CS205 C/ C++ Programming - Lab Assignment 3

Name: 任振裕(Ren Zhenyu)

SID: 11812214

Part 1 - Analysis

- Q1: traverse the matrix.
- Q2: Keep writing number to the array when the next term is zero.
- Q3: Use the function `utf8_to_codepoint(const unsigned char *u,int *lenptr)`.

Part 2 - Code

- `q1.cpp`

```
#include <iostream>
#include <iomanip>

using namespace std;

int main()
{
    int n, m, k;
    cin >> n >> m >> k;
    int lineset[k][4];
    int **array = new int *[n];
    for (int i = 0; i < n; i++)
    {
        array[i] = new int[m];
    }
    int x, y, dx, dy;
    for (int i = 0; i < k; i++)
    {
        cin >> x >> y >> dx >> dy;
        if (x >= n || x < 0 || y >= m || y < 0)
        {
            cout << "Error, start point not on the screen." << endl;
            exit(0);
        }
        if (!(dx == 0 || dx == 1 || dx == -1) || !(dy == 0 || dy == 1 || dy
== -1) || (dx == 0 && dy == 0))
        {
            cout << "Error, wrong direction of the bullet." << endl;
            exit(0);
        }
        do
        {
            array[x][y] = 1;
            x = x + dx;
            y = y + dy;
        } while (x >= 0 && x < n && y >= 0 && y < m);
```

```
    }
    int count = 0;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            if (array[i][j] == 0)
            {
                count++;
            }
        }
    }
    cout << count << endl;
}
```

- `q2.cpp`

```
#include <iostream>
#include <iomanip>
#include <math.h>
using namespace std;

void changeDirection(int &dx, int &dy)
{
    if (dy == -1)
    {
        dx = 1;
        dy = 0;
        return;
    }
    if (dx == 1)
    {
        dx = 0;
        dy = 1;
        return;
    }
    if (dy == 1)
    {
        dx = -1;
        dy = 0;
        return;
    }
    if (dx == -1)
    {
        dx = 0;
        dy = -1;
        return;
    }
}

int main()
{
    int n, m;
    cin >> n >> m;
    int **a;
    a = new int *[n];
    for (int i = 0; i < n; i++)
```

```
    {
        a[i] = new int[m];
    }
    int x = 0, y = m;
    int dx = 0;
    int dy = -1;
    for (int i = 1; i <= n * m; i++)
    {
        if (x + dx < 0 || x + dx > n - 1 || y + dy < 0 || y + dy > m - 1 ||
a[x + dx][y + dy] != 0)
        {
            changeDirection(dx, dy);
        }
        x = x + dx;
        y = y + dy;
        a[x][y] = i;
    }
    for (int i = 0; i < n; i++)
    {
        int width = 0;
        for (int j = 0; j < m; j++)
        {
            width = ((int)log10(n * m) + 1) + 1 - ((int)log10(a[i][j]) + 1);
            cout << a[i][j];
            cout << setw(width) << "";
        }
        cout << endl;
    }
}
```

- `q3.cpp`

```
#include <iostream>
#include <cstring>
#include <fstream>
#include "utf8.c"

#define ARRAY_SIZE 300
#define FILENAME "Blocks.txt"

using namespace std;

struct Block
{
    unsigned int start;
    unsigned int end;
    string block_name;
    int count = 0;
};

Block *readArray(string filename)
{
    Block *blockArray;
    blockArray = new Block[ARRAY_SIZE];
    ifstream myfile(filename);
    if (!myfile.is_open())
    {
```

```
        cout << "Error, can not open the file: " << filename << endl;
    }
    string temp = "";
    int i = 0;
    while (getline(myfile, temp))
    {
        if (temp[0] == '#' || temp == "")
        {
            continue;
        }
        int dot = temp.find_first_of('.');
        int semicolon = temp.find_first_of(';');
        blockArray[i].start = stoi(temp.substr(0, dot), nullptr, 16);
        blockArray[i].end = stoi(temp.substr(dot + 2, semicolon - dot - 2),
nullptr, 16);
        blockArray[i].block_name = temp.substr(semicolon + 2);
        // cout << "Block[" << i << "]: start: " << blockArray[i].start <<
", end: " << blockArray[i].end << ", name: " << blockArray[i].block_name <<
", count: " << blockArray[i].count << endl;
        i++;
    }
    return blockArray;
}

int main()
{
    Block *blockArray = new Block[ARRAY_SIZE];
    blockArray = readArray(FILENAME);
    while (!cin.eof())
    {
        unsigned char *temp;
        string line;
        int bytes_in_char;
        getline(cin, line);
        temp = (unsigned char *)&(line[0]);
        while (*temp)
        {
            unsigned int code = utf8_to_codepoint(temp, &bytes_in_char);
            // cout << bytes_in_char << endl;
            if (code)
            {
                temp = temp + bytes_in_char;
            }
            else
            {
                temp++;
            }
            for (int i = 0; i < ARRAY_SIZE; i++)
            {
                if (code <= blockArray[i].end && code >= blockArray[i].start
&& blockArray[i].block_name != "")
                {
                    blockArray[i].count++;
                }
            }
        }
    }
    Block max = blockArray[0];
```

```
    for (int i = 0; i < ARRAY_SIZE; i++)
    {
        // cout << "Block[" << i << "]: name: " << blockArray[i].block_name
<< ", count: " << blockArray[i].count << endl;
        if (blockArray[i].count > max.count)
        {
            max = blockArray[i];
        }
    }
    cout << "Max's name: " << max.block_name << ", max's count:" <<
max.count << endl;
    return 0;
}
```

Part 3 - Result & Verification

1. Test case for question 1:
    - input:

```
3 4 5
1 1 1 -1
1 1 -1 1
0 3 1 0
0 2 1 0
0 0 -1 -1
```

    - output:

```
3
```

2. Test case for question 2:
    - case 1: input and output:

```
4 4
4  3  2  1
5  14 13 12
6  15 16 11
7  8  9  10
```

    - case 2: input and output:

```
5 3
3  2  1
4  13 12
5  14 11
6  15 10
7  8  9
```

3. Test case for question 3:
    - case 1: sample.txt

```
11812214@cpplab2:~/HW/HW3/q3/build$ ./q3 < ../TEST\ DATA\ FOR\ LAB\
4/sample.txt
```

```
Max's name: Armenian, max's count:3230
```

- case 2: sample2.txt

```
11812214@cpplab2:~/HW/HW3/q3/build$ ./q3 < ../TEST\ DATA\ FOR\ LAB\
4/sample2.txt
```

```
Max's name: Georgian, max's count:1127
```

- case 3: sample3.txt

```
11812214@cpplab2:~/HW/HW3/q3/build$ ./q3 < ../TEST\ DATA\ FOR\ LAB\
4/sample3.txt
```

```
Max's name: Lao, max's count:454
```

- case 4: sample4.txt

```
11812214@cpplab2:~/HW/HW3/q3/build$ ./q3 < ../TEST\ DATA\ FOR\ LAB\
4/sample4.txt
```

```
Max's name: Malayalam, max's count:1961
```

- case 5: sample5.txt

```
11812214@cpplab2:~/HW/HW3/q3/build$ ./q3 < ../TEST\ DATA\ FOR\ LAB\
4/sample5.txt
```

```
Max's name: Devanagari, max's count:3960
```

- case 6: sample6.txt

```
11812214@cpplab2:~/HW/HW3/q3/build$ ./q3 < ../TEST\ DATA\ FOR\ LAB\
4/sample6.txt
```

```
Max's name: Georgian, max's count:1088
```

Part 4 - Difficulties & Solutions

- Use `new` to initialize the array to be all-zero valued.