Breadth First Traversal

```
#include<stdio.h>
#include<time.h>
void insertq(int q[], int node, int *f, int *r)
{
 if (*f==-1) && (*r==-1)
  {
   (*f)++, (*r)++, q[*f]=node;
  }
 else
  {
  (*r)++, q[*r]=node;
 }
}
int deleteq (int q[], int *f, int *r)
 {
  int temp;
 temp= q[*f];
 if (*f==*r) *f=*r-1;
  else (*f)++;
  return temp;
  }
void bfs (int n, int adj[][10], int src, int visited[])
 {
 int q[20], f=-1, r=-1, v, i;
 insertq(q, src, &f, &r);
  while ((f<=r) && (f!=-1))
   {
    v= deleteq (q, &f, &r);
    if (visited[v]!=1)
    {
    visited [v]=1;
      printf ("%d", v);
```

```
for(i=1; i<=n; i++)
  if((adj[v][i]==1) && (visited[i]!=1))
  insertq(q, i, &f, &r);
 }
}
Void main()
 {
 int i, j, adj[10][10], src, visited[10];
  clock_t start, end;
  double time;
  printf("enter the no. of. vertices\n");
  scanf("%d", &n);
  printf("enter adjacency matrix\n");
  start=clock()
  for(i=1; i<n; i++)
  {
  visited[i]=0;
  for(j=1; j<n; j++)
   scanf("%d", &adj[i][j]);
  }
  printf("enter starting vertex\n");
  scanf("%d", &src);
 printf("The nodes reachable from src are\n");
  bfs(n, adj, src, visited);
  end=clock();
  time=(double)(end-start)/clocks_per_sec;
  printf("\ntime complexity = %fs", time);
 }
```