```
//insert and delete element code
// https://www.geeksforgeeks.org/c-program-to-insert-an-element-in-an-array/
#include <stdio.h>

int main() {
   int arr[10], pos,n=5,x;
   for(int i=1;i<=n;i++)
   {
      arr[i]=i;
      printf("%d ",arr[i]);
   }
   printf("\n");
   pos=2;

   n--;
   for(int i=pos;i<=n;i++){
      arr[i]=arr[i+1];
   }

   for(int i=1;i<=n;i++)
   {
      printf("%d ",arr[i]);
   }
   return 0;
}
//insert and delete element code
// linear search
#include <stdio.h>
```

```
int main() {
   int arr[10], pos,n=5,x;
   for(int i=1;i<=n;i++)
   {
      arr[i]=i;
      printf("%d ",arr[i]);
   }
   x=9;
  pos=0;
   for(int i=1;i<=n;i++)
   {
      if(arr[i]==x){
         pos=1;
         printf("item found at %d\n",i);
         break;
      }

   }
   if(pos==0)printf("item not found\n");
   return 0;
}
/* Bubble sort code */
#include <stdio.h>

int main()
{
  int array[100], n, c, d, swap;

  printf("Enter number of elements\n");
```

```
scanf("%d", &n);

printf("Enter %d integers\n", n);

for (c = 0; c < n; c++)
  scanf("%d", &array[c]);

for (c = 0 ; c < n - 1; c++)
{
    int flag=1;
  for (d = 0 ; d < n - c - 1; d++)
  {
    if (array[d] > array[d+1]) /* For decreasing order use '<' instead of '>' */
    {
      flag=0;
      swap       = array[d];
      array[d]   = array[d+1];
      array[d+1] = swap;
    }
   if(flag==1)break;
  }

}

printf("Sorted list in ascending order:\n");

for (c = 0; c < n; c++)
   printf("%d\n", array[c]);
```

```
  return 0;
}
// stack
#include<stdio.h>

int stack[10],top=-1;
void push(int e){
   if(top>=10)printf("Full\n");
   else{
   stack[++top]=e;
   }
}
int pop(){
   int e=stack[top];
   printf("poped element=%d\n",stack[top]);
   top--;
   return e;
}
void display()
{
   if(top<0)printf("Empty\n");
   else{
   for(int i=0;i<=top;i++)
   printf("%d ",stack[i]);
   printf("\n");
   }
}
int main()
{
```

```
    push(10);
    display();
    int e=pop();
    display();
    return 0;
}

// queue
#include<stdio.h>

int queue[10],font=-1,rear=-1;
void enqueue(int e){
    if(rear>=10)rear=rear%10;
    queue[++rear]=e;
}
int dequeue(){
    if(font>=10)font=font%10;
    int e=queue[++font];
    printf("poped element=%d\n",queue[++font]);
    return e;
}

int main()
{
   enqueue(10);
    enqueue(20);
    for(int i=0;i<=rear;i++)printf("%d ",queue[i]);
    return 0;
}
```

BFS code:

```
// Online C compiler to run C program online
#include <stdio.h>
#define max 5
int matrix[max][max];
int queue[max];
int visited[max];
int level[max];
int f=0,r=0;
// intialization
void intia()
{
   for(int i=0;i<max;i++)
   {
      for(int j=0;j<max;j++)
         matrix[i][j]=0;
   }
}

void print()
{
   for(int i=0;i<max;i++){
      for(int j=0;j<max;j++)
         printf("%d ",matrix[i][j]);
      printf("\n");
   }
}
void enqueue(int a){
   queue[r++]=a;
}
int dequeue(){
   return queue[f++];
}
int main() {
   int edge=7,a,b,i=0;
   f=0;r=0;
   intia();
   while(i<edge)
   {
      scanf("%d%d",&a,&b);
      matrix[a][b]=1;
      matrix[b][a]=1;
      i++;
   }
   //BFS
   for(int i=0;i<max;i++){
      visited[i]=0;
      level[i]=0;
   }
   int s=4;f=0;r=0;
   enqueue(s);
   visited[s]=1;
```

```
    while(f!=r)
    {
        int u=dequeue();

        for(int i=0;i<max;i++){
            if(visited[i]==0&&matrix[u][i]==1){
                printf("enqueue=%d\n",i);
                enqueue(i);
                visited[i]=1;
                level[i]=level[u]+1;
            }
        }


    }

    for(i=0;i<max;i++)
        if(visited[i]==1)
            printf("%d\n",level[i]);

    return 0;
}
```

//MST

Prim's Code:

```
// Online C compiler to run C program online
#include <stdio.h>
#define ver 5
int matrix[ver][ver];
void init()
{
    for(int i=0;i<ver;i++)
        for(int j=0;j<ver;j++)
            matrix[i][j]=0;
}
void print()
{
    for(int i=0;i<ver;i++){
        for(int j=0;j<ver;j++){
           printf("%d ",matrix[i][j]);
        }
        printf("\n");
    }
}

int main() {
    int edge,ver1,ver2,cost;
    init();
    printf("Enter the number of edge: ");
    scanf("%d",&edge);
    for(int i=1;i<=edge;i++)
    {   printf("Enter the edge pair(n1,n2) with cost: ");
```

```
        scanf("%d%d%d",&ver1,&ver2,&cost) ;
        matrix[ver1][ver2]=cost;
        matrix[ver2][ver1]=cost;
    }
    print();
    int visit[ver]={0};
    visit[0]=1;
    edge=0;
    cost=0;
    while(edge<ver-1)
    {
        int min=9999;
        int x=-1,y=-1;
        //choice a edge with minimum cost
        for(int i=0;i<ver;i++)
        {
            if(visit[i]){
                for(int j=0;j<ver;j++){
                    if(matrix[i][j]&&visit[j]==0){
                        if(matrix[i][j]<min){
                            x=i;
                            y=j;
                            min=matrix[i][j];
                        }
                    }
                }
            }
        }
        visit[y]=1;
        printf("%d %d\n",x,y);
        edge++;
        cost+=matrix[x][y];
    }
    printf("MST cost: %d\n",cost);
    return 0;
}
```