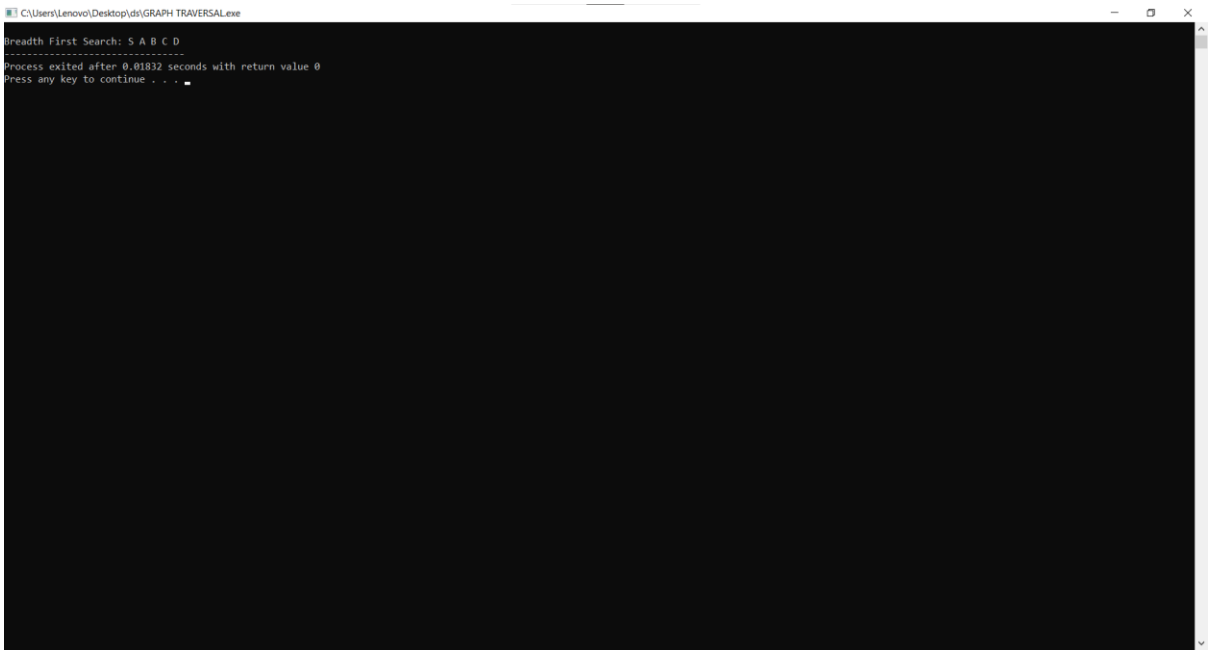
```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#define MAX 5
struct Vertex {
 char label;
 bool visited;
};
int queue[MAX];
int rear = -1;
int front = 0;
int queueItemCount = 0;
struct Vertex* lstVertices[MAX];
int adjMatrix[MAX][MAX];
int vertexCount = 0;
void insert(int data) {
 queue[++rear] = data;
 queueItemCount++;
}
int removeData() {
 queueItemCount--;
 return queue[front++];
}
bool isQueueEmpty() {
 return queueItemCount == 0;
}
void addVertex(char label) {
 struct Vertex* vertex = (struct Vertex*) malloc(sizeof(struct Vertex));
 vertex->label = label;
 vertex->visited = false;
 lstVertices[vertexCount++] = vertex;
```

```
}
void addEdge(int start,int end) {
 adjMatrix[start][end] = 1;
 adjMatrix[end][start] = 1;
}
void displayVertex(int vertexIndex) {
 printf("%c ",lstVertices[vertexIndex]->label);
}
int getAdjUnvisitedVertex(int vertexIndex) {
 int i;

 for(i = 0; i<vertexCount; i++) {
 if(adjMatrix[vertexIndex][i] == 1 && lstVertices[i]->visited == false)
 return i;
 }
 return -1;
}
void breadthFirstSearch() {
 int i;
 lstVertices[0]->visited = true;
 displayVertex(0);
 insert(0);
 int unvisitedVertex;
 while(!isQueueEmpty()) {
 int tempVertex = removeData();
 while((unvisitedVertex = getAdjUnvisitedVertex(tempVertex)) != -1) {
 lstVertices[unvisitedVertex]->visited = true;
 displayVertex(unvisitedVertex);
 insert(unvisitedVertex);
 }
```

```
 }
 for(i = 0;i<vertexCount;i++) {
 lstVertices[i]->visited = false;
 }
}
int main() {
 int i, j;
 for(i = 0; i<MAX; i++){
 for(j = 0; j<MAX; j++)
 adjMatrix[i][j] = 0;
 }
 addVertex('S');
 addVertex('A');
 addVertex('B');
 addVertex('C');
 addVertex('D');
 addEdge(0, 1);
 addEdge(0, 2);
 addEdge(0, 3);
 addEdge(1, 4);
 addEdge(2, 4);
 addEdge(3, 4);
 printf("\nBreadth First Search: ");
 breadthFirstSearch();
 return 0;
}
```


C:\Users\Lenovo\Desktop\ds\GRAPH TRAVERSAL.exe
Breadth First Search: S A B C D

Process exited after 0.01832 seconds with return value 0
Press any key to continue . . .