LAB ASSIGNMENT	
NAME: Aashirwad Kumar	**Roll NUMBER: IMH/10004/18**

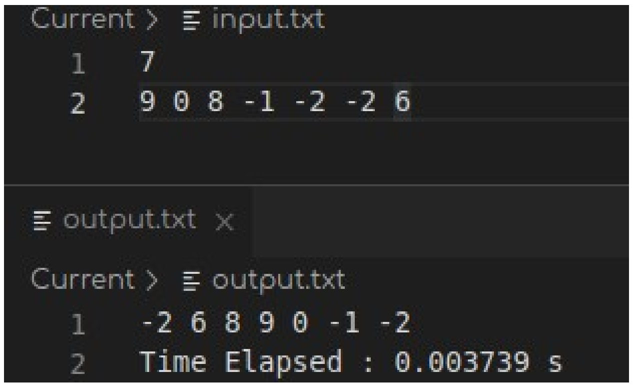
Question 1	Given an array of integers. The task is to rearrange elements of the array such that no two adjacent elements in the array are same.	Date : **04/06/2021**

Write the code with proper indentation

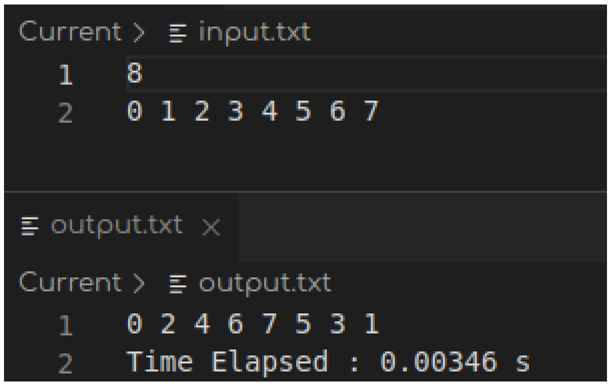
```
#include <bits/stdc++.h>
using namespace std;
#define fast ios::sync_with_stdio(0);cin.tie(0);cout.tie(0);
typedef long long ll;typedef long double ld;typedef pair<int,int> pii;
#define F first
#define S second
#define PB push_back
#define MP make_pair



void solve(){
   int n;int i;int j;
       cin>>n;
   int a[n];
   for(i=0;i<n;i++){
      cin>>a[i];
   }
   if((n)%2==0){
      i=1;
      j=floor(n/2);
      int k=j;
      for (i=1;i<=floor(n/2);i=i+2){
      swap(a[i],a[j]);
      j=j+2;
      if(j>=n){break;}
       }
       sort(a,a+k);
       sort(a+k,a+n,greater<int>());


   }
   else{
      i=1;
      j=floor(n/2)+1;
      int k=j;
      for (i=1;i<=floor(n/2);i=i+2){
      swap(a[i],a[j]);
      j=j+2;
      if(j>=n){break;}
       }
       sort(a,a+k);
```

```
        sort(a+k,a+n,greater<int>());
    }



        for(i=0;i<n;i++){
            cout<<a[i]<<" ";
        }
}




int main(){
    fast;
    int t = 1;

    while(t--){
         solve();
    }

    #ifndef ONLINE_JUDGE
       cout<<"\nTime Elapsed : " << 1.0*clock() / CLOCKS_PER_SEC << " s\n";
    #endif

    return 0;
}
```

Take a screenshot of your output and show here



```
Current > input.txt
7
9 0 8 -1 -2 -2 6

output.txt
Current > output.txt
-2 6 8 9 0 -1 -2
Time Elapsed : 0.003739 s
```



```
Current > input.txt
8
0 1 2 3 4 5 6 7

output.txt
Current > output.txt
0 2 4 6 7 5 3 1
Time Elapsed : 0.00346 s
```

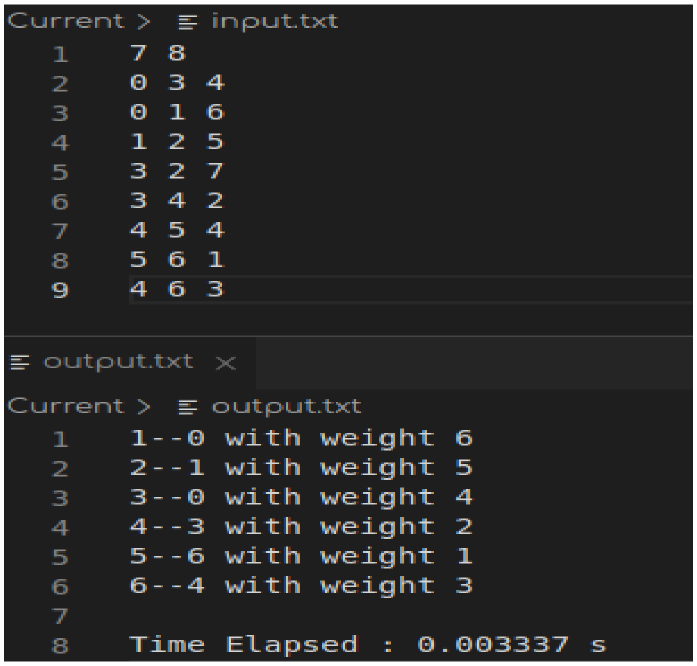
Question 2	Given a connected and undirected graph, find a minimum spanning tree that has minimum cost.	Date : 04/06/2021

Write the code with proper indentation

```
 #include <bits/stdc++.h>
using namespace std;
#define fast ios::sync_with_stdio(0);cin.tie(0);cout.tie(0);
typedef long long ll;typedef long double ld;typedef pair<int,int> pii;
#define F first
#define S second
#define PB push_back
#define MP make_pair



class Graph{
 int V;
 list<pair<int,int>>*l;

public :

Graph(int v){
     V=v;
     l=new list<pair<int,int>>[V];
 }
 void addEdge(int u,int v,int w){
    l[u].push_back(make_pair(w,v));
    l[v].push_back(make_pair(w,u));


 }

 int find(bool* visited , int* weight ,int v){
     int minvertex=-1;
     for(int i=0;i<v;i++){
         if(!visited[i]&&(minvertex==-1||weight[i]<weight[minvertex]))
         {
             minvertex=i;
         }
     }
     return minvertex;
 }

 void prim(){
     bool* visited = new bool[V];
     int *parent =new int [V];
     int* weight =new int[V];

    for(int i=0;i<V;i++){
        weight[i]=INT_MAX;
        visited[i]=false;
```

```
   }

   parent[0]=-1;
   weight[0]=0;

   for(int i=0;i<V-1;i++){
      int minvertex=find(visited,weight,V);
      visited[minvertex]=true;

      for(auto n:l[minvertex]){
         if(!visited[n.second]){
            if(weight[n.second]>n.first){
               weight[n.second]=n.first;
               parent[n.second]=minvertex;
            }
         }
      }

   }

for(int i=1;i<V;i++)
      {
         cout<<i<<"--"<<parent[i]<<" with weight "<<weight[i]<<"\n";
      }


 }


};




int main(){
   fast;

   int n,m;
   cin>>n>>m;


   Graph g(n);
   for(int i=0;i<m;i++){
      int x,y,w;
      cin>>x>>y>>w;
      g.addEdge(x,y,w);
   }
   g.prim();


   #ifndef ONLINE_JUDGE
      cout<<"\nTime Elapsed : " << 1.0*clock() / CLOCKS_PER_SEC << " s\n";
   #endif

   return 0;
}
```

Take a **screenshot** **of your output and show here**


Current > input.txt

```
7 8
0 3 4
0 1 6
1 2 5
3 2 7
3 4 2
4 5 4
5 6 1
4 6 3
```

output.txt

Current > output.txt

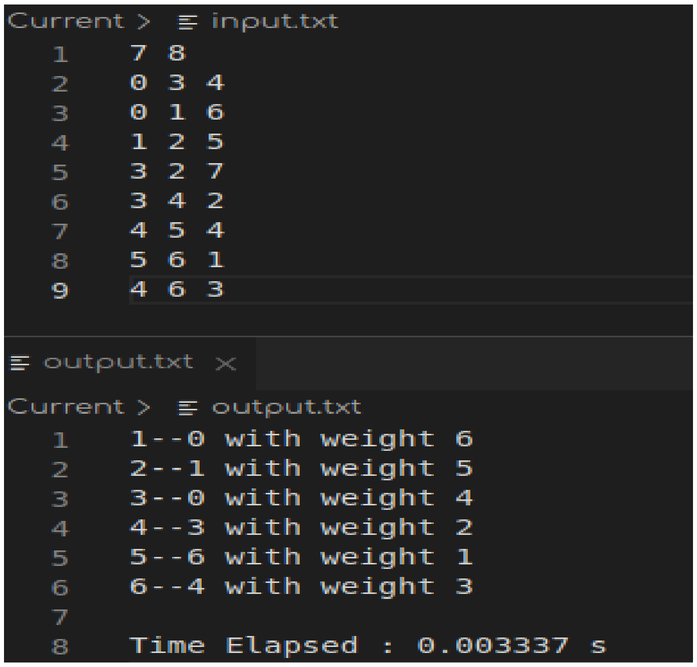
```
1--0 with weight 6
2--1 with weight 5
3--0 with weight 4
4--3 with weight 2
5--6 with weight 1
6--4 with weight 3

Time Elapsed : 0.003337 s
```

Question 3	Given a weighted undirected graph. Finds a subset of the edges that forms a tree that includes every vertex, where the total weight of all the edges in the tree is minimized by using prims approach.	Date : 04/06/2021

Write the code with proper indentation

```
#include <bits/stdc++.h>
using namespace std;
#define fast ios::sync_with_stdio(0);cin.tie(0);cout.tie(0);
typedef long long ll;typedef long double ld;typedef pair<int,int> pii;
#define F first
#define S second
#define PB push_back
#define MP make_pair



class Graph{
 int V;
 list<pair<int,int>>*l;

public :

Graph(int v){
    V=v;
    l=new list<pair<int,int>>[V];
 }
 void addEdge(int u,int v,int w){
   l[u].push_back(make_pair(w,v));
   l[v].push_back(make_pair(w,u));


 }

 int find(bool* visited , int* weight ,int v){
    int minvertex=-1;
    for(int i=0;i<v;i++){
        if(!visited[i]&&(minvertex==-1||weight[i]<weight[minvertex]))
        {
            minvertex=i;
        }
    }
    return minvertex;
 }

 void prim(){
    bool* visited = new bool[V];
    int *parent =new int [V];
    int* weight =new int[V];
   for(int i=0;i<V;i++){
       weight[i]=INT_MAX;
       visited[i]=false;
   }

   parent[0]=-1;
   weight[0]=0;

   for(int i=0;i<V-1;i++){
       int minvertex=find(visited,weight,V);
       visited[minvertex]=true;

       for(auto n:l[minvertex]){
           if(!visited[n.second]){
               if(weight[n.second]>n.first){
                   weight[n.second]=n.first;
                   parent[n.second]=minvertex;
               }
           }
       }

   }

for(int i=1;i<V;i++)
       {
           cout<<i<<"--"<<parent[i]<<" with weight "<<weight[i]<<"\n";
       }


 }


};




int main(){
   fast;

   int n,m;
   cin>>n>>m;


   Graph g(n);
   for(int i=0;i<m;i++){
       int x,y,w;
       cin>>x>>y>>w;
       g.addEdge(x,y,w);
   }
   g.prim();


   #ifndef ONLINE_JUDGE
       cout<<"\nTime Elapsed : " << 1.0*clock() / CLOCKS_PER_SEC << " s\n";
   #endif
   return 0;
}
```

Take a **screenshot** **of your output and show here**



Current > input.txt

```
7 8
0 3 4
0 1 6
1 2 5
3 2 7
3 4 2
4 5 4
5 6 1
4 6 3
```

output.txt ×

Current > output.txt

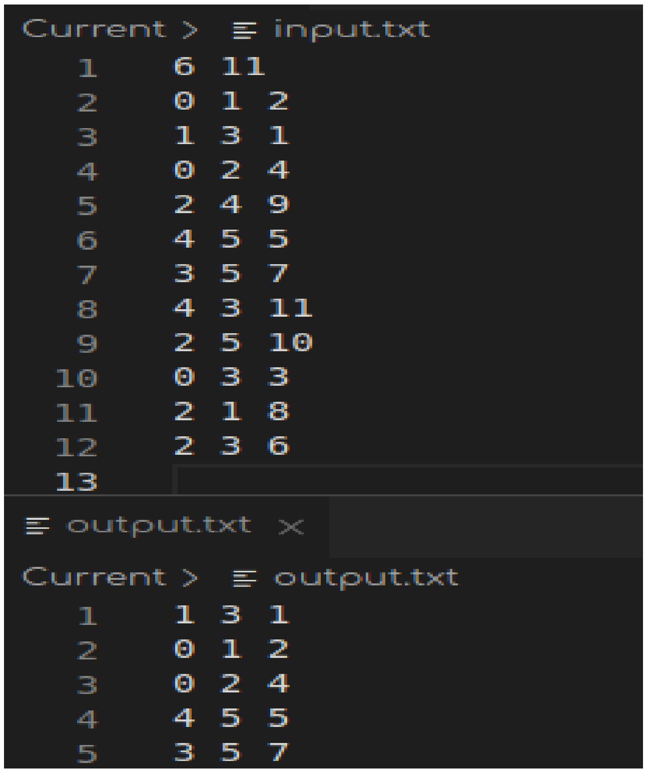
```
1--0 with weight 6
2--1 with weight 5
3--0 with weight 4
4--3 with weight 2
5--6 with weight 1
6--4 with weight 3

Time Elapsed : 0.003337 s
```

Question 4	Find Minimum Cost Spanning Tree of a given connected undirected graph using Kruskal's algorithm. Use Union-Find algorithms in your program.	Date : **04/06/2021**

Write the code with proper indentation

```
#include <bits/stdc++.h>
using namespace std;
#define fast ios::sync_with_stdio(0);cin.tie(0);cout.tie(0);
typedef long long ll;typedef long double ld;typedef pair<int,int> pii;
#define F first
#define S second
#define PB push_back
#define MP make_pair
const ll mod = 1e9+7, N = 2e6+7, M = 2e6+7, INF = INT_MAX/10;
ll powe(ll x, ll y){ x = x%mod, y=y%(mod-1);ll ans = 1;while(y>0){if (y&1){ans = (1ll
* x * ans)%mod;}y>>=1;x = (1ll * x * x)%mod;}return ans;}




class Edge{
   public:
   int src;
   int dest;
   int weight;

};

int findParent(int v,int *parent){
   if(parent[v]==v){
      return v;
   }

   return findParent(parent[v],parent);
}



bool compare(Edge e1, Edge e2){
   return e1.weight<e2.weight;
}
void Kruskals(Edge *input,int n,int E){
   sort(input,input+E,compare);
   Edge *output = new Edge[n-1];


int *parent =new int[n];
for(int i=0;i<n;i++){
   parent[i]=i;
}


   int count =0;
   int i=0;
   while(count!=n-1){
      Edge currentEdge=input[i];
      int sourceParent=findParent(currentEdge.src ,parent);
      int destParent=findParent(currentEdge.dest ,parent);

      if(sourceParent != destParent){
         output[count]=currentEdge;
         count++;
         parent[sourceParent]=destParent;
      }
      i++;
   }

   for(int i=0;i<n-1;i++){

      cout<<output[i].src<<" "<<output[i].dest<<" "<<output[i].weight<<endl;

   }

}


int main(){
   fast;
   int n,E;
   cin>>n>>E;
   Edge *input =new Edge[E];

   for(int i=0;i<E;i++){
      int s,d,w;
      cin>>s>>d>>w;
      input[i].src=s;
      input[i].dest=d;
      input[i].weight=w;

   }


Kruskals(input,n,E);


   return 0;
}
```

Take a screenshot of your output and show here



```
Current > input.txt
6 11
0 1 2
1 3 1
0 2 4
2 4 9
4 5 5
3 5 7
4 3 11
2 5 10
0 3 3
2 1 8
2 3 6

output.txt ×
Current > output.txt
1 3 1
0 1 2
0 2 4
4 5 5
3 5 7
```

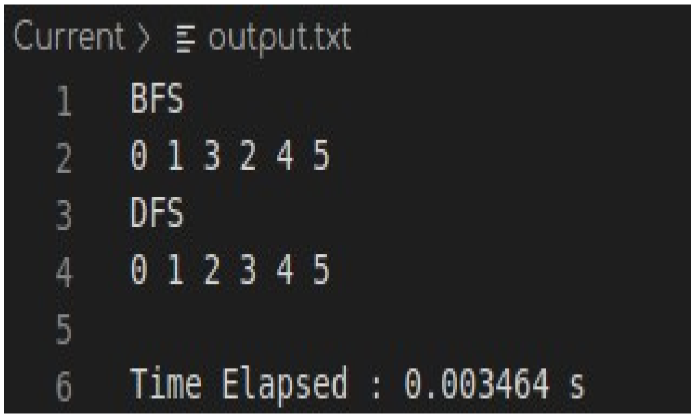
Question 5	Write a program to implement BFS and DFS	Date : 01/06/2021

Write the code with proper indentation

```
#include <bits/stdc++.h>
using namespace std;
#define fast ios::sync_with_stdio(0);cin.tie(0);cout.tie(0);
typedef long long ll;typedef long double ld;typedef pair<int,int> pii;
#define F first
#define S second
#define PB push_back
#define MP make_pair
```

```
class Graph{
 int V;
 list<int> *adjList;

public :

Graph(int v){
     V=v;
     adjList=new list<int>[V];
 }
 void addEdge(int u,int v,bool bidir=true){
     adjList[u].push_back(v);
     if(bidir){
         adjList[v].push_back(u);
     }
 }



void printAdjList(){
     for(int i=0;i<V;i++){
         cout<<i<<" ->";
         for(int node:adjList[i]){
             cout<<node<<",";

         }
         cout<<endl;
     }
 }

 void bfs(int src){
    queue<int>q;
    bool *visited=new bool[V+1]{0};


    q.push(src);
    visited[src]=true;
    while(!q.empty()){
        int node = q.front();
        cout<<node<<" ";
        q.pop();
        for(int n:adjList[node]){
            if(!visited[n]){
                q.push(n);
                visited[n]=true;

            }
        }
    }
    cout<<endl;

}
void dfs_helper(int src,bool* visited){
cout<<src<<" ";
visited[src]=true;
for(int n:adjList[src] ){
    if(!visited[n]){
        dfs_helper(n,visited);

    }
}
}

void dfs(int src){
    bool *visited = new bool[V+1]{0};

    dfs_helper(src,visited);
}

};




int main(){
    fast;

    Graph g(6);
    g.addEdge(0,1);
    g.addEdge(1,2);

    g.addEdge(2,3);

    g.addEdge(3,4);
    g.addEdge(4,5);
    g.addEdge(3,0);

    // g.printAdjList();
    cout<<"BFS"<<endl;
    g.bfs(0);
    cout<<"DFS"<<endl;
    g.dfs(0);
    cout<<endl;

    #ifndef ONLINE_JUDGE
        cout<<"\nTime Elapsed : " << 1.0*clock() / CLOCKS_PER_SEC << " s\n";
    #endif

    return 0;
}
```

Take a screenshot of your output and show here



```
Current > output.txt
BFS
0 1 3 2 4 5
DFS
0 1 2 3 4 5

Time Elapsed : 0.003464 s
```

Instruction:

1 **Don’t try to copy and paste the code from each other or from the internet and write all the lab assignment in the above format only.**

2 **After writing all the lab assignments convert the word file to PDF then submit it in the google classroom in the assignment section.**

3 **All the file names must be your roll number in proper format .**