

Pract7  
A8\_B2\_18

```
#include <stdio.h>

#define MAX 10

void nextValue(int k, int n, int G[MAX][MAX], int x[MAX]) {
    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, int n, int G[MAX][MAX], int x[MAX]) {
    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]); // to complete the cycle
        } else {
            hamiltonian(k + 1, n, G, x);
        }
    }
}
```

```

    }
}

int main() {
    int n = 4;
    int G[MAX][MAX] = {
        {0, 0, 0, 0, 0},
        {0, 0, 1, 1, 1},
        {0, 1, 0, 1, 0},
        {0, 1, 1, 0, 1},
        {0, 1, 0, 1, 0}
    };

    int x[MAX] = {0};
    x[1] = 1; // fix the first vertex
    printf("Hamiltonian cycles:\n");
    hamiltonian(2, n, G, x);

    return 0;
}

```

```

    if (x[1] == 0)
        return;

    if (k == n) {
        for (int i = 1; i <= n; i++)
            printf("%d ", x[i]);
        printf("%d\n", x[1]); // to complete the cycle
    } else {
        hamiltonian(k + 1, n, G, x);
    }
}

int main() {
    int n = 4;
    int G[MAX][MAX] = {
        {0, 0, 0, 0, 0},
        {0, 0, 1, 1, 1},
        {0, 1, 0, 1, 0},
        {0, 1, 1, 0, 1},
        {0, 1, 0, 1, 0}
    };

    int x[MAX] = {0};
    x[1] = 1; // fix the first vertex
    printf("Hamiltonian cycles:\n");
    hamiltonian(2, n, G, x);
}

```

Hamiltonian cycles:

```

1 2 3 4 1
1 4 3 2 1

```

=== Code Execution Successful ===