Assignment No. 2

**TITLE : Operations on Sets**

Name : METHIKA M Date : 10/08/2025

UNo : UIT2024936

Class: IT- I DIV

Batch : I2  
  
**SOURCE CODE**

/\*

\* set.c

\*

\* Created on: Aug 4, 2025

\* Author: administrator

\*/

#include <stdio.h>

int accept(int fa[], int fn);

int display(int fa[], int fn);

void display\_result(int arr[], int count);

int set\_union(int fa[], int fb[], int fn, int fm, int result[]);

int set\_intersection(int fa[], int fb[], int fn, int fm, int result[]);

int set\_difference(int fa[], int fb[], int fn, int fm, int result[]);

int menu(void);

int accept(int fa[], int fn)

{

for (int i = 0; i < fn; i++)

{

printf("\nEnter element %d: ", i + 1);

scanf("%d", &fa[i]);

}

return 0;

}

int display(int fa[], int fn)

{

for (int i = 0; i < fn; i++)

{

printf("%d ", fa[i]);

}

return 0;

}

void display\_result(int arr[], int count)

{

if (count == 0)

{

printf("NULL");

}

else

{

display(arr, count);

}

printf("\n");

}

int set\_union(int fa[], int fb[], int fn, int fm, int result[])

{

int count = 0;

int flag=0;

int i=0;

int j=0;

for (int i = 0; i < fn; i++)

{

result[count] = fa[i];

count++;

}

for (j=0;j<fm;j++)

{

for (i=0;i<fn;i++)

{

if (fa[i]==fb[j])

{

flag=1;

break;

}

}

if (flag==0)

{

result[count]=fb[j];

count++;

}

flag=0;

}

return count;

}

int set\_intersection(int fa[], int fb[], int fn, int fm, int result[])

{

int count = 0;

int flag=0;

int i=0;

int j=0;

for (j=0;j<fm;j++)

{

for (i=0;i<fn;i++)

{

if (fa[i]==fb[j])

{

flag=1;

break;

}

}

if (flag==1)

{

result[count]=fb[j];

count++;

}

flag=0;

}

return count;

}

int set\_difference(int fa[], int fb[], int fn, int fm, int result[])

{

int count = 0;

for (int i = 0; i < fn; i++)

{

int found = 0;

for (int j = 0; j < fm; j++)

{

if (fa[i] == fb[j])

{

found = 1;

break;

}

}

if (found==0)

{

result[count] = fa[i];

count++;

}

}

return count;

}

int menu(void)

{

int choice = 0;

do

{

printf("\nMENU OPERATIONS:\n");

printf("1. Union\n");

printf("2. Intersection\n");

printf("3. Set Difference\n");

printf("4. Symmetric Difference\n");

printf("0. Exit\n");

printf("What operation would you like to perform? ");

scanf("%d", &choice);

} while (choice < 0 || choice > 4);

return choice;

}

int main(void)

{

int n=0;

int m=0;

int choice=0;

printf("\nEnter how many elements you want in array A? ");

scanf("%d", &n);

int a[n];

accept(a, n);

printf("The set A is: ");

display(a, n);

printf("\n");

printf("\nEnter how many elements you want in array B? ");

scanf("%d", &m);

int b[m];

accept(b, m);

printf("The set B is: ");

display(b, m);

printf("\n");

int result[n+m];

int temp[n+m];

do

{

choice = menu();

switch (choice)

{

case 1: // Union

{

int countU = set\_union(a, b, n, m, result);

printf("\nUnion: ");

display\_result(result, countU);

break;

}

case 2: // Intersection

{

int countI = set\_intersection(a, b, n, m, result);

printf("\nIntersection: ");

display\_result(result, countI);

break;

}

case 3: // Set Difference

{

printf("\nA - B: ");

int countAB = set\_difference(a, b, n, m, result);

display\_result(result, countAB);

printf("B - A: ");

int countBA = set\_difference(b, a, m, n, result);

display\_result(result, countBA);

break;

}

case 4: // Symmetric Difference

{

int count1 = set\_difference(a, b, n, m, result);

int count2 = set\_difference(b, a, m, n, temp);

int countSD = set\_union(result, temp, count1, count2, result);

printf("\nSymmetric Difference: ");

display\_result(result, countSD);

break;

}

case 0:

printf("\nExiting...\n");

printf("\nThank you!");

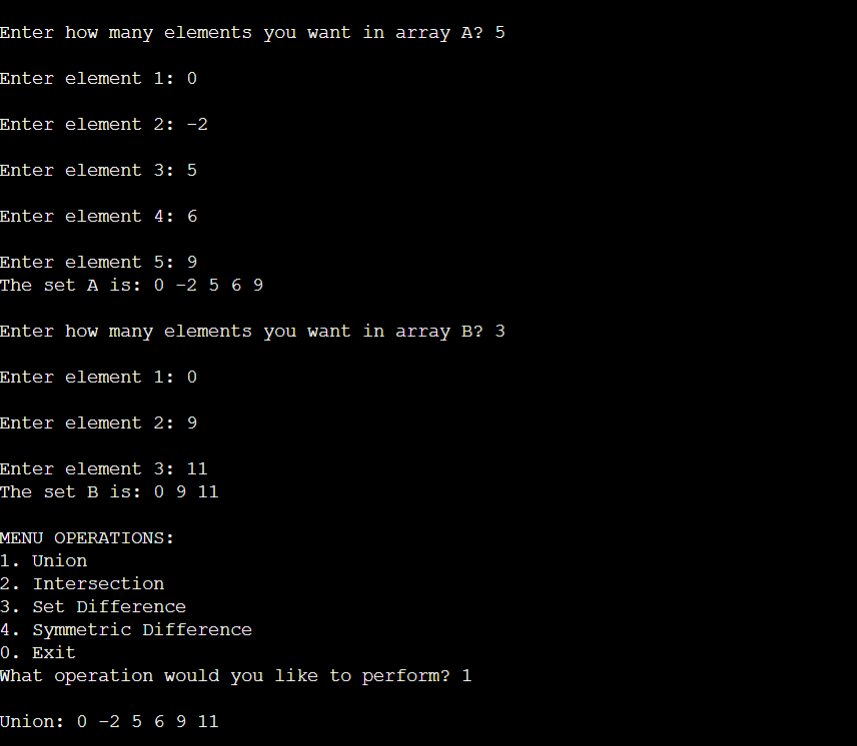
break;

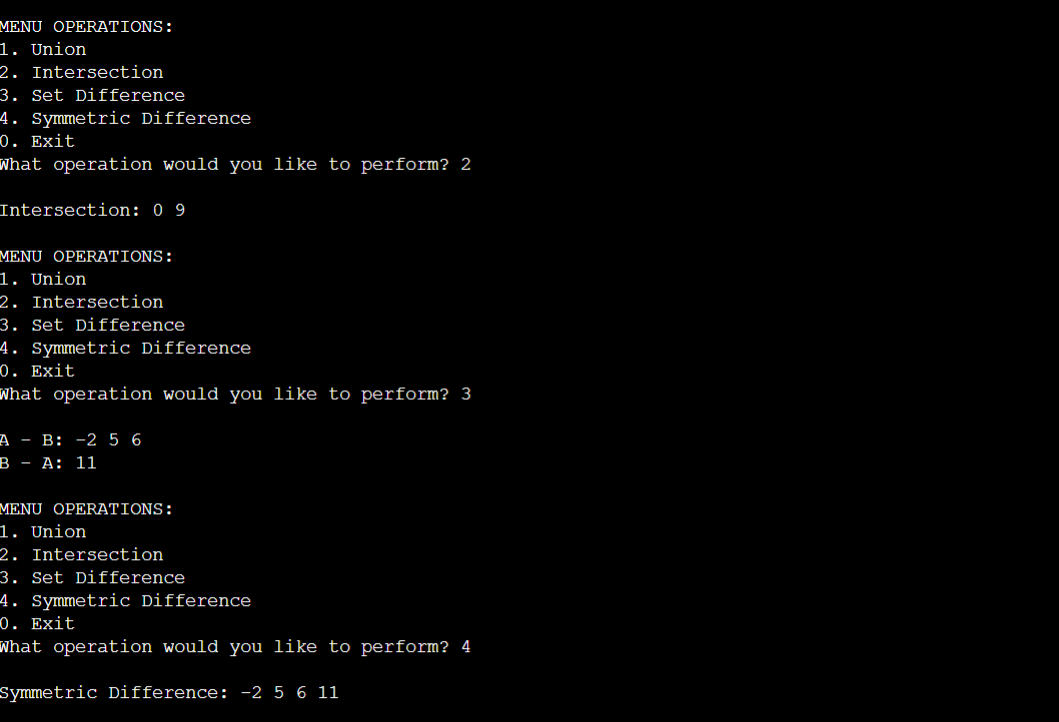
}

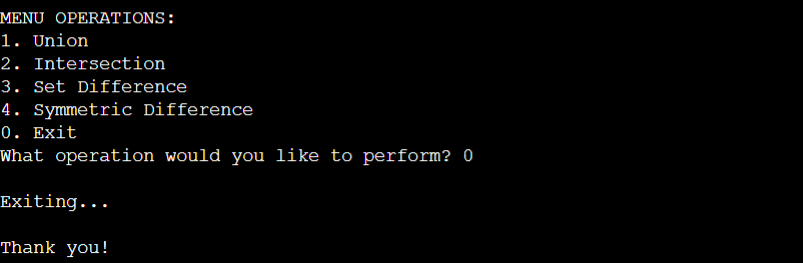
} while (choice != 0);

return 0;

}

**OUTPUT:  
**

****

****