

# README

---

Greeting! This code sample is an simple implementation of A\* algorithm with visualization on command line.

```
// Display board
const vector<vector<int>> board = {

    {0,0,0,0,0,0,2},
    {0,-1,-1,-1,-1,-1,0},
    {0,0,0,0,0,0,-1},
    {0,-1,0,0,0,-1,-1},
    {-1,-1,0,-1,0,0,0},
    {0,0,0,0,0,-1,-1},
    {1,0,0,0,0,0,0}

};
```

The initial board consists of numbers

- 1: start point
- 2: end point
- -1: blocks
- 0: valid path

By running the program, command line will print out the visualization of the grid board

```

=== Input path finding board ===

*-----*
|      |      |      |      |      |      |*****|
|      |      |      |      |      |      |*****|
*-----*
|      |XXXXX|XXXXX|XXXXX|XXXXX|XXXXX|      |
|      |XXXXX|XXXXX|XXXXX|XXXXX|XXXXX|      |
*-----*
|      |      |      |      |      |      |XXXXX|
|      |      |      |      |      |      |XXXXX|
*-----*
|      |XXXXX|      |      |      |XXXXX|XXXXX|
|      |XXXXX|      |      |      |XXXXX|XXXXX|
*-----*
|XXXXX|XXXXX|      |XXXXX|      |      |      |
|XXXXX|XXXXX|      |XXXXX|      |      |      |
*-----*
|      |      |      |      |      |XXXXX|XXXXX|
|      |      |      |      |      |XXXXX|XXXXX|
*-----*
|00000|      |      |      |      |      |      |
|00000|      |      |      |      |      |      |
*-----*

=== Board legend ===
1) Start point:
                                     *-----*
                                     |00000|
                                     |00000|
                                     *-----*

2) End point:
                                     *-----*
                                     |*****|
                                     |*****|
                                     *-----*

3) Blocks:
                                     *-----*
                                     |XXXXX|
                                     |XXXXX|
                                     *-----*

4) Empty path:
                                     *-----*
                                     |      |
                                     |      |
                                     *-----*

5) Result path:
                                     *-----*
                                     |+++++|
                                     |+++++|
                                     *-----*

```

If the input board is valid, then the `pathFinding()` function will perform A\* algorithm to find the shortest path from start point to end point

This is a sample output:

```
=== RESULT ===: Found a path!
*-----*-----*-----*-----*-----*-----*-----*
|+++++|+++++|+++++|+++++|+++++|+++++|*****|
|+++++|+++++|+++++|+++++|+++++|+++++|*****|
*-----*-----*-----*-----*-----*-----*-----*
|+++++|XXXXX|XXXXX|XXXXX|XXXXX|XXXXX|      |
|+++++|XXXXX|XXXXX|XXXXX|XXXXX|XXXXX|      |
*-----*-----*-----*-----*-----*-----*-----*
|+++++|+++++|+++++|      |      |      |XXXXX|
|+++++|+++++|+++++|      |      |      |XXXXX|
*-----*-----*-----*-----*-----*-----*-----*
|      |XXXXX|+++++|      |      |XXXXX|XXXXX|
|      |XXXXX|+++++|      |      |XXXXX|XXXXX|
*-----*-----*-----*-----*-----*-----*-----*
|XXXXX|XXXXX|+++++|XXXXX|      |      |      |
|XXXXX|XXXXX|+++++|XXXXX|      |      |      |
*-----*-----*-----*-----*-----*-----*-----*
|+++++|+++++|+++++|      |      |XXXXX|XXXXX|
|+++++|+++++|+++++|      |      |XXXXX|XXXXX|
*-----*-----*-----*-----*-----*-----*-----*
|00000|      |      |      |      |      |      |
|00000|      |      |      |      |      |      |
*-----*-----*-----*-----*-----*-----*-----*
```