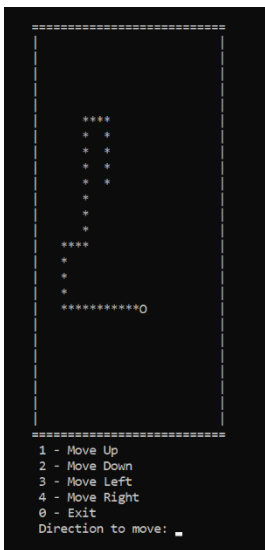


2D Arrays – Lab 2 (Turtle Graphics)

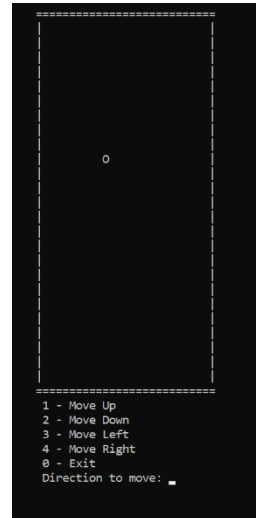
Turtle Graphics is a method for creating graphic images originally in the Logo programming language. The "turtle" is an imaginary pen that is given drawing commands, such as go up, down, left and right. You will be writing a version of turtle graphics using a 2D array to move the turtle around the matrix to draw an image.

Your turtle graphics program will be relatively simple. You will create a 25x25 2D char array. The capital letter 'O' will represent the turtle and lines will be created using the * symbol. As the turtle moves around the screen it will leave the * symbol behind.



When the program begins, the turtle's field is shown with the = symbol marking the top and bottom border of the field, and the | symbol marking the left and right borders. The turtle is shown in its starting position. The starting position can be anywhere on the field.

The user can use the numbers, 8, 2, 4, and 6 to move the turtle around the field to create drawings. These numbers correspond with the arrows on the number pad of your keyboard.



To move the turtle around the screen, you will use the **getch()** method rather than the *cin* command. The getch() will pause the program to wait for a keyboard input and it will return the key entered as a character. If 1 is entered it will return '1', if 2 is entered it will return '2'. This will allow you to enter the direction without having to press the Enter key after each entry and easily move the turtle around the field.

Your program will require the following preprocessor directives. Depending on other features you add, you may need to add additional directives.

<iostream> – This allows you to use cout

<conio.h> – This allows you to use the getch() method.

Part 1: Write the basic Turtle Graphics program.

You have been provided with the following starter file. No additional changes should be made to the main function.

```
//Add the appropriate preprocessor directives here.

//Add the function prototypes here

using namespace std;

//Globals for the turtle's current and starting position.
int tr=9;
int tc=9;

int main()
{

char field[25][25];
initialize(field);
display(field);

char dir;
do{
cout << "\t 8 - Move Up" << endl;
cout << "\t 2 - Move Down" << endl;
cout << "\t 4 - Move Left" << endl;
cout << "\t 6 - Move Right" << endl;
cout << "\t 0 - Exit" << endl;

cout << "\t Direction to move: ";
dir = getch();

if(dir != '0')
    move(field, dir);

}while(dir != '0');

return 0;
}
```

You will need to create the following methods. Reference the main function to determine the necessary parameter values. All functions are void.

- **initialize** – This function should initialize the field with spaces and assign the turtle to its starting position. You choose its starting position.
- **display** – This function will display the field array to show the, path of asterisks, and current position of the turtle. The screen should be cleared before redrawing the field.
- **move** – this method will process the desired move. It will place '*' symbol in the turtle's current position and assign the turtle to its new position based on the direction entered by the user. This method should then call the **display** function to redraw the field.

Part 2: Keeping the Turtle on the field.

The program has a major flaw in it. There is nothing that keeps the turtle from moving out of the range of the field.

For this part, you should incorporate a check function that will return true if the move entered by the user is valid or false if it is invalid. The function should be run before any move is made to prevent the turtle from walking off the field.