**★☆☆☆☆**

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

**題號：10189: Minesweeper**

**解題者：葉杰明**

**解題日期：2024年3月7日**

**題意：**

踩地雷遊戲，程式目標為找出MN區域內每個方塊附近的地雷數。

**題意範例：**

輸入測資有n(列數)、 m(行數)，當n和m皆為0時即結束，再來輸入nm區域中每個方塊內的資料。’.’代表安全點，’\*’代表地雷。輸出為”Field #x:”(x從1開始)，再來將區域中的’.’改為附近的地雷數，’\*’則照原樣輸出。

4 4

\* . . .

. . . .

. \* . .

. . . .

➔

Field #1:

\*100

2210

1\*10

1110

3 5

\* \* . . .

. . . . .

. \* . . .

➔

Field #2:

\*\*100

33200

1\*100

0 0

➔

end

**解法：**

為避免索引值超出範圍以及方便程式的編寫，所以用(n+2)(m+2)二維字元陣列field儲存資料，並將每格初始化維0，當讀取到‘\*’時，將‘\*’填入field中，並將周圍非‘\*’的格子加一，如範例所示。最後只輸出第1~n列以及第1~m行，‘\*’直接輸出，其餘轉成整數型態輸出。

**解法範例：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | \* | 1 |
| 0 | 1 | 1+1 | 1+1 | 1 |
| 0 | 1 | \* | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 |

**討論：**

1. 二維陣列多加了外圍一圈，可能增加了記憶體的使用，但可以減少判斷索引值是否超過範圍的步驟，讓程式寫起來比較簡潔。
2. 時間複雜度：O(9\*n\*m)

**程式：**

#include <iostream>

using namespace std;

int main()

{

int row, col, T = 0;

while (cin >> row >> col && (row && col))

{

char\*\* field = new char\* [row + 2];

for (int i = 0; i < row + 2; i++)

field[i] = new char[col + 2]();

for (int i = 1; i <= row; i++)

{

for (int j = 1; j <= col; j++)

{

char input;

cin >> input;

if(input == '\*')

{

field[i][j] = '\*';

for (int k = i - 1; k <= i + 1; k++)

{

for (int l = j - 1; l <= j + 1; l++)

{

if (field[k][l] == '\*')

continue;

else

field[k][l]++;

}

}

}

}

}

if (T)

cout << endl;

cout << "Field #" << ++T << ":\n";

for (int i = 1; i <= row; i++)

{

for (int j = 1; j <= col; j++)

{

if (field[i][j] == '\*')

cout << '\*';

else

cout << (int)field[i][j];

}

cout << endl;

}

for (int i = 0; i < row + 2; i++)

delete[]field[i];

delete[]field;

}

return 0;

}