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

#define MAX_VERTICES 1000

struct Queue {
    int front, rear, size;
    unsigned capacity;
    int* array;
};

struct Queue* createQueue(unsigned capacity) {
    struct Queue* queue = (struct Queue*)malloc(sizeof(struct Queue));
    queue->capacity = capacity;
    queue->front = queue->size = 0;
    queue->rear = capacity - 1;
    queue->array = (int*)malloc(queue->capacity * sizeof(int));
    return queue;
}

bool isEmpty(struct Queue* queue) {
    return (queue->size == 0);
}

void enqueue(struct Queue* queue, int item) {
    queue->rear = (queue->rear + 1) % queue->capacity;
    queue->array[queue->rear] = item;
    queue->size = queue->size + 1;
}
```

```
int dequeue(struct Queue* queue) {
    int item = queue->array[queue->front];
    queue->front = (queue->front + 1) % queue->capacity;
    queue->size = queue->size - 1;
    return item;
}

int minEdgesBetweenVertices(int u, int v, int N, int graph[][N]) {
    int visited[N];
    int distance[N];
    for (int i = 0; i < N; i++) {
        visited[i] = 0;
        distance[i] = 0;
    }

    struct Queue* q = createQueue(MAX_VERTICES);
    enqueue(q, u);
    visited[u] = 1;

    while (!isEmpty(q)) {
        int current = dequeue(q);

        for (int i = 0; i < N; i++) {
            if (graph[current][i] == 1 && !visited[i]) {
                enqueue(q, i);
                visited[i] = 1;
                distance[i] = distance[current] + 1;

                if (i == v) {
                    return distance[v];
                }
```

```
            }
        }
    }

    return -1; // Return -1 if there is no path between u and v
}

int main() {
    int N = 5; // Number of vertices
    int graph[5][5] = {
        {0, 1, 1, 0, 0},
        {1, 0, 1, 1, 0},
        {1, 1, 0, 0, 1},
        {0, 1, 0, 0, 0},
        {0, 0, 1, 0, 0}
    };

    int u = 0, v = 4; // Source and destination vertices
    int minEdges = minEdgesBetweenVertices(u, v, N, graph);

    if (minEdges == -1) {
        printf("No path exists between %d and %d\n", u, v);
    } else {
        printf("Minimum number of edges between %d and %d is %d\n", u, v, minEdges);
    }

    return 0;
}
```