

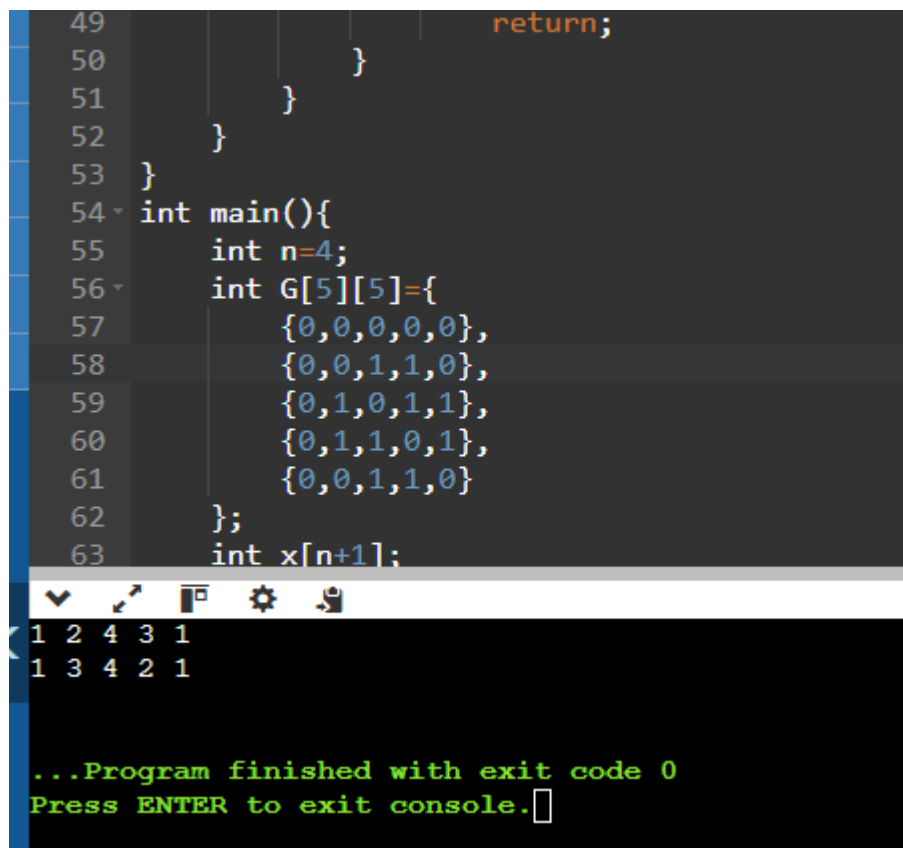
Pract 7  
A8\_b2\_20

```
#include <stdio.h>
void nextValue(int k,int n,int G[n+1][n+1],int x[n+1]);
void hamiltonian(int k,int n,int G[n+1][n+1],int x[n+1])
{
    while(1)
    {
        nextValue(k,n,G,x);
        if(x[k]==0)
            return;
        if(k==n)
        {
            for(int i=1;i<=n;i++)
            {
                printf("%d ",x[i]);
            }
            printf("%d\n",x[1]);
        }
        else
        {
            hamiltonian(k+1,n,G,x);
        }
    }
}
void nextValue(int k,int n,int G[n+1][n+1],int x[n+1]){
    while(1){
        x[k]= (x[k]+1)%(n+1);
        if(x[k]==0){
            return;
        }
        if(G[x[k-1]][x[k]]!=0)
        {
            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]] != 0))
                    return;
            }
        }
    }
}
```

```

}
int main(){
    int n=4;
    int G[5][5]={
        {0,0,0,0,0},
        {0,0,1,1,0},
        {0,1,0,1,1},
        {0,1,1,0,1},
        {0,0,1,1,0}
    };
    int x[n+1];
    for(int i=0;i<=n;i++)
        x[i]=0;
    x[1]=1;
    hamiltonian(2,n,G,x);
    return 0;
}

```



The screenshot shows a C++ IDE with a dark theme. The code is displayed in the editor, and the output is shown in the console below. The code defines a graph with 5 nodes and 10 edges, and a function that finds a Hamiltonian path starting from node 2. The output shows two possible paths: 1 2 4 3 1 and 1 3 4 2 1.

```

49         return;
50     }
51 }
52 }
53 }
54 int main(){
55     int n=4;
56     int G[5][5]={
57         {0,0,0,0,0},
58         {0,0,1,1,0},
59         {0,1,0,1,1},
60         {0,1,1,0,1},
61         {0,0,1,1,0}
62     };
63     int x[n+1];

```

1 2 4 3 1  
1 3 4 2 1

...Program finished with exit code 0  
Press ENTER to exit console.