PRACTICAL – 9

1.PRACTICAL STATEMENT –

Implementation of Power Set.

2.OBJECTIVE OF PRACTICAL –

Write a program in C/C++ for implementation of power set.

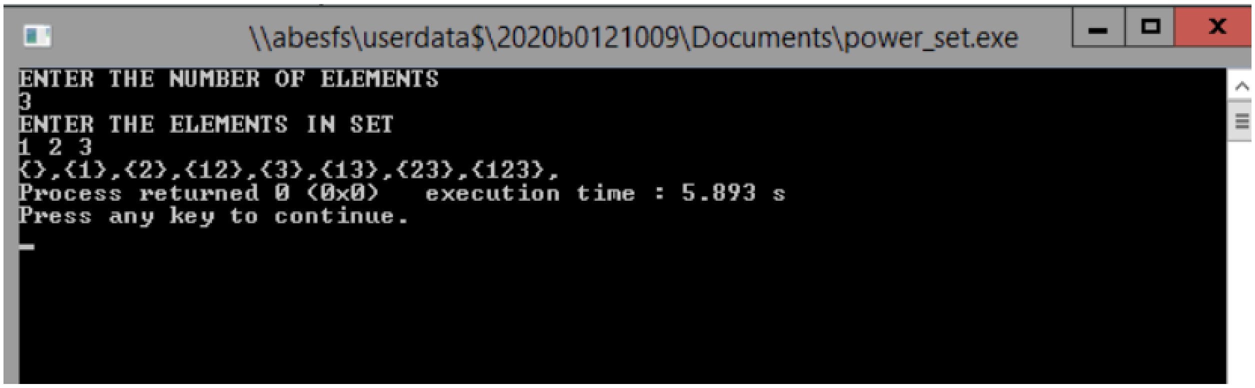
3.ALGORITHM -

```
1.BEGIN:
2.Initialize an array a[n].
3.Read the elements.
4.for i=0 to pow(2,n)
5.for j=0 to n
6. if(i&(1<<j))
7.Print a[j]
8.END
```

4.IMPLEMENTATION -

```
#include<stdio.h>
#include<math.h>
int main()
{
   int i,j,n;
   printf("ENTER THE NUMBER OF ELEMENTS\n");
   scanf("%d",&n);
   int a[n];
   printf("ENTER THE ELEMENTS IN SET\n");
```

```
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    for(i=0;i<pow(2,n);i++)
    {
        printf("{");
        for(j=0;j<n;j++)
        {
            if(i&(1<<j))
            {
                printf("%d",a[j]);
            }
        }
        printf("},");
    }
}
```



```
ENTER THE NUMBER OF ELEMENTS
3
ENTER THE ELEMENTS IN SET
1 2 3
{},{1},{2},{12},{3},{13},{23},{123},
Process returned 0 (0x0)   execution time : 5.893 s
Press any key to continue.
```

PRACTICAL – 10

1.PRACTICAL STATEMENT –

Equivalence Relation

2.OBJECTIVE OF PRACTICAL –

Write a program in C/C++ to check whether a Relation R is Equivalence Relation or not

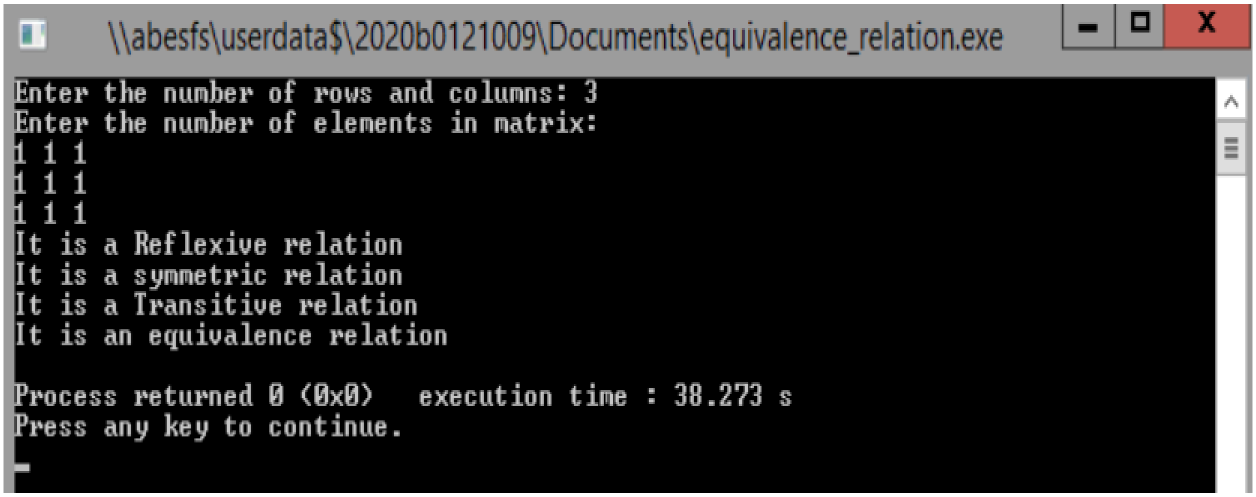
3.ALGORITHM -

1. For Reflexive: Check If Diagonal elements are 1 then it is Reflexive.
2. For Symmetric: If every row coincide column then it is Symmetric Relation.
3. For Transitive: $M_R^2 + M_R = M_R$

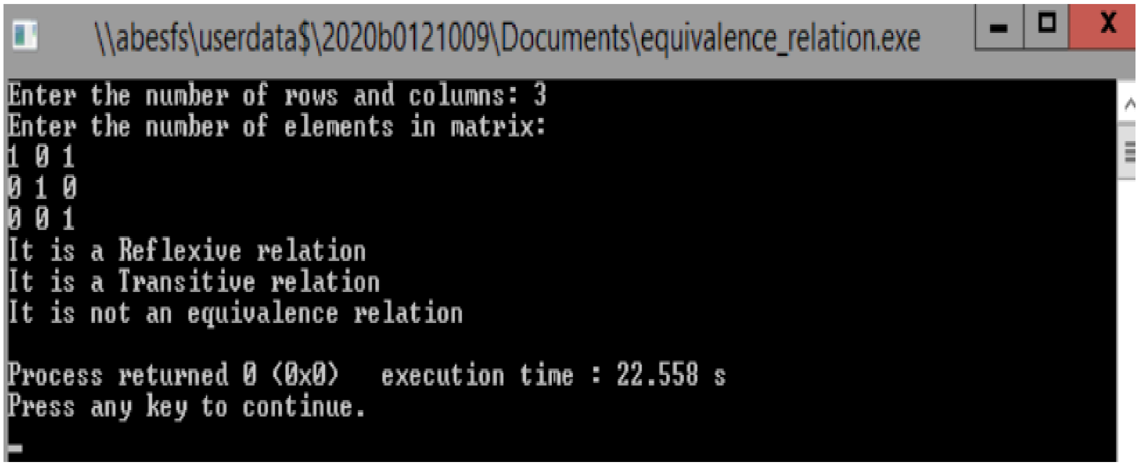
4.IMPLEMENTATION -

```
#include<stdio.h>

main()

{

   int s[50],i,j,n,m[100][100],b[100][100],f,c,flag;

   f=0,c=0;

   flag=0;

   printf("Enter the number of rows and columns:");

   scanf("%d",&n);

   printf("Enter the number of elements in matrix:\n");

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

   {

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

      {

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

      }
}
for(i=0;i<n;i++)
{
    for(j=0;j<n;j++)
    {
        if(i==j && m[i][j]==1)
        {
            f++;
        }
        if(m[i][j]==m[j][i])
        {
            flag++;
        }
        if((m[i][j]&m[i][j]|m[i][j]==m[i][j]))
        {
        c++;
        }
    }
}
if(f==n)
{
    printf("It is a Reflexive relation\n");
}
if(flag==n*n)
```

```
   {
      printf("It is a symmetric relation\n");
   }
   if(c==n*n)
   {
      printf("It is a Transitive relation\n");
   }
   if((f==n)&&(flag==n*n)&&(c==n*n))
   {
      printf("It is an equivalence relation\n");
   }
   else
   {
        printf("It is not an equivalence relation\n");
   }
}
```


\\abesfs\userdata$\2020b0121009\Documents\equivalence_relation.exe

```
Enter the number of rows and columns: 3
Enter the number of elements in matrix:
1 1 1
1 1 1
1 1 1
It is a Reflexive relation
It is a symmetric relation
It is a Transitive relation
It is an equivalence relation

Process returned 0 (0x0)   execution time : 38.273 s
Press any key to continue.
```


\\abesfs\userdata$\2020b0121009\Documents\equivalence_relation.exe

```
Enter the number of rows and columns: 3
Enter the number of elements in matrix:
1 0 1
0 1 0
0 0 1
It is a Reflexive relation
It is a Transitive relation
It is not an equivalence relation

Process returned 0 (0x0)   execution time : 22.558 s
Press any key to continue.
```

PRACTICAL – 11

1.PRACTICAL STATEMENT -

Sum of two numbers without using arithmetic operations.

2.OBJECTIVE OF PRACTICAL –

Write a program in C/C++ to find the Sum of two numbers without using arithmetic operations.

3.ALGORITHM -

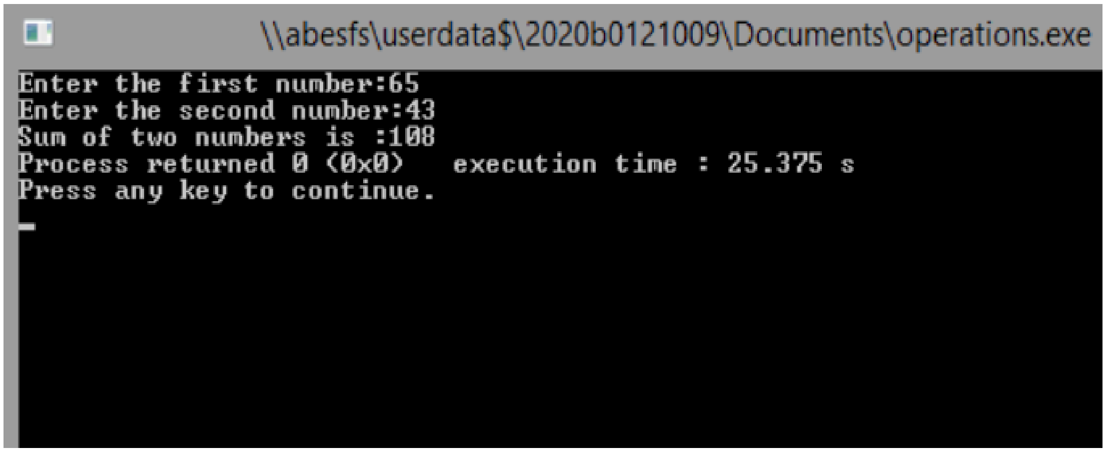
```
 Begin:
     Input two numbers (n1,n2)
     For i=0 to n2
        n1=n1+1
     i incremented by 1
     Print n1 as sum of the numbers
  End;
```

4. IMPLEMENTATION -

```
#include<stdio.h>
#include<stdlib.h>
int add(int ,int);
int main()
{
  int n1,n2;
  printf("Enter the first number:");
  scanf("%d",&n1);
  printf("Enter the second number:");
  scanf("%d",&n2);
```

```
    printf("Sum of two numbers is :%d",add(n1,n2));

    return 0;

}

int add(int num1,int num2){

int i;

for(i=0;i<num2;i++){

    num1++;

}

return num1;

}
```

5. __OUTPUT -__



```
Enter the first number:65
Enter the second number:43
Sum of two numbers is :108
Process returned 0 (0x0)   execution time : 25.375 s
Press any key to continue.
```