

CS205 C/ C++ Programming - Lab Assignment3

Name:金冬阳 (Jin Dongyang)

SID:11911221

Part 1 - Analysis

环境:

CLion 2020.3.1x64

操作系统: Windows10 (家庭中文版64位)

CPU: Intel Core i5-9300H

内存: 8G

According to the different requirements of the three questions were judged.

Part 2 - Code

A1

```
#include <iostream>

using namespace std;
int input(const string& n);
```

```

int readNum(int a);
int readDirection();
int main() {
    int n,m,k;
    n= input("n");
    m= input("m");
    k= input("k");

    if(k==0){
        cout<<"there are no bullets on the screen."<<endl;
        cout<<n*m;
        return 0;
    }

    int **array;
    array= new int *[n];
    for(int i=0;i<n;i++){
        array[i]=new int [m];
    }

    for (int i = 0; i < k; i++) {
        int x, y, direction_x, direction_y;
        x=readNum(n);
        y=readNum(m);
        direction_x=readDirection();
        direction_y=readDirection();

        while ((x >= 0 && x < n) && (y >= 0 && y < m)) {
            array[x][y] = 1;
            x += direction_x;
            y += direction_y;
        }
    }

    int count = 0;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            if(array[i][j]!=1){
                count++;
            }
        }
    }

    cout << count;

```

```

        return 0;
    }

int input(const string& n){
    int num;
    cin>>num;
    if(cin.fail()){
        cout<<n<<" : type error, must be an integer "<<endl;
        exit(1);
    } else if(num<=0&&n!="k"){//m,n must be a positive integer!
        cout<<n<<" : number error, must be a positive integer"<<endl;
        exit(1);
    } else if(num<0&&n=="k") { //k may be 0
        cout<<n<<" : number error, must be a positive integer"<<endl;
        exit(1);
    }
    return num;
}

int readNum(int a){
    int num;
    cin>>num;
    if(cin.fail()) {
        cout << "bullet: type error, must be an integer " << endl;
        exit(1);
    } else if(num < 0 || num >= a){
        cout << "bullet missing, bullet must be in screen!" << endl;
    }
    return num;
}

int readDirection(){
    int num;
    cin>>num;
    if(cin.fail()) {
        cout << "direction: type error, must be an integer " << endl;
        exit(1);
    } else if(abs(num) > 1){
        cout << "direction error, exit" << endl;
    }
    return num;
}

```

```

#include <iostream>
#include <iomanip>

using namespace std;
void change(int &x, int &y);
int readNum();
int calculate_length(int nm);

int main() {
    int n, m;
    n=readNum();
    m=readNum();
    int length = calculate_length(n*m);//in order to fill in the space
when aligning left

    int **array;
    array= new int *[n];
    for(int i=0;i<n;i++){
        array[i]=new int [m];
    }
    for(int j=0;j<n;j++) {
        for (int i = 0; i < m; i++) {
            array[j][i] = 0;
        }
    }

    int x = m, y = 0;
    int direction_x = -1, direction_y = 0;

    for (int i = 1; i <= n * m; i++) {
        x += direction_x, y += direction_y;
        array[y][x] = i;

        int nx = x + direction_x;
        int ny = y + direction_y;
        if (nx < 0 || nx >= m || ny < 0 || ny >= n || array[ny][nx] !=
0) {
            change(direction_x, direction_y);
        }
    }
}

```

```

        for(int i=0;i<n;i++){
            for(int j=0;j<m;j++){
                cout<<setiosflags(ios::left) << setw(length) << setfill(' ')
<< array[i][j] << " ";
                //align left, fill in space
            }
            cout<<endl;
        }

        return 0;
    }

    void change(int &x, int &y) {
        if (x == -1) {
            x = 0;
            y = 1;
        } else if (y == 1) {
            y = 0;
            x = 1;
        } else if (x == 1) {
            x = 0;
            y = -1;
        } else {
            y = 0;
            x = -1;
        }
    }
}

int readNum() {
    int num;
    cin>>num;
    if (cin.fail() || num < 1 ) {
        cout << "error input, exit" << endl;
        exit(1);
    }

    return num;
}

int calculate_length(int nm) {
    int length=1;
    while(nm/10>0){
        length++;
        nm=nm/10;
    }
}

```

```

        return length;
    }

```

A3

```

#include <iostream>
#include <vector>
#include <fstream>
#include <map>
#include <sstream>
#include "utf8.c"

using namespace std;
struct utf8_block {
    string name;
    int start;
    int end;
    int count = 0;
};

vector<utf8_block > read_utf8_blocks(const string& loadingway) ;

int main() {
    vector<utf8_block > array=read_utf8_blocks("Blocks.txt");
    string full;
    char input;
    while ((input = cin.get()) != char_traits<char>::eof()) {
        full += input;
    }

    unsigned char *u;
    int bytesWithchar;
    unsigned int num;

    u=(unsigned char*)full.data();
    while (*u) {
        num = utf8_to_codepoint(u, &bytesWithchar);
        if (num) {

```

```

        u += bytesWithchar;
        _utf8_incr(u);
    } else {
        u++;
    }
    for (int j = 0; j < array.size(); j++) {
        if (num >= array[j].start && num <= array[j].end) {
            array[j].count += 1;
            break;
        }
    }
}
utf8_block language = array[0];
for (int j = 0; j < array.size(); j++) {
    if (language.count < array[j].count) {
        language = array[j];
    }
}
cout << language.name<<" matches.";
return 0;
}

vector<utf8_block > read_utf8_blocks(const string& loadingway) {
    vector<utf8_block > array;
    array.clear();
    ifstream myfile(loadingway);
    if (myfile.fail()) {
        cout << "Input failed.";
        exit(1) ;
    }
    string line;
    int count = 0;
    int judge_if_cin = 0;
    while (getline(myfile, line)) {
        if (judge_if_cin < 34) {
            judge_if_cin += 1;
            continue;
        }
        if (line.empty() || line[0] == '#') {
            continue;
        }
        utf8_block temp;
        stringstream sin(line);
        string field;
        string whole;

```

```

        vector<string> token;
        getline(sin, field, '.');
        token.push_back(field);
        getline(sin, field, '.');
        getline(sin, field, ';');
        token.push_back(field);
        getline(sin, field);
        token.push_back(field);
        temp.name=token[2];
        temp.start = stoi(token[0], nullptr, 16);
        temp.end = stoi(token[1], nullptr, 16);
        array.push_back(temp);
        count += 1;
    }
    myfile.close();
    return array;
}

```

Part 3 - Result & Verification

Test case #1_A1:

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

Test case #2_A1:

input:

```
4 4 4
0 0 1 0
0 1 1 0
0 2 1 0
0 3 1 0
```

output:

```
0
```

Test case #3_A2:

input:

```
5 3
```

output:

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

Test case #4_A2:

input:

```
3 5
```

output:

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

Test case #5_A3:

input:

```
sample.txt
```

output:

```
Armenian matches.
```

Test case #6_A3:

input:

```
sample2.txt
```

output:

```
Georgian matches.
```

Part 4 - Difficulties & Solutions

dynamic setting of two dimensional array.

