```
#include<stdio.h>
int a[20][20],q[20],visited[20],n,f=-1,r=-1,b[20][20],u[20],visit[20],s;
void bfs(int v)
{
      int i;
      for (i=0;i<n;i++)
      {
            if(a[v][i] != 0 && visited[i] == 0)
            {
                  r=r+1;
                  q[r]=i;
                  visited[i]=1;
                  printf("%d  ",i);
            }
     }
     f=f+1;
           if(f<=r)
            {
                  bfs(q[f]);
                                    }

}
void dfs(int g)
{
      int i;
      for (i=0;i<s;i++)
      {
            if(b[g][i] != 0 && visit[i] == 0)
            {
                  visit[i]=1;
                  printf("%d  ",i);
                  dfs(i);
            }
     }
}
int main()
{
   int v,i,j,g,choice;
   do{
         printf("\nchoices:\n");
         printf("\n1:bfs");
         printf("\n2:dfs\n");
         printf("\n3:exit point\n");
         printf("enter the choice");
```

```
        scanf("%d",&choice);
        switch(choice)
        {
                {

                case 1:
                        printf("\n Enter the number of vertices:");
scanf("%d",&n);
for (i=0;i<n;i++)
{
   visited[i]=0;
}
printf("\n Enter graph data in matrix form:\n");
for (i=0;i<n;i++)
  for (j=0;j<n;j++)
   scanf("%d",&a[i][j]);
printf("\n Enter the starting vertex:");
scanf("%d",&v);
f=r=0;
q[r]=v;
printf("\n BFS traversal is:\n");
visited[v]=1;
printf("%d   ",v);

bfs(v);
if(r != n-1)
   printf("\n BFS is not possible");
   break;
}
case 2:
       {

        printf("\n Enter the number of vertices:");
scanf("%d",&s);
for (i=0;i<s;i++)
{
   visit[i]=0;
}
printf("\n Enter graph data in matrix form:\n");
for (i=0;i<s;i++)
  for (j=0;j<s;j++)
   scanf("%d",&b[i][j]);
printf("\n Enter the starting vertex:");
scanf("%d",&g);
```

```
    printf("\n DFS traversal is:\n");
    visit[g]=1;
    printf("%d   ",g);

    dfs(g);
    break;
                    }
          case 3:
                printf("\nexit point");
                break;
          default:
                 printf("\n please enter a valid choice\n");
                break;
          }
}
while(choice!=3);
}
```