\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Report: HW6

Author: E94074029 江羿賢 <e94074029@gs.ncku.edu.tw>

Class: 資訊111級乙班

Description:

這次的踩地雷的遊戲非常的有趣，在當中需要利用指標去找到array的位置，非常充分的練習了指標，並且也比較熟悉傳入二維的陣列去操作，以及注意到了不能不小心更動非屬於自己的記憶體，要對這些做一些設定，是個好玩的練習題呢。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Code:

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define Nmax 30

void expand(int \*p, int a[][Nmax]);

void build\_and\_count(int N, int M, int a[][Nmax]);

int output(int N, int a[][Nmax]);

int main(int argc, char \*argv[])

{

int N = atoi(argv[1]);

int M = atoi(argv[2]);

if (N > Nmax)

{

fprintf(stderr, "error input , N is too large\n");

exit(1);

}

srand(time(0));

int a[Nmax][Nmax];

int \*p;

for (p = &a[0][0]; p <= &a[Nmax - 1][Nmax - 1]; p++)

{

\*p = 0;

}

//build and count mine

build\_and\_count(N, M, a);

//build and count mine end

//set the initial mine

int time = 0;

while (time != N)

{

for (p = &a[time][0]; p <= &a[time][N - 1]; p++)

{

if (\*p != -1)

\*p = \*p + 100;

printf("? ");

}

printf("\n");

time++;

}

printf("\n");

//set the initial mine end

int row = 0;

int col = 0;

int win = 0;

//check win or dead

while (1)

{

//input

scanf(" %d", &row);

scanf(" %d", &col);

//input end

win = 0;

p = &a[row][col];

if (\*p < 0)

{

printf("You are dead!\n");

break;

}

system("clear");

if (\*p == 100)

expand(p, a);

else

\*p = \*p - 100;

win = output(N, a); //out put the result and check is win?

printf("\n");

if (win == N \* N - M)

{

printf("You win!\n");

break;

}

}

//check win or dead end

}

void build\_and\_count(int N, int M, int a[][Nmax])

{

int count = M;

int r = 0;

int c = 0;

int \*p;

int site = 0;

int \*s;

while (count)

{

r = rand() % N;

c = rand() % N;

if (a[r][c] != -1)

{

p = &a[r][c];

\*p = -1;

if (\*(p + 1) != -1)

{

site = 0;

for (s = &a[0][0]; s <= &a[Nmax - 1][Nmax - 1]; s++)

{

if (s == (p + 1))

{

site++;

break;

}

}

if (site)

\*(p + 1) = \*(p + 1) + 1;

}

if (\*(p - 1) != -1)

{

site = 0;

for (s = &a[0][0]; s <= &a[Nmax - 1][Nmax - 1]; s++)

{

if (s == (p - 1))

{

site++;

break;

}

}

if (site)

\*(p - 1) = \*(p - 1) + 1;

}

for (p = &a[r][c] - Nmax - 1; p <= &a[r][c] - Nmax + 1; p++)

{

if (\*p != -1)

{

site = 0;

for (s = &a[0][0]; s <= &a[Nmax - 1][Nmax - 1]; s++)

{

if (s == p)

{

site++;

break;

}

}

if (site)

\*p = \*p + 1;

}

}

for (p = &a[r][c] + Nmax - 1; p <= &a[r][c] + Nmax + 1; p++)

{

if (\*p != -1)

{

site = 0;

for (s = &a[0][0]; s <= &a[Nmax - 1][Nmax - 1]; s++)

{

if (s == p)

{

site++;

break;

}

}

if (site)

\*p = \*p + 1;

}

}

count--;

}

}

}

int output(int N, int a[][Nmax])

{

int win = 0;

int time = 0;

int \*p;

while (time != N)

{

for (p = &a[time][0]; p <= &a[time][N - 1]; p++)

{

if (\*p >= 100 || \*p == -1)

{

printf("? ");

}

else

{

if (\*p != 0)

printf("%d ", \*p);

else

printf("\_ ");

win++;

}

}

printf("\n");

time++;

}

return win;

}

void expand(int \*p, int a[][Nmax])

{

\*p = \*p - 100;

int site = 0;

int \*s;

//1

for (s = &a[0][0]; s <= &a[Nmax - 1][Nmax - 1]; s++)

{

if (s == (p + 1))

{

site++;

break;

}

}

if (site)

{

if (\*(p + 1) > 100)

\*(p + 1) = \*(p + 1) - 100;

if (\*(p + 1) == 100)

expand(p + 1, a);

}

//2

site = 0;

for (s = &a[0][0]; s <= &a[Nmax - 1][Nmax - 1]; s++)

{

if (s == (p - 1))

{

site++;

break;

}

}

if (site)

{

if (\*(p - 1) > 100)

\*(p - 1) = \*(p - 1) - 100;

if (\*(p - 1) == 100)

expand(p - 1, a);

}

//3

site = 0;

for (s = &a[0][0]; s <= &a[Nmax - 1][Nmax - 1]; s++)

{

if (s == (p + Nmax))

{

site++;

break;

}

}

if (site)

{

if (\*(p + Nmax) > 100)

\*(p + Nmax) = \*(p + Nmax) - 100;

if (\*(p + Nmax) == 100)

expand(p + Nmax, a);

}

//4

site = 0;

for (s = &a[0][0]; s <= &a[Nmax - 1][Nmax - 1]; s++)

{

if (s == (p - Nmax))

{

site++;

break;

}

}

if (site)

{

if (\*(p - Nmax) > 100)

\*(p - Nmax) = \*(p - Nmax) - 100;

if (\*(p - Nmax) == 100)

expand(p - Nmax, a);

}

//5

site = 0;

for (s = &a[0][0]; s <= &a[Nmax - 1][Nmax - 1]; s++)

{

if (s == (p + Nmax - 1))

{

site++;

break;

}

}

if (site)

{

if (\*(p + Nmax - 1) > 100)

\*(p + Nmax - 1) = \*(p + Nmax - 1) - 100;

if (\*(p + Nmax - 1) == 100)

expand(p + Nmax - 1, a);

}

//6

site = 0;

for (s = &a[0][0]; s <= &a[Nmax - 1][Nmax - 1]; s++)

{

if (s == (p + Nmax + 1))

{

site++;

break;

}

}

if (site)

{

if (\*(p + Nmax + 1) > 100)

\*(p + Nmax + 1) = \*(p + Nmax + 1) - 100;

if (\*(p + Nmax + 1) == 100)

expand(p + Nmax + 1, a);

}

//7

site = 0;

for (s = &a[0][0]; s <= &a[Nmax - 1][Nmax - 1]; s++)

{

if (s == (p - Nmax + 1))

{

site++;

break;

}

}

if (site)

{

if (\*(p - Nmax + 1) > 100)

\*(p - Nmax + 1) = \*(p - Nmax + 1) - 100;

if (\*(p - Nmax + 1) == 100)

expand(p - Nmax + 1, a);

}

//8

site = 0;

for (s = &a[0][0]; s <= &a[Nmax - 1][Nmax - 1]; s++)

{

if (s == (p - Nmax - 1))

{

site++;

break;

}

}

if (site)

{

if (\*(p - Nmax - 1) > 100)

\*(p - Nmax - 1) = \*(p - Nmax - 1) - 100;

if (\*(p - Nmax - 1) == 100)

expand(p - Nmax - 1, a);

}

}

Compilation:

gcc mine.c -o mine

Execution:

./mine 3 2

Output:

? ? ?

? ? ?

? ? ?

0 0

\_ 1 ?

1 2 ?

? ? ?

2 2

\_ 1 ?

1 2 ?

? ? 2

2 0

\_ 1 ?

1 2 ?

1 ? 2

0 2

\_ 1 1

1 2 ?

1 ? 2

You win!