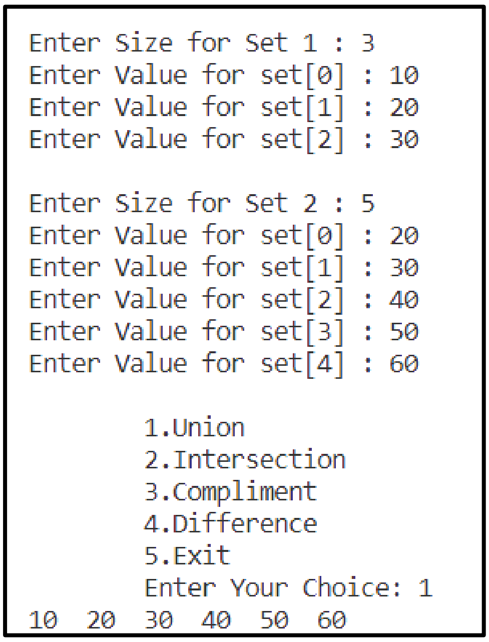
Practical No. 05

Program:

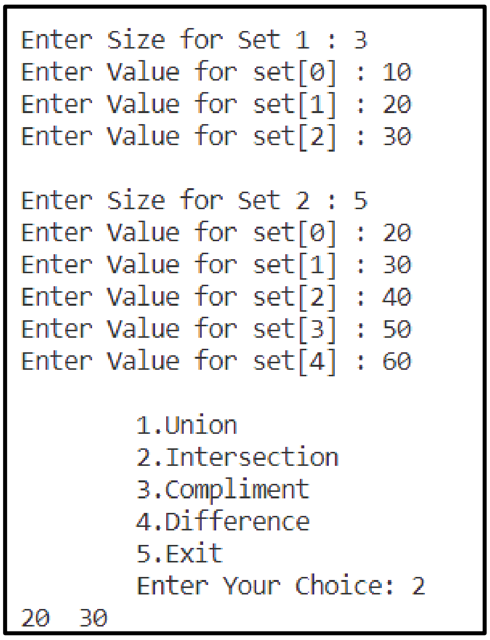
```
/*
 Write the C program to print Following statements :
    ❖  Union
    ❖  Intersection
    ❖  Difference
    ❖  Complement
*/

#include <stdio.h>

void setValue(int arr[], int size)
{
    for (int i = 0; i < size; i++)
    {
        printf("Enter Value for set[%d] : ", i);
        scanf("%d", &arr[i]);
    }
}

void unionOperation(int arr1[], int arr2[], int size1, int size2)
{
    for (int i = 0; i < size1; i++)
    {
        int flag = 0;

        for (int j = 0; j < size2; j++)
        {
            if (arr1[i] == arr2[j])
            {
                printf("%d  ", arr1[i]);
                flag = 1;
            }
        }
        if (flag == 0)
        {
            printf("%d  ", arr1[i]);
        }
    }

    for (int i = 0; i < size2; i++)
    {
        int flag = 0;

        for (int j = 0; j < size1; j++)
        {
            if (arr2[i] == arr1[j])
            {
                // printf("%d  ", arr2[j]);
                flag = 1;
            }
        }
        if (flag == 0)
        {
            printf("%d  ", arr2[i]);
        }
    }
}

void intersectionOperation(int arr1[], int arr2[], int size1, int size2)
{
    for (int i = 0; i < size1; i++)
    {
        for (int j = 0; j < size2; j++)
        {
            if (arr1[i] == arr2[j])
            {
                printf("%d  ", arr1[i]);
            }
        }
    }
}

void complimentOperation(int set[], int size1, int compSet[], int size2)
{
    for (int i = 0; i < size1; i++)
    {
        int flag = 0;

        for (int j = 0; j < size2; j++)
        {
            if (set[i] == compSet[j])
            {
                // printf("%d  ", arr1[i]);
                flag = 1;
            }
        }
        if (flag == 0)
        {
            printf("%d  ", set[i]);
        }
    }
}
```

```c
void differenceOperation(int arr1[], int size1, int arr2[], int size2)
{
    for (int i = 0; i < size1; i++)
    {
        int flag = 0;
        for (int j = 0; j < size2; j++)
        {
            if (arr1[i] == arr2[j])
            {
                flag = 1;
            }
        }
        if (flag == 0)
        {
            printf("%d  ", arr1[i]);
        }
    }
}

int main()
{
    int size1, size2, n, compArrSize;

    printf("\nEnter Size for Set 1 : ");
    scanf("%d", &size1);
    int set1[size1];
    setValue(set1, size1);

    printf("\nEnter Size for Set 2 : ");
    scanf("%d", &size2);
    int set2[size2];
    setValue(set2, size2);

    int choice;

    OUTER_SWITCH:

    printf("\n\t1.Union\n\t2.Intersection\n\t3.Compliment\n\t4.Difference\n\t5
.Exit\n\tEnter Your Choice: ");
    scanf("%d", &choice);

    switch (choice)
    {
    case 1:
        unionOperation(set1, set2, size1, size2);
        break;

    case 2:
        intersectionOperation(set1, set2, size1, size2);
        break;

    case 3:

        printf("\n\t1. Compliment for SET-1.\n\t2. Compliment for SET-2.\n\t3. 
Exit");
        printf("\nEnter Your Choice : ");
        scanf("%d", &n);

        int compArr[100];

        switch (n)
        {

        case 1:
            printf("How Many Element you want to compliment : ");
            scanf("%d", &compArrSize);
            compArr[compArrSize];
            setValue(compArr, compArrSize);
            complimentOperation(set1, size1, compArr, compArrSize);
            break;

        case 2:
            printf("How Many Element you want to compliment : ");
            scanf("%d", &compArrSize);
            compArr[compArrSize];
            setValue(compArr, compArrSize);
            complimentOperation(set2, size2, compArr, compArrSize);
            break;

        case 3:
            goto OUTER_SWITCH;

        default:
            printf("Enter Valid Choice.");
        }

    case 4:
        differenceOperation(set1, size1, set2, size2);
        break;

    case 5:
        break;

    default:
        printf("Invalid Input...");
    }
    return 0;
}
```

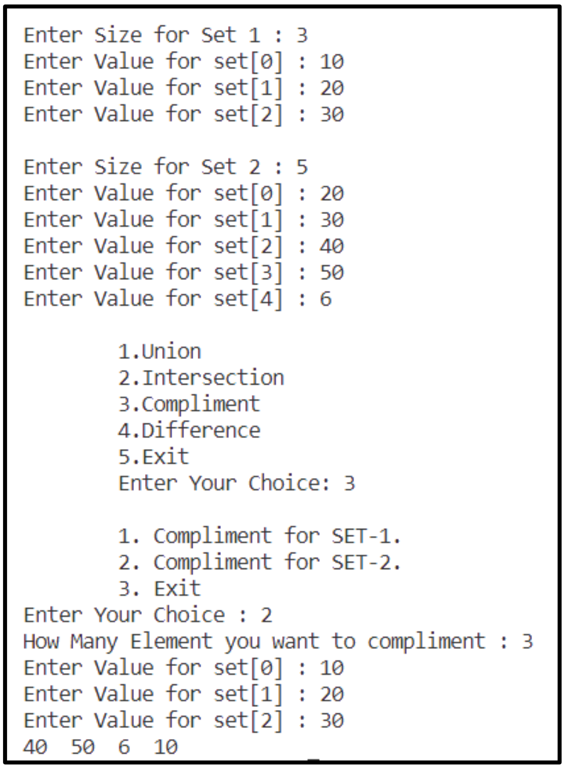
Output :



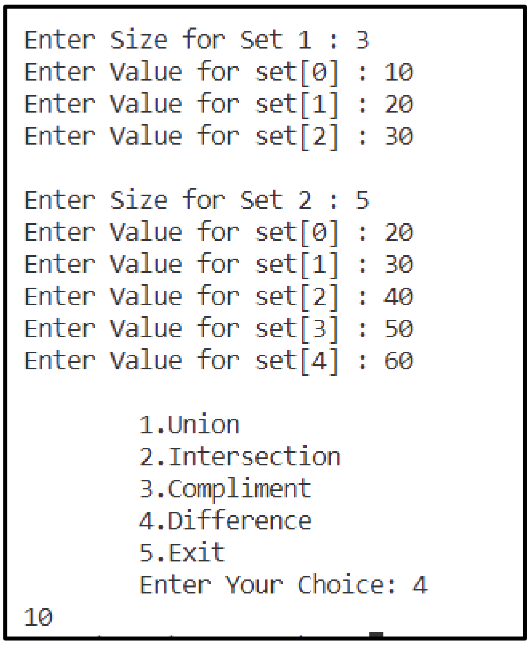
```
Enter Size for Set 1 : 3
Enter Value for set[0] : 10
Enter Value for set[1] : 20
Enter Value for set[2] : 30

Enter Size for Set 2 : 5
Enter Value for set[0] : 20
Enter Value for set[1] : 30
Enter Value for set[2] : 40
Enter Value for set[3] : 50
Enter Value for set[4] : 60

        1.Union
        2.Intersection
        3.Compliment
        4.Difference
        5.Exit
        Enter Your Choice: 1
10  20  30  40  50  60
```



```
Enter Size for Set 1 : 3
Enter Value for set[0] : 10
Enter Value for set[1] : 20
Enter Value for set[2] : 30

Enter Size for Set 2 : 5
Enter Value for set[0] : 20
Enter Value for set[1] : 30
Enter Value for set[2] : 40
Enter Value for set[3] : 50
Enter Value for set[4] : 60

        1.Union
        2.Intersection
        3.Compliment
        4.Difference
        5.Exit
        Enter Your Choice: 2
20  30
```



```
Enter Size for Set 1 : 3
Enter Value for set[0] : 10
Enter Value for set[1] : 20
Enter Value for set[2] : 30

Enter Size for Set 2 : 5
Enter Value for set[0] : 20
Enter Value for set[1] : 30
Enter Value for set[2] : 40
Enter Value for set[3] : 50
Enter Value for set[4] : 6

        1.Union
        2.Intersection
        3.Compliment
        4.Difference
        5.Exit
        Enter Your Choice: 3

        1. Compliment for SET-1.
        2. Compliment for SET-2.
        3. Exit
Enter Your Choice : 2
How Many Element you want to compliment : 3
Enter Value for set[0] : 10
Enter Value for set[1] : 20
Enter Value for set[2] : 30
40  50  6  10
```



```
Enter Size for Set 1 : 3
Enter Value for set[0] : 10
Enter Value for set[1] : 20
Enter Value for set[2] : 30

Enter Size for Set 2 : 5
Enter Value for set[0] : 20
Enter Value for set[1] : 30
Enter Value for set[2] : 40
Enter Value for set[3] : 50
Enter Value for set[4] : 60

        1.Union
        2.Intersection
        3.Compliment
        4.Difference
        5.Exit
        Enter Your Choice: 4
10
```