**★★☆☆☆**

**題組：Problem Set Archive** **with Online Judge**

**題號：10267: Graphical Editor**

**解題者：呂晉豪**

**解題日期：2024年3月14日**

**題意：**

模擬一個繪圖軟體，根據題目給的指令，對二維陣列上色或輸出(二維陣列的每個元素代表顏色)。

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述以下為指令表：

**題意範例：**

輸入：

I 5 6 → 創建6x5的二維陣列，初始顏色都塗成O

L 2 3 A → 把(2,3)這個位置塗成A

S one.bmp → 印出one.bmp字串，再印出整個二維陣列

G 2 3 J → G沒有在指令表上面，它是不合法的，題目說要忽略它

F 3 3 J → 把(3,3)所在的區域全部塗成J，同區域會同顏色，並且區域中的每點都可以互相抵達(不能跨過其他顏色)

V 2 3 4 W → V代表vertical，把(2,3)到(2,4)塗成W

H 3 4 2 Z → H代表horizontal，把(3,2)到(4,2)塗成Z

S two.bmp → 印出two.bmp字串，再印出整個二維陣列X

輸出：

one.bmp

OOOOO

OOOOO

OAOOO

OOOOO

OOOOO

OOOOO

two.bmp

JJJJJ

JJZZJ

JWJJJ

JWJJJ

JJJJJ

JJJJJ

**解法：**

F是將某個區域填滿，可以用BFS或DFS，其他指令照著題目做即可。

有以下幾點注意事項：

1.讀到錯誤的指令要忽略

2.V的參數Y1,Y2，H的X1,X2大小可能顛倒

3.左上角是原點，I的M,N分別代表x軸與y軸的長度

4.其餘指令的X,Y代表對(X,Y)座標進行操作

**解法範例：**

無

**討論：**

我自己在寫這題時，沒有看清楚題目就去寫程式，導致許多bug的出現，浪費許多時間，而我這時候才體悟到楊老師所說的解題三步驟：

一、理解題目

二、設計演算法

三、寫程式

是非常重要的，我花太少時間在第一步，導致我第三步會花很多時間，如果我能看清楚題目再寫，程式寫對的機率會大幅提高，debug時間就能大幅縮短。

因此，從這題中我學到，要嚴格依照解題三步驟來解題，以免花了大把的時間卻徒勞無功。

**程式：**

#include <bits/stdc++.h>

using namespace std;

vector <char> v[255];

int N , M;

int dy[] = {1 , 0 , -1 , 0};

int dx[] = {0 , 1 , 0 , -1};

void I(int n , int m){

for(int i = 0 ; i <= n + 1 ; i++){

v[i].resize(m + 2 , 0);

}

for(int i = 1 ; i <= n ; i++){

for(int j = 1 ; j <= m ; j++){

v[i][j] = 'O';

}

}

}

void C(){

for(int i = 1 ; i <= N ; i++){

for(int j = 1 ; j <= M ; j++){

v[i][j] = 'O';

}

}

}

void L(int x , int y , int c){

v[y][x] = c;

}

void V(int x , int y1 , int y2 , int c){

if(y1 > y2){

swap(y1 , y2);

}

for(int i = y1 ; i <= y2 ; i++){

v[i][x] = c;

}

}

void H(int x1 , int x2 , int y , int c){

if(x1 > x2){

swap(x1 , x2);

}

for(int i = x1 ; i <= x2 ; i++){

v[y][i] = c;

}

}

void K(int x1 , int y1 , int x2 , int y2 , int c){

for(int i = y1 ; i <= y2 ; i++){

for(int j = x1 ; j <= x2 ; j++){

v[i][j] = c;

}

}

}

void F(int x , int y , int c){ //bfs

int xyC = v[y][x];

v[y][x] = c;

bool isVisited[255][255] = {};

isVisited[y][x] = 1;

queue <pair <int , int>> q;

q.push({y , x});

while(!q.empty()){

pair <int , int> p = q.front();

q.pop();

int itY = p.first;

int itX = p.second;

v[itY][itX] = c;

for(int i = 0 ; i < 4 ; i++){

if(!isVisited[itY + dy[i]][itX + dx[i]]){

if(v[itY + dy[i]][itX + dx[i]] == xyC){

q.push({itY + dy[i] , itX + dx[i]});

}

isVisited[itY + dy[i]][itX + dx[i]] = 1;

}

}

}

}

void S(string str){

cout << str << endl;

for(int i = 1 ; i <= N ; i++){

for(int j = 1 ; j <= M ; j++){

cout << v[i][j];

}

cout << endl;

}

}

int main() {

string str;

while(1){

getline(cin , str);

int flag = 1;

switch(str[0]){

case 'I':

case 'C':

case 'L':

case 'V':

case 'H':

case 'K':

case 'F':

case 'S':

case 'X':

break;

default:

flag = 0;

}

if(flag){

stringstream ss;

ss << str;

char ins;

ss >> ins;

int m1 , n1 , x , y , x1 , x2 , y1 , y2;

string str2;

char c;

switch(ins){

case 'I':

ss >> m1 >> n1;

M = m1;

N = n1;

I(n1 , m1);

break;

case 'C':

C();

break;

case 'L':

ss >> x >> y >> c;

L(x , y , c);

break;

case 'V':

ss >> x >> y1 >> y2 >> c;

V(x , y1 , y2 ,c);

break;

case 'H':

ss >> x1 >> x2 >> y >> c;

H(x1 , x2 , y , c);

break;

case 'K':

ss >> x1 >> y1 >> x2 >> y2 >> c;

K(x1 , y1 , x2 , y2 , c);

break;

case 'F':

ss >> x >> y >> c;

F(x , y , c);

break;

case 'S':

ss >> str2;

S(str2);

break;

case 'X':

return 0;

}

}

}

return 0;

}