NAME: BHAVYA JINDAL

ADDMISSION NO - 2019b101043

**EXPERIMENT NUMBER: 1**

Objective: To write a program in C to create two sets and perform the Union operation on sets.

Definition: The union of two sets A and B, is the set of elements which are in A or in B or in both. It is denoted by A B, and is read "A union B".

Procedure: To find union of two sorted arrays, follow the following merge procedure:

1) Use two index variables i and j, initial values i = 0, j = 0 2) If arr1[i] is smaller than arr2[j] then print arr1[i] and increment i. 3) If arr1[i] is greater than arr2[j] then print arr2[j] and increment j. 4) If both are same then print any of them and increment both i and j. 5) Print remaining elements of the larger array.

**Code:**

#include <stdio.h>

int Union(int arr1[], int arr2[], int m, int n)

{

int i = 0, j = 0,k=0;

printf("First Array is : ");

for(k=0;k<m;k++)

{

printf("%d ",arr1[k]);

}

printf("\nSecond Array is : ");

for(k=0;k<n;k++)

{

printf("%d ",arr2[k]);

}

printf("\nUnion of two array is : ");

while (i < m && j < n)

{

if (arr1[i] < arr2[j])

printf(" %d ", arr1[i++]);

else if (arr2[j] < arr1[i])

printf(" %d ", arr2[j++]);

else

{

printf(" %d ", arr2[j++]);

i++;

}

}

while (i < m)

printf(" %d ", arr1[i++]);

while (j < n)

printf(" %d ", arr2[j++]);

}

int main()

{

int arr1[] = { 1, 2, 4, 5, 6 };

int arr2[] = { 2, 3, 5, 7 };

int m = sizeof(arr1) / sizeof(arr1[0]);

int n = sizeof(arr2) / sizeof(arr2[0]);

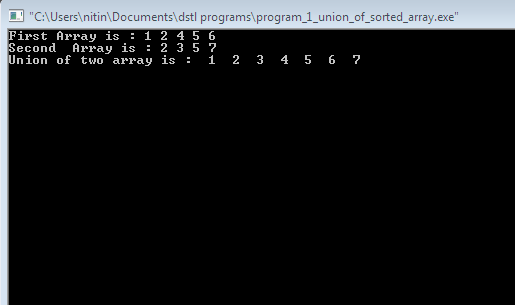
Union(arr1, arr2, m, n);

getchar();

return 0;

}

**Output Snapshot:**

****

**EXPERIMENT NUMBER: 2**

Objective: Write a program in C to create two sets and perform the Intersection operation on sets.

Definition : The intersection of two sets X and Y is the set of elements that are common to both set X and set Y. It is denoted by X ∩ Y and is read ‘X intersection Y ’.

Example: X = {1, 2, 5, 6, 7, 9, 10} and Y = {1, 3, 4, 5, 6, 8, 10}

Solution: We find that X ∩ Y = {1, 5, 6, 10} ← in both X and Y

Procedure: To find intersection of 2 sorted arrays, follow the below approach :

1) Use two index variables i and j, initial values i = 0, j = 0 2) If arr1[i] is smaller than arr2[j] then increment i. 3) If arr1[i] is greater than arr2[j] then increment j. 4) If both are same then print any of them and increment both i and j.

**Code:**

#include <stdio.h>

int Intersection(int a1[], int a2[], int m, int n)

{

int i = 0, j = 0,k=0;

printf("First Array is : ");

for(k=0;k<m;k++)

{

printf("%d ",a1[k]);

}

printf("\nSecond Array is : ");

for(k=0;k<n;k++)

{

printf("%d ",a2[k]);

}

printf("\nIntersection of two array is : ");

while (i<m && j<n)

{

if (a1[i] < a2[j])

i++;

else if (a2[j] < a1[i])

j++;

else

{

printf(" %d ", a2[j]);

i++;

j++;

}

}

printf(“\n”);

}

int main()

{

int a1[] = { 1, 2, 4, 5, 7, 8 };

int a2[] = { 2, 3, 5, 7, 8 };

int m = sizeof(a1) / sizeof(a1[0]);

int n = sizeof(a2) / sizeof(a2[0]);

Intersection(a1, a2, m, n);

return 0;

}

**Output:**

