Birla Institute of Technology,

Off Campus Deoghar

PROGRAMMING FOR PROBLEM SOLVING LAB

NAME:	AKASH DIP
ROLL NUMBER:	BTECH/60002/20
SEMESTER:	2ND
BRANCH:	COMPUTER SCIENCE ENGINEERING

Birla Institute of Technology, Mesra

Off Campus Deoghar

INDEX

S. NO.	ASSIGNMENT NO.	DATE	PAGE NO.	REMARKS
1.	**ASSIGNMENT 1**	**28/04/2021**	**Turned In**	
2.	**ASSIGNMENT 2**	**05/05/2021**	**Turned In**	
3.	**ASSIGNMENT 3**	**12/05/2021**	**2**	

PROGRAMMING FOR PROBLEM SOLVING LAB

Akash Dip
BTECH/60002/20
CSE 2020
btech60002.20@bitmesra.ac.in

ASSIGNMENT NUMBER – 3

[DATE: 12/05/2021]

PROBLEM NO.	PROBLEM STATEMENT	PAGE NO.	REMARKS
1	Write a C program to convert a binary number into decimal number.	3	
2	Write a C program to read the given n numbers. Find the sum of all positive and negative numbers. Find out which sum is larger in magnitude. Also print the difference in magnitude.	4	
3	Write a C program to find out the duplicate elements in an array. Example {1,2,2,5,5,6}. It will print 2 and 5 are duplicate.	6	
4	Write a C program to implement binary search. Explain the working procedure of the algorithm.	7	
5	Write a C program to implement binary search. Explain the working procedure of the algorithm.	9	
6	Write a C program to implement binary search. Explain the working procedure of the algorithm.	10	

PROGRAMMING LANGUAGE USED: **C**

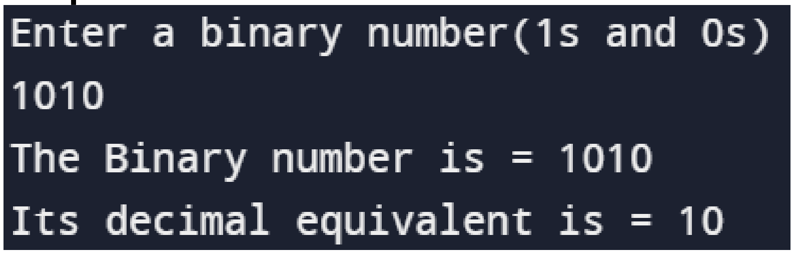
Problem Number: 1

Problem Statement: Write a C program to convert a binary number into decimal number.

Solution:

```
//1
#include <stdio.h>
void main()
{
    int num, binary_val, decimal_val = 0, base = 1, rem;
    printf("Enter a binary number(1s and 0s) \n");
    scanf("%d", &num); /* maximum five digits */
    binary_val = num;
    while (num > 0)
    {
        rem = num % 10;
        decimal_val = decimal_val + rem * base;
        num = num / 10;
        base = base * 2;
    }
    printf("The Binary number is = %d \n", binary_val);
    printf("Its decimal equivalent is = %d \n", decimal_val);
}
```

Output Discussion:



```
Enter a binary number(1s and 0s)
1010
The Binary number is = 1010
Its decimal equivalent is = 10
```

On accepting a binary value of 1010, the program calculates its equivalence in decimal form by multiplying each digit of the binary number with the power of 2 and add each multiplication result. The power starts from 0 to n-1 where n is the total number of digits in the binary number and finally returns 10.

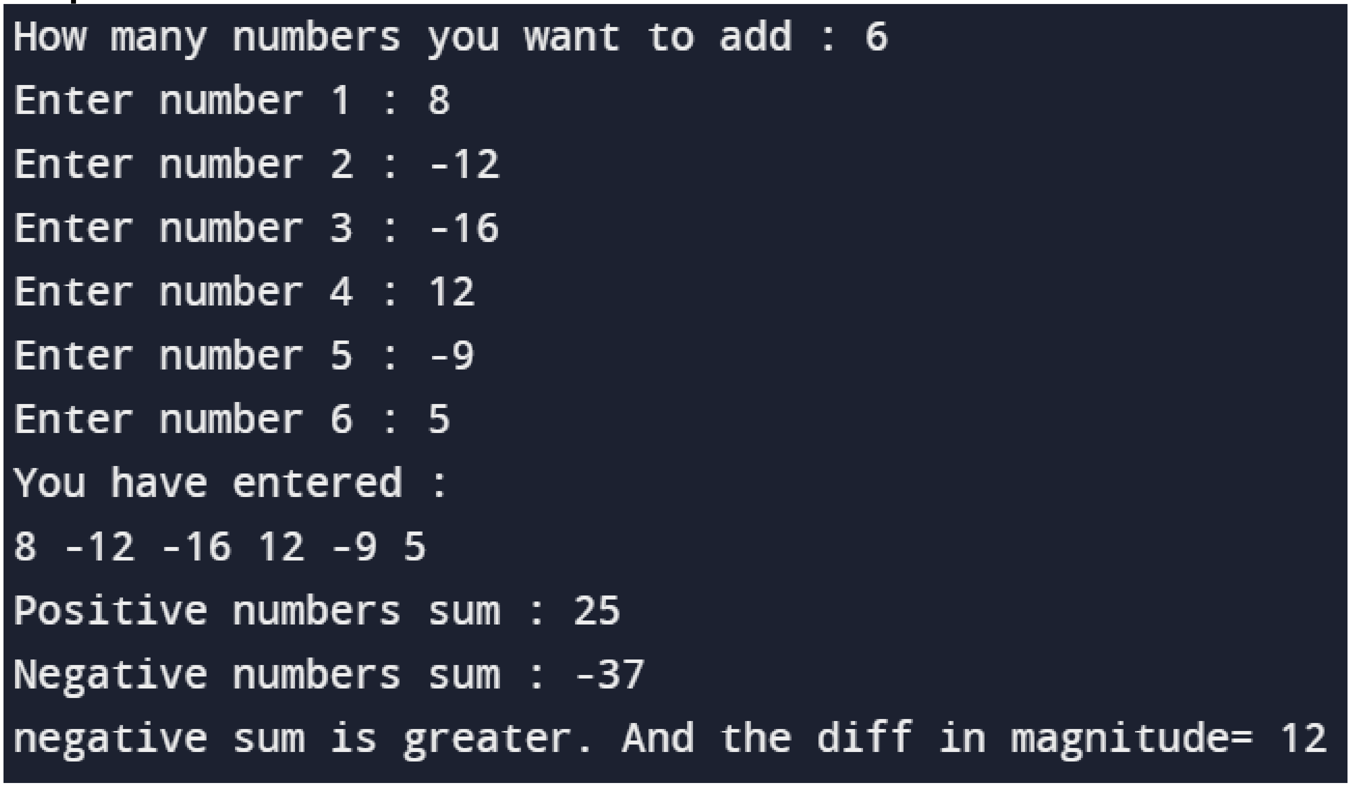
Problem Number: 2

Problem Statement: Write a C program to read the given n numbers. Find the sum of all positive and negative numbers. Find out which sum is larger in magnitude. Also print the difference in magnitude.

Solution:

```
//2
#include<stdio.h>
#include<math.h>
int main(){
    int total;
    int i;
    int positiveSum = 0;
    int negativeSum = 0;
    printf("How many numbers you want to add : ");
    scanf("%d",&total);
    int numbers[total];
    for(i=0; i<total; i++){
        printf("Enter number %d : ",(i+1));
        scanf("%d",&numbers[i]);
    }
    for(i=0 ; i<total ; i++){
       if(numbers[i] < 0){
         negativeSum += numbers[i];
       }else{
         positiveSum += numbers[i];
       }
    }
    printf("You have entered : \n");
    for(i=0 ; i<total; i++){
     printf("%d ",numbers[i]);
    }
    printf("\nPositive numbers sum : %d",positiveSum);
    printf("\nNegative numbers sum : %d\n",negativeSum);
    if(positiveSum==abs(negativeSum))
    printf("Equal");
    if(positiveSum<abs(negativeSum))
    printf("negative sum is greater. And the diff in magnitude= %d",
(abs(negativeSum)-positiveSum));
    else
    printf("positive sum is greater. And the diff in magnitude= %d",
(positiveSum-abs(negativeSum)));
    return 0;
}
```

Output Discussion:



```
How many numbers you want to add : 6
Enter number 1 : 8
Enter number 2 : -12
Enter number 3 : -16
Enter number 4 : 12
Enter number 5 : -9
Enter number 6 : 5
You have entered :
8 -12 -16 12 -9 5
Positive numbers sum : 25
Negative numbers sum : -37
negative sum is greater. And the diff in magnitude= 12
```

On addition of positive inputs i.e., 8 + 12 + 5 = 25 likewise of negative inputs i.e., (-12) + (-16) + (-9) = -37 and then on subtracting the higher magnitude to the lower gives an output of 12.

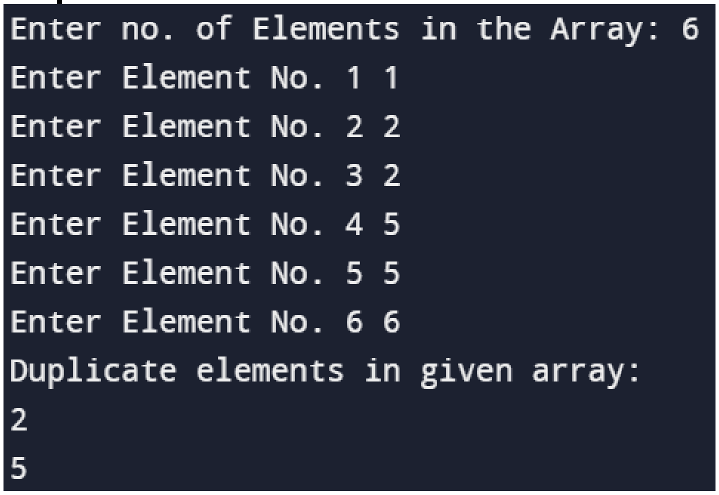
Problem Number: 3

Problem Statement: Write a C program to find out the duplicate elements in an array. Example {1,2,2,5,5,6}. It will print 2 and 5 are duplicate.

Solution:

```c
//3
#include <stdio.h>
int main()
{
    //Initialize array
    int n;
    printf("Enter no. of Elements in the Array: ");
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        printf("Enter Element No. %d ", (i+1));
        scanf("%d", &arr[i]);
    }
    printf("Duplicate elements in given array: \n");
    //Searches for duplicate element
    for(int i = 0; i < n; i++) {
        for(int j = i + 1; j < n; j++) {
            if(arr[i] == arr[j])
                printf("%d\n", arr[j]);
        }
    }
    return 0;
}
```

Output Discussion:



```
Enter no. of Elements in the Array: 6
Enter Element No. 1 1
Enter Element No. 2 2
Enter Element No. 3 2
Enter Element No. 4 5
Enter Element No. 5 5
Enter Element No. 6 6
Duplicate elements in given array:
2
5
```

The program accepts Number of Elements in the array and then searches for duplicate numbers and then ultimately prints the values.

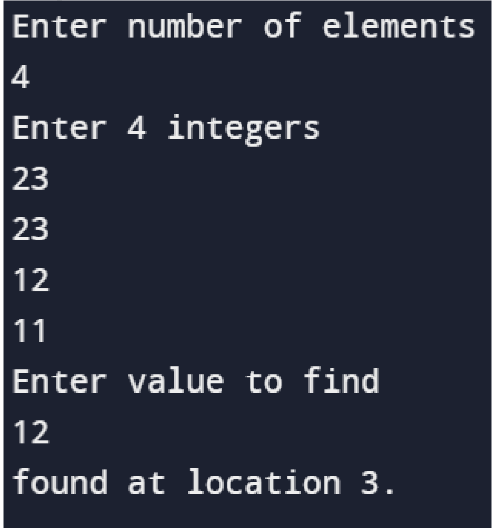
Problem Number: 4

Problem Statement: Write a C program to implement binary search. Explain the working procedure of the algorithm.

Solution:

```
//4
#include <stdio.h>
int main()
{
  int c, first, last, middle, n, search, array[100];
  printf("Enter number of elements\n");
  scanf("%d", &n);
  printf("Enter %d integers\n", n);
  for (c = 0; c < n; c++)
    scanf("%d", &array[c]);
  printf("Enter value to find\n");
  scanf("%d", &search);
  first = 0;
  last = n - 1;
  middle = (first+last)/2;
  while (first <= last) {
    if (array[middle] < search)
      first = middle + 1;
    else if (array[middle] == search) {
      printf("%d found at location %d.\n", search, middle+1);
      break;
    }
    else
      last = middle - 1;
    middle = (first + last)/2;
  }
  if (first > last)
    printf("Not found! %d isn't present in the list.\n", search);
  return 0;
}
```

Output Discussion:



```
Enter number of elements
4
Enter 4 integers
23
23
12
11
Enter value to find
12
found at location 3.
```

Search a sorted array by repeatedly dividing the search interval in half. Begin with an interval covering the whole array. If the value of the search key is less than the item in the middle of the interval, narrow the interval to the lower half. Otherwise, narrow it to the upper half. Repeatedly check until the value is found or the interval is empty. The idea of binary search is to use the information that the array is sorted and reduce the time complexity to O(Log n).

ALGORITHM DISCUSSION:

```
➢ Step 1: [INITIALIZE] SET BEG = lower_bound
  END = upper_bound, POS = - 1
➢ Step 2: Repeat Steps 3 and 4 while BEG <=END
➢ Step 3: SET MID = (BEG + END)/2
➢ Step 4: IF A[MID] = VAL
  SET POS = MID
  PRINT POS
  Go to Step 6
  ELSE IF A[MID] > VAL
  SET END = MID - 1
  ELSE
  SET BEG = MID + 1
  [END OF IF]
  [END OF LOOP]
➢ Step 5: IF POS = -1
  PRINT "VALUE IS NOT PRESENT IN THE ARRAY"
  [END OF IF]
➢ Step 6: EXIT
```

Problem Number: 5

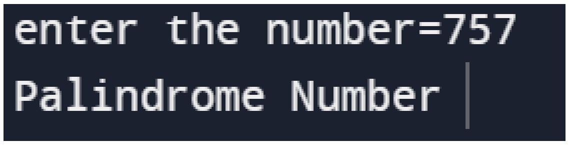
Problem Statement: Write a C program to implement binary search. Explain the working procedure of the algorithm.

Solution:

```
//5
#include<stdio.h>
int main()
{
    int n,r,sum=0,temp;
    printf("enter the number=");
    scanf("%d",&n);
    temp=n;
    while(n>0)
    {
        r=n%10;
        sum=(sum*10)+r;
        n=n/10;
    }
    if(temp==sum)
    printf("Palindrome Number ");
    else
    printf("Not a Palindrome Number");
    return 0;
}
```

Output Discussion:

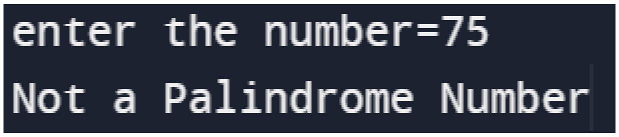
Sample I/O 1:



```
enter the number=757
Palindrome Number
```

Number 57 is not equal to its reverse hence the number is not a palindrome number.

Sample I/O 2:



```
enter the number=75
Not a Palindrome Number
```

Number 75 is not equal to its reverse hence the number is not a palindrome number.

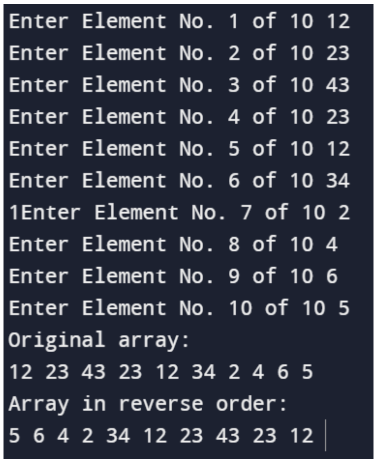
Problem Number: 6

Problem Statement: Write a C program to implement binary search. Explain the working procedure of the algorithm.

Solution:

```
//6
#include <stdio.h>
int main()
{
    //Initialize array
    int arr[10];
    for(int i=0;i<10;i++)
    {
        printf("Enter Element No. %d of 10 ", (i+1));
        scanf("%d", &arr[i]);
    }
    printf("Original array: \n");
    for (int i = 0; i < 10; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
    printf("Array in reverse order: \n");
    //Loop through the array in reverse order
    for (int i = 9; i >= 0; i--) {
        printf("%d ", arr[i]);
    }
    return 0;
}
```

Output Discussion:



```
Enter Element No. 1 of 10 12
Enter Element No. 2 of 10 23
Enter Element No. 3 of 10 43
Enter Element No. 4 of 10 23
Enter Element No. 5 of 10 12
Enter Element No. 6 of 10 34
1Enter Element No. 7 of 10 2
Enter Element No. 8 of 10 4
Enter Element No. 9 of 10 6
Enter Element No. 10 of 10 5
Original array:
12 23 43 23 12 34 2 4 6 5
Array in reverse order:
5 6 4 2 34 12 23 43 23 12
```

On entering the elements as 1,3,5,7,9,6,7,8,9 and 3, the reverse array comes out to be 3 9 8 7 6 9 7 5 3 1 .