A program that will allow the player to navigate and explore a dungeon that is described with a level file
A level file will specify the rooms on the first line delimited by a space, following that, each line will describe a room's pathway to another room.

The first room listed on the first line is the starting point of the level.

The level file structure :

```
START FOYER ELEVATOR
START > NORTH > FOYER
FOYER > SOUTH > START
START > WEST > ELEVATOR
```

Each pathway is broken up by spaces and the > symbols. <roomA> > WEST > <roomB> this translates to:

roomA's west's pathway connects to roomB

Annotation for the test level file:

```
START FOYER ELEVATOR //START is the first room of the dungeon.
START > NORTH > FOYER //START's north pathway connects to the FOYER
FOYER > SOUTH > START //FOYER's  south pathway connects to the START
START > WEST > ELEVATOR //START's west pathway connects to the ELEVATOR
```

The **commands** that a player can input:

```
NORTH
SOUTH
EAST
WEST
QUIT
```

**QUIT** command allows the user to quit the program, please ensure that you clean up any memory you have allocated.

The **NORTH, SOUTH, EAST** and **WEST** commands will allow the player to move between the rooms via the room's pathways. Each room has a maximum of 4 pathways.

A room will outline if the paths that are available to it by specifying the direction at the side of the room. For example if the room has a path to another room by going north it will show N on the north side of the room. Example:

```
 ---N---
|       |
|       |
|       |
|       |
|       |
 -------
```

If it does not have a path, it will be a - or | depending on the side it is on.

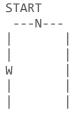If a user specifies a direction that does not have a pathway the program should output:

```
No Path This Way
```

If the user inputs an invalid command the program should respond with:

```
What?
```

The room is always 9 x 5 (with a space at the start and end of the top and bottom of the room);

The name should also be outputted before drawing the room.

```
NORTH

START
  ---N---
 |     |
 |     |
 W     |
 |     |
 |     |
  -------
```

Specific functions to know for this task:

**fopen, fgets, sscanf, malloc, realloc, free, memset, strcpy, strcmp**

**Assumptions/Clarifications:**

* 8192 is largest number of characters in a line from the level file.

* Room name can be variable length.

* Commands are case sensitive

* On every command (even invalid ones) you need to redraw the room

* You may assume the test files are correct.

* Room pathways do **not** have logical entries. If the player moves NORTH you do not need to map the previous room to SOUTH of the current room.

* If a room's pathway has already been assigned it can be overridden when reading the level file.

**Examples:**

Example 1:

```
./dungeon test_dungeon.dg

START
  ---N---
 |     |
 |     |
 W     |
 |     |
 |     |
  -------

NORTH

FOYER
  -------
 |     |
 |     |
```

```
  |       |
  |       |
  |       |
  |       |
  ---S---
```

SOUTH

START
```
  ---N---
  |       |
  |       |
  W       |
  |       |
  |       |
  -------
```

QUIT

Example 2:

./dungeon test_dungeon.dg

START
```
  ---N---
  |       |
  |       |
  W       |
  |       |
  |       |
  -------
```

WEST

ELEVATOR
```
  -------
  |       |
  |       |
  |       |
  |       |
  |       |
  -------
```

NORTH
No Path This Way

ELEVATOR
```
  -------
  |       |
  |       |
  |       |
  |       |
  |       |
  -------
```

EAST
No Path This Way

ELEVATOR
```
  -------
  |       |
  |       |
  |       |
  |       |
```

```
|       |
 -------
```

QUIT

## Example 3:

```
./dungeon
No Level File Specified
```

## Example 4:

```
./dungeon test_dungeon.dg
```

```
START
  ---N---
|       |
|       |
W       |
|       |
|       |
  -------
```

```
ksamdkl
What?
```

```
START
  ---N---
|       |
|       |
W       |
|       |
|       |
  -------
```