**NAME – ABHIRAJ YOGESH SRIVASTAVA**

**ROLL NO. – 1906137**

**SUBJECT NAME – DESIGN AND ANALYSIS OF ALGORITHMS LAB**

**SUBJECT CODE – CSL4403**

**DATE – 5TH APRIL, 2021**

**BRANCH – CSE 2**

**ASSIGNMENT-13**

**Q13. WAP to implement Hamiltonian Cycle problem using Backtracking.**

**Source Code in C++ Language:**

#include <bits/stdc++.h>

using namespace std;

int n=5;

int g[6][6]={{0,0,0,0,0,0},

{0,0,1,1,0,1},

{0,1,0,1,1,1},

{0,1,1,0,1,0},

{0,0,1,1,0,1},

{0,1,1,0,1,0}};

int x[6]={0,0,0,0,0,0};

void Next(int k)

{

do

{

x[k]=(x[k]+1)%(n+1);

//cout<<k<<" "<<x[k]<<endl;

if(x[k]==0)

return;

if(k==1)

return;

if(g[x[k]][x[k-1]]==1)

{

int j;

for(j=1;j<=k-1;j++)

{

if(x[j]==x[k])

break;

}

if(j==k)

{

if(k<n || (k==n && g[x[n]][x[1]]==1))

return;

}

}

}

while(true);

}

void Hamilton(int k)

{

do

{

Next(k);

if(x[k]==0)

return;

if(k==n)

{

for(int i=1;i<=n;i++)

cout<<x[i]<<" ";

cout<<x[1]<<endl;

}

else

{

//cout<<x[k]<<" ";

Hamilton(k+1);

}

}

while(true);

}

int main()

{

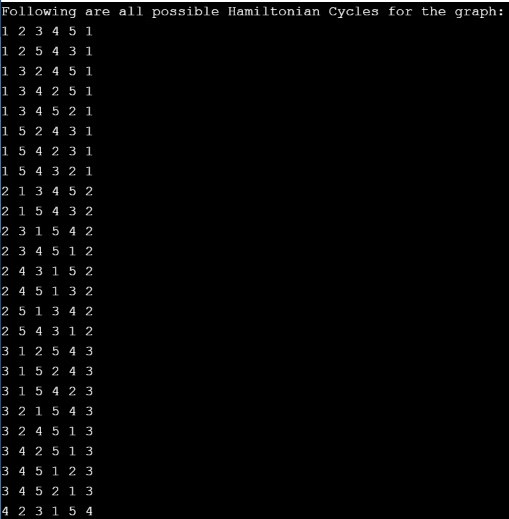
cout<<"Following are all possible Hamiltonian Cycles for the graph:\n";

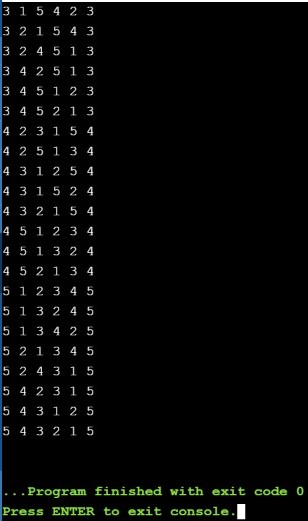
Hamilton(1);

return 0;

}

**Output Screenshots:**

****

****