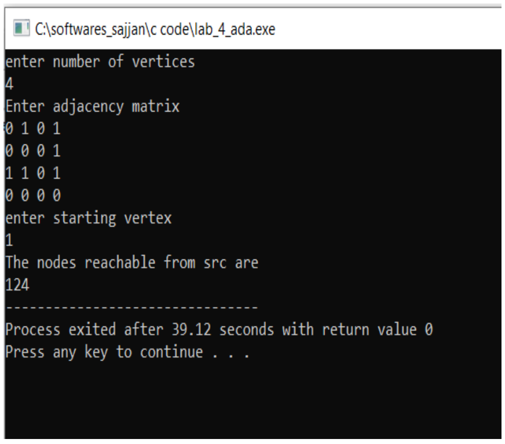
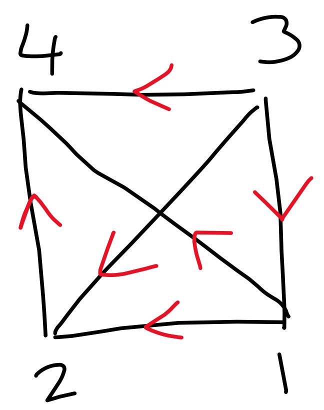
Print all the nodes reachable from a given starting node in a digraph using BFS method.

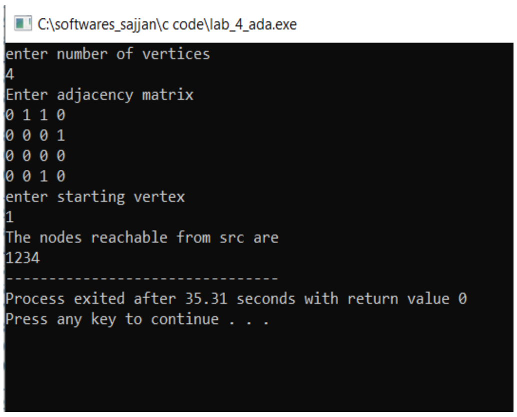
```
#include<stdio.h>
#include<conio.h>

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 n,i,j,adj[10][10],src,visited[10];

clrscr();

printf("enter number of vertices\n");

scanf("%d",&n);

printf("Enter adjacency matrix\n");

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);

getch();
       }
```



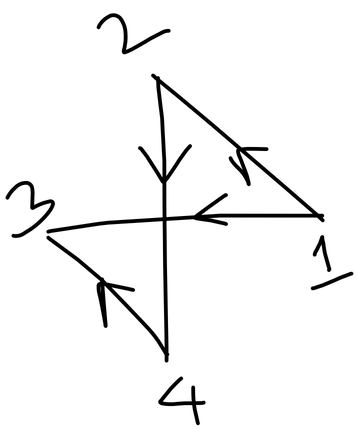
```
C:\softwares_sajjan\c code\lab_4_ada.exe
enter number of vertices
4
Enter adjacency matrix
0 1 0 1
0 0 0 1
1 1 0 1
0 0 0 0
enter starting vertex
1
The nodes reachable from src are
124
--------------------------------
Process exited after 39.12 seconds with return value 0
Press any key to continue . . .
```



$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 1 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 \\ 3 & 1 & 1 & 0 & 1 \\ 4 & 0 & 0 & 0 & 0 \end{array}$$



```
C:\softwares_sajjan\c code\lab_4_ada.exe
enter number of vertices
4
Enter adjacency matrix
0 1 1 0
0 0 0 1
0 0 0 0
0 0 1 0
enter starting vertex
1
The nodes reachable from src are
1234
--------------------------------
Process exited after 35.31 seconds with return value 0
Press any key to continue . . .
```

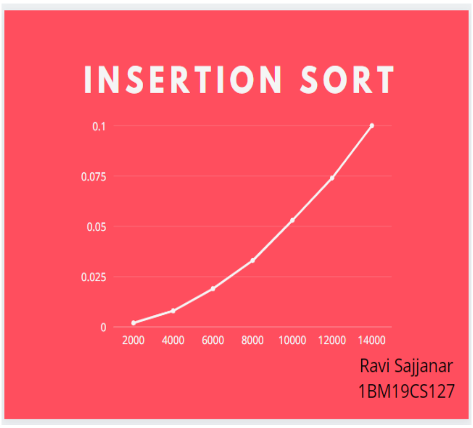


$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 1 & 1 & 0 \\ 2 & 0 & 0 & 0 & 1 \\ 3 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 1 & 0 \end{array}$$

Sort a given set of N integer elements using Insertion Sort technique and compute its time taken.

```
#include<stdio.h>
#include <stdlib.h>
#include<time.h>

int main(){
  int last,arr[5000];
  clock_t end,start;
printf("Enter the Size of array  :");
scanf("%d",&last);
srand(time(NULL));
for(int i=0;i<last;i++){
  arr[i]=rand()%100;
  printf("%d\t",arr[i]);
}printf("\n");
start=clock();
for(int i=1;i<=last-1;i++){
  int key=arr[i];
  int j=i-1;
  while (j >= 0 && arr[j] > key) {
            arr[j + 1] = arr[j];
            j = j - 1;
        }
        arr[j + 1] = key;
}end=clock();
for(int i=0;i<last;i++){
  printf("%d\t",arr[i]);
}
printf("Time in sec %f\n",(((double)(end-start))/CLOCKS_PER_SEC));
}
```

Array Size	Time (in Sec)
2000	0.002
4000	0.008
6000	0.019
8000	0.033
10000	0.053
12000	0.074
14000	0.100


INSERTION SORT
Ravi Sajjanar
1BM19CS127