1. BFS

```
#include <stdio.h>
#include <stdlib.h>
#define MAX_VERTICES 100
void bfs(int graph[MAX_VERTICES][MAX_VERTICES], int vertices, int start) {
  int queue[MAX_VERTICES];
  int visited[MAX_VERTICES] = {0};
  int front = -1, rear = -1;
  queue[++rear] = start;
  visited[start] = 1;
  while (front != rear) {
    int currentVertex = queue[++front];
    printf("%d ", currentVertex);
    for (int i = 0; i < vertices; ++i) {
      if (graph[currentVertex][i] == 1 && !visited[i]) {
        queue[++rear] = i;
        visited[i] = 1;
      }
    }
  }
}
int main() {
  int vertices = 5, edges = 4;
  int graph[MAX_VERTICES][MAX_VERTICES] = {{0, 1, 1, 0, 0},{1, 0, 1, 1, 0},{1, 1, 0, 0, 1},{0, 1, 0, 0, 1},{0, 0, 1, 1, 0}};
  int startVertex = 0;
  printf("BFS traversal starting from vertex %d: ", startVertex);
  bfs(graph, vertices, startVertex);
  return 0;
}
```

2. DFS

```
#include <stdio.h>
#include <stdlib.h>
#define MAX_VERTICES 100
void dfs(int graph[MAX_VERTICES][MAX_VERTICES], int vertices, int currentVertex, int visited[]) {
  printf("%d ", currentVertex);
  visited[currentVertex] = 1;
  for (int i = 0; i < vertices; ++i)
    if (graph[currentVertex][i] == 1 && !visited[i])
      dfs(graph, vertices, i, visited);
}

int main() {
  int vertices = 5, edges = 4;
  int graph[MAX_VERTICES][MAX_VERTICES] = {{0, 1, 1, 0, 0},{1, 0, 1, 1, 0},{1, 1, 0, 0, 1},{0, 1, 0, 0, 1},{0, 0, 1, 1, 0}};
  int startVertex = 0;
  int visited[MAX_VERTICES] = {0};
  printf("DFS traversal starting from vertex %d: ", startVertex);
  dfs(graph, vertices, startVertex, visited);
  return 0;
}
```

Output -

```
PS C:\Users\hello\Documents\SEM3\DSA PCC-CS-391\Lab11> gcc .\38-bfs.c
PS C:\Users\hello\Documents\SEM3\DSA PCC-CS-391\Lab11> .\a.exe
BFS traversal starting from vertex 0: 0 1 2 3 4
PS C:\Users\hello\Documents\SEM3\DSA PCC-CS-391\Lab11> gcc .\39-dfs.c
PS C:\Users\hello\Documents\SEM3\DSA PCC-CS-391\Lab11> .\a.exe
DFS traversal starting from vertex 0: 0 1 2 4 3
PS C:\Users\hello\Documents\SEM3\DSA PCC-CS-391\Lab11>
```

```
Before Sorting:
5       8       1       10      6       3       9       2       4       7
Sorted List (Using Bubble Sort):
1       2       3       4       5       6       7       8       9       10
Sorted List (Using Intertion Sort):
1       2       3       4       5       6       7       8       9       10
Sorted List (Using Selection Sort):
1       2       3       4       5       6       7       8       9       10
Sorted List (Using Quick Sort):
1       2       3       4       5       6       7       8       9       10
Sorted List (Using Merge Sort):
1       2       3       4       5       6       7       8       9       10
PS C:\Users\hello\Documents\SEM3\DSA PCC-CS-391\Lab10>
```