Program Code

```
#include<stdio.h>
#include<string.h>


int state, alpha, k;

int check(int p,int unvisited[50])
{
     for(int i=0;i<k;i++)
     {
          if(p==unvisited[i])
               return i;
     }
     return -1;
}

void main()
{

     printf("\nEnter the number of states : ");
     scanf("%d",&state);

     printf("\nEnter the number of alphabets : ");
     scanf("%d",&alpha);

     int alphabet[alpha];

     printf("\nEnter the alphabets : \n");
     for(int i=0;i<alpha;i++)
          scanf("%d",&alphabet[i]);

     int transition[state][alpha];

     printf("\nEnter the transition table : ");

     for(int i=0;i<state;i++)
          for(int j=0;j<alpha;j++)
          {
               printf("\n(q%d,%d)->",i,alphabet[j]);
               scanf("%d",&transition[i][j]);
          }

     int myhill[state][state];

     for(int i=0;i<state;i++)
     {
          for(int j=0;j<state;j++)
               myhill[i][j]=-1;
     }

     printf("\nEnter the number of final states : ");
     int final;
     scanf("%d",&final);

     int fin[final];

     printf("\nEnter the final states : ");
```

```
for(int i=0;i<final;i++)
     scanf("%d",&fin[i]);

for(int i=0;i<final;i++)
{
     for(int j=0;j<state;j++)
     {
          int flag=0;
          for(int k=0;k<final;k++)
          {
               if(j==fin[k])
               {
                    flag=1;
                    break;
               }
          }
          if(flag==0)
          {
               myhill[j][fin[i]] = 1;
               myhill[fin[i]][j] = 1;
          }

     }
}
int c;
do
{
     c=0;
     for(int i=1;i<state;i++)
     {
          for(int j=0;j<i;j++)
          {
               if(myhill[i][j]==-1)
               {
                    for(int k=0;k<alpha;k++)
                    {
                         int a = transition[i][k];
                         int b = transition[j][k];
                         //printf ("\n%d %d %d %d",a,b,i,j);
                         if(myhill[a][b]!=-1)
                         {
                              myhill[i][j]=1;
                              c++;
                              break;
                         }
                    }
               }
          }
     }
}while(c>0);

printf("\nFollowing states can be combined: ");

for(int i=1;i<state;i++)
{
     for(int j=0;j<i;j++)
     {
          if(myhill[i][j]==-1)
          {
               printf("\n\t(q%dq%d)",i,j);
```

```
                }
            }
        }
        printf("\n");

}
```

Output

```
students@pgcse-HP-280-G1-MT:~/Desktop/R7_66/R7_66/1/5$ ./dfa_minimization

Enter the number of states : 6

Enter the number of alphabets : 2

Enter the alphabets :
0 1

Enter the transition table :
(q0,0)->3

(q0,1)->1

(q1,0)->2

(q1,1)->5

(q2,0)->2

(q2,1)->5

(q3,0)->0

(q3,1)->4

(q4,0)->2

(q4,1)->5

(q5,0)->5

(q5,1)->5

Enter the number of final states : 3

Enter the final states : 1 2 4

Following states can be combined:
        (q2q1)
        (q3q0)
        (q4q1)
        (q4q2)
```