5. Floyd algorithm

```c
#include <stdio.h>

int a[10][10], D[10][10], n;
void floyd(int a[][10], int n);
int min(int a, int b);

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

    printf("Enter the cost adjacency matrix:\n");
    int i, j;


    for(i = 0; i < n; i++) {
        for(j = 0; j < n; j++) {
            scanf("%d", &a[i][j]);

            if (a[i][j] == 0 && i != j) {
                a[i][j] = 9999;
            }
        }
    }

    floyd(a, n);

    printf("Distance Matrix:\n");
    for(i = 0; i < n; i++) {
        for(j = 0; j < n; j++) {
            printf("%d ", D[i][j]);
        }
        printf("\n");
    }

    return 0;
}

void floyd(int a[][10], int n) {
    int i, j, k;

    for(i = 0; i < n; i++) {
        for(j = 0; j < n; j++) {
            D[i][j] = a[i][j];
        }
    }


    for(k = 0; k < n; k++) {
        for(i = 0; i < n; i++) {
            for(j = 0; j < n; j++) {
                D[i][j] = min(D[i][j], (D[i][k] + D[k][j]));
            }
        }
    }
}

int min(int a, int b) {
    if(a < b) {
        return a;
    } else {
        return b;
    }
}
```

```
Output
Enter the number of vertices: 4
Enter the cost adjacency matrix:
0 3 9999 7
3 0 5 9999
9999 5 0 1
7 9999 1 0
Distance Matrix:
0 3 8 7
3 0 5 6
8 5 0 1
7 6 1 0
```