 1 2 3 4 5 6 11 8 12 10
Enter Target Number :
12
Nearest Lesser : 11
Nearest Greater : 12

E:\COEP\DSA\Assignments\ConditionalStatementsLabAssignment>a
Enter Question Number ( 1 - 15 ):11
Original Array: 10 45 32 16 88
Reverse of Given Array is : 88 16 32 45 10
E:\COEP\DSA\Assignments\ConditionalStatementsLabAssignment>a
Enter Question Number ( 1 - 15 ):12
Original Array: 1 2 3 4 5 6 11 8 12 10
Enter Target Number :
12
Nearest Lesser : 11
Nearest Greater : 12

E:\COEP\DSA\Assignments\ConditionalStatementsLabAssignment>a
Enter Question Number ( 1 - 15 ):13
Sorted Mixed Array is :
30 40 45 50 60 70 75 80 85 90
```

```
E:\COEP\DSA\Assignments\ConditionalStatementsLabAssignment>a
Enter Question Number ( 1 - 15 ):15
Original Array: 1 2 3 4 5 6 7
Second Largest Element in the Given Array is : 6
E:\COEP\DSA\Assignments\ConditionalStatementsLabAssignment>
```