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

Report: HW6

Author: C24089036 楊佳恩 <[charles870126@gmail.com](mailto:charles870126@gmail.com)>

Class: 物理系

Description:

How do you finish this homework? Gooooooooogle

What did you learned from this homework? 用pointer去取出array的值(我不是故意用晶晶體 = =)

Did you do or write something special for bonus? 沒☹

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

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define Nmax 30

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

int N = atoi(argv[1]);

int M = atoi(argv[2]);

int grid[N+2][N+2], \*p, \*q, \*op;

int i, j;

int op\_x, op\_y;

int n = 0;

int over = 0;

//initial

for (p = &grid[0][0]; p <= &grid[N+1][N+1]; p++)

\*p = 0;

//put\_mine

srand(time(NULL));

do {

int mine\_x = rand() % N +1;

int mine\_y = rand() % N +1;

int \*mine = &grid[mine\_x][mine\_y];

if (\*mine != -1) {

\*mine = -1;

n++;

}

} while (n < M);

//calculate\_num\_of\_cells

for (p = &grid[0][0]; p <= &grid[N+1][N+1]; p++){

if (\*p == 0){

for (i = -(N+2); i <= (N+2); i+=(N+2)){

for (q = p+i - 1; q <= p+i + 1; q++)

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

}

}

}

//print\_blanks

for (i = 0; i < N; i++) {

for (j = 0; j < N; j++)

printf(" ?");

printf("\n");

}

while (over == 0){

//N is greater than Nmax

if (N > Nmax) {

printf("N is too large ");

system("clear");

break;

}

//choose a position

scanf("%d%d", &op\_x, &op\_y);

op\_x++;

op\_y++;

op = &grid[op\_x][op\_y];

system("clear"); //clear screen

//assign\_inf\_after\_open

if (\*op == -1) \*op = -2;

if (\*op >= 0 && \*op <= 8) \*op += 10;

//open\_surrounding

open\_surrounding(grid[N+2][N+2], op, N);

//check\_over

over = 1;

if (\*op == -2) over = -1;

else

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

for (p = grid[i]+1; p <= grid[i]+N; p++)

if (\*p >= 0 && \*p <= 8)

over = 0;

//print\_cells

if (over == 0)

{

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

for (p = grid[i]+1; p <= grid[i]+N; p++){

if (\*p < 10)

printf(" ?");

else if (\*p == 10)

printf(" \_");

else

printf(" %d", \*p - 10);

}

printf("\n");

}

printf("\n");

}

else

{

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

for (p = grid[i]+1; p <= grid[i]+N; p++){

if (\*p < 10 && \*p >= 0)

printf(" ?");

else if (\*p <= -1)

printf(" \*");

else if (\*p == 10)

printf(" \_");

else

printf(" %d", \*p - 10);

}

printf("\n");

}

printf("\n");

if (over == 1)

printf("you win ");

if (over == -1)

printf("you are dead ");

}

}

return(0);

}

void open\_surrounding(int grid[][Nmax], int \*op, int N)

{

int \*p, i;

if (\*op == 10)

{

for (i = -(N+2); i <= (N+2); i+=(N+2)){

for (p = op+i - 1; p <= op+i + 1; p++){

if (\*p >= 0 && \*p <=8)

{

\*p += 10;

open\_surrounding(grid, op, N);

}

}

}

}

}

Compilation:

gcc mine.c -o mine

Execution:

./mine 5 4

Output:

? ? ? ? ?

? ? ? ? ?

? ? ? ? ?

? ? ? ? ?

? ? ? ? ?