```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
int a[20][20],reach[20],n;
void dfs(int v)
{
   int i;
   reach[v]=1;
   for(i=1;i<=n;i++)
   {
      if(a[v][i] && !reach[i])
      {
         printf("\n%d->%d\n",v,i);
         dfs(i);
      }
   }
}
int main()
{
   int i,j,count=0;
   clock_t start,end;
   double tot;
   printf("Enter the limit\n");
   scanf("%d",&n);
   printf("Enter the matrix\n");
   for(i=1;i<=n;i++)
   {
      for(j=1;j<=n;j++)
      {
         scanf("%d",&a[i][j]);
      }
   }
   for(i=1;i<=n;i++)
   {
      reach[i]=0;
   }
   start=clock();
   dfs(1);
   end=clock();
   tot=(double)(end-start)/CLOCKS_PER_SEC;
   for(i=1;i<=n;i++)
   {
      if(reach[i])
         count++;
   }
   if(count==n)
      printf("Graph is connected\n");
   else
      printf("Graph is disconnected\n");
   printf("Execution time in seconds : %f\n",tot);
   return 0;
}
```

OUTPUT:

```
Enter the limit
3
Enter the matrix
0 0 1
1 0 1
1 1 0

1->3

3->2
Graph is connected
Execution time in seconds : 0.000000

Enter the limit
3
Enter the matrix
1 0 0
0 1 0
0 0 1
Graph is disconnected
Execution time in seconds : 0.000000
```