

DAA – P7

NAME : Dipansh Tiwari

SEC : A4-B3

ROLL NO : 35

CODE :

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

```
#define MAX 20
```

```
int G[MAX][MAX];
```

```
int x[MAX];
```

```
int n;
```

```
void NextValue(int k) {
```

```
int j;
```

```
while (1) {
```

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

```
if (x[k] == 0)
```

```
return;
```

```
if (G[x[k - 1]][x[k]] != 0) {
```

```
for (j = 1; j < k; j++) {
```

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

```
break;
```

}

```

        if (j == k) {
            if ((k < n) || (k == n && G[x[n]][x[1]] != 0))
                return;
        }
    }
}

void Hamiltonian(int k) {
    while (1) {
        NextValue(k);
        if (x[k] == 0)
            return;
        if (k == n) {
            printf("Hamiltonian Cycle: ");
            for (int i = 1; i <= n; i++)
                printf("%d ", x[i]);
            printf("%d\n", x[1]);
        } else {
            Hamiltonian(k + 1);
        }
    }
}

int main() {
    int i, j;

    printf("Enter number of vertices: ");
    scanf("%d", &n);

```

```

printf("Enter adjacency matrix:\n");
for (i = 1; i <= n; i++) {
    for (j = 1; j <= n; j++) {
        scanf("%d", &G[i][j]);
    }
}

for (i = 1; i <= n; i++)
    x[i] = 0;
x[1] = 1;

printf("\nHamiltonian cycles in the graph:\n");
Hamiltonian(2);

return 0;
}

```

OUTPUT :

Output

Clea

```

Enter number of vertices: 3
Enter adjacency matrix:
2
3
4
5
6
3
2
3
6

Hamiltonian cycles in the graph:
Hamiltonian Cycle: 1 2 3 1
Hamiltonian Cycle: 1 3 2 1

```