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

Report: HW6

Author: E34071061 謝沅承 <andy420811@gmail.com>

Class: 化工 (甲班)

Description:

使用rand將地雷隨機存入陣列中，並計算周圍炸彈數，存入陣列a中，並另外宣告一個Output陣列，讀取使用者輸入的行和列，如果為炸彈則結束程式，否則將a和output傳入extend的函數，如果周圍炸彈數大於0則將output的該位置改為1，如果周圍炸彈為0將output的九宮格內改為1，且周圍有同為0個炸彈時，且該位置的output不為1時，將該格重複執行entend的函數，重複讀入使用者輸入的行列，直到output的1總數和a中不為地雷的總數相同時印出win結束程式

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

#include<stdio.h>

#include<time.h>

#include<stdlib.h>

#define Nmax 30

void prtarr(const int a[][32],int n);

void Extend(int a[][32],int b[][32],int r,int c);

int Count(const int a[][32],int n);

int state(const int a[][32],int n,int t);

void prtarr\_u(const int a[][32],const int b[][32],int n);

int end(const int a[][32],int b[][32],int r,int c);

int checknozero( int a[][32],const int r,int c);

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

int N,M,i,\*p;int c=0,d=0;int \*check[8];

N=atol(argv[1]);M=atol(argv[2]);

if(N>30){printf("N larger than 30"); return -1;}

int a[32][32]={0,0,0};

srand(time(NULL));

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

c=N\*((float)rand()/RAND\_MAX);

d=N\*((float)rand()/RAND\_MAX);

if(\*(a[c+1]+d+1)==1){i--;}

else \*(\*(a+c+1)+d+1)=-1;

}

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

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

if(\*(\*(a+i)+j)!=-1){

int count=0;

check[0]=(\*(a+i-1)+j-1);check[1]=(\*(a+i-1)+j);check[2]=(\*(a+i-1)+j+1);

check[3]=(\*(a+i )+j-1); check[4]=(\*(a+i )+j+1);

check[5]=(\*(a+i+1)+j-1);check[6]=(\*(a+i+1)+j);check[7]=(\*(a+i+1)+j+1);

for(int k=0;k<8;k++){

if(\*(check[k])==-1){count-=\*check[k];}

}

\*(\*(a+i)+j)=count;

}

}

}

for(int i=0;i<N+2;i++){

\*(a[0]+i)=-20;

\*(a[N+1]+i)=-20;

\*(a[i]+0)=-20;

\*(a[i]+N+1)=-20;

}

//prtarr(a,N);

int out[32][32]={0};

int r,col;

prtarr\_u(a,out,N);

int step=Count(a,N);

while(state(out,N,step)){

scanf("%d%d",&r,&col);

while(\*(\*(out+r+1)+col+1)==1){

printf("you have guessed");scanf("%d%d",&r,&col);}

r+=1;col+=1;

Extend(a,out,r,col);

system("clear");

prtarr\_u(a,out,N);

if(end(a,out,r,col)){

system(“clear”);

prtarr\_u(a,out,N);

printf("You are dead\n");

return 0;

}

}

printf("You win");

return 0;

}

void Extend(int a[][32],int b[][32],int r,int c){

if(\*(\*(a+r)+c)>0){

\*(\*(b+r)+c)=1; //output bitmap update to 1 on that grid

}else if(\*(\*(a+r)+c)==0){

int count=0;

int R[3]={r-1,r,r+1};int C[3]={c-1,c,c+1};

//prtarr\_u(a,b,5);

//prtarr(b,5);

int check[8],checkb[8];

checkb[0]=\*(\*(b+r-1)+c-1);checkb[1]=\*(\*(b+r-1)+c);checkb[2]=\*(\*(b+r-1)+c+1);

checkb[3]=\*(\*(b+r )+c-1); ;checkb[4]=\*(\*(b+r )+c+1);

checkb[5]=\*(\*(b+r+1)+c-1);checkb[6]=\*(\*(b+r+1)+c);checkb[7]=\*(\*(b+r+1)+c+1);

check[0]=\*(\*(a+r-1)+c-1);check[1]=\*(\*(a+r-1)+c);check[2]=\*(\*(a+r-1)+c+1);

check[3]=\*(\*(a+r )+c-1); ;check[4]=\*(\*(a+r )+c+1);

check[5]=\*(\*(a+r+1)+c-1);check[6]=\*(\*(a+r+1)+c);check[7]=\*(\*(a+r+1)+c+1);

if(!checknozero(a,r,c)){

\*(\*(b+r-1)+c-1)=1;\*(\*(b+r-1)+c )=1;\*(\*(b+r-1)+c+1)=1;

\*(\*(b+r )+c-1)=1;\*(\*(b+r )+c )=1;\*(\*(b+r )+c+1)=1;

\*(\*(b+r+1)+c-1)=1;\*(\*(b+r+1)+c )=1;\*(\*(b+r+1)+c+1)=1;

}else{

int k[8],g=0,count=0;

\*(\*(b+r-1)+c-1)=1;\*(\*(b+r-1)+c )=1;\*(\*(b+r-1)+c+1)=1;

\*(\*(b+r )+c-1)=1;\*(\*(b+r )+c )=1;\*(\*(b+r )+c+1)=1;

\*(\*(b+r+1)+c-1)=1;\*(\*(b+r+1)+c )=1;\*(\*(b+r+1)+c+1)=1;

for(int i=0;i<8;i++){

if(check[i]==0){

k[g++]=i;count++;

}

}

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

int j=k[i];

switch(k[i]){

case 0:if(!checkb[j])Extend(a,b,R[0],C[0]);break;

case 1:if(!checkb[j])Extend(a,b,R[0],C[1]);break;

case 2:if(!checkb[j])Extend(a,b,R[0],C[2]);break;

case 3:if(!checkb[j])Extend(a,b,R[1],C[0]);break;

case 4:if(!checkb[j])Extend(a,b,R[1],C[2]);break;

case 5:if(!checkb[j])Extend(a,b,R[2],C[0]);break;

case 6:if(!checkb[j])Extend(a,b,R[2],C[1]);break;

case 7:if(!checkb[j])Extend(a,b,R[2],C[2]);break;

}}

}

}

}

int checknozero(int a[][32],int r,int c){

int \*check[8];int count;

check[0]=(\*(a+r-1)+c-1);check[1]=(\*(a+r-1)+c);check[2]=(\*(a+r-1)+c+1);

check[3]=(\*(a+r )+c-1); ;check[4]=(\*(a+r )+c+1);

check[5]=(\*(a+r+1)+c-1);check[6]=(\*(a+r+1)+c);check[7]=(\*(a+r+1)+c+1);

for(int i=0;i<8;i++){

if(\*check[i]==0)count++;

}

return count?1:0;

}

void prtarr(const int a[][32],int n){

int i,j;

for(i=0;i<n+2;i++){

for(j=0;j<n+2;j++){

printf("%3d ",\*(a[i]+j));}printf("\n\n");}}

int state(const int a[][32],int n,int t){

int deter=0;

//prtarr(a,n);

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

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

if(1U&(\*(a[r]+c))){

deter+=1;

}

}

}

return deter>=t?0:1;

}

int Count(const int a[][32],int n){

int count=0;

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

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

count+=\*(a[i]+j)>-1?1:0;

}

}

return count;

}

void prtarr\_u(const int a[][32],const int b[][32],int n){

int k;

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

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

if(\*(b[i]+j)==1){

if(\*(a[i]+j)>0){

k=\*(a[i]+j);

printf("%d ",\*(a[i]+j));

}else if(\*(a[i]+j)==0){

printf("- ");

}else if(\*(a[i]+j)==-1){

printf("# ");

}

}else if(\*(b[i]+j)==0){

printf("? ");

}

}

printf("\n");

}

}

int end(const int a[][32],int b[][32],int r,int c){

if(\*(\*(a+r)+c)==-1){

\*(\*(b+r)+c)=1;

return 1;

}

return 0;

}

Compilation:

gcc -o mine mine.c

Execution:

./mine 9 1

Output:

? ? ? ? ? ? ? ? ?

? ? ? ? ? ? ? ? ?

? ? ? ? ? ? ? ? ?

? ? ? ? ? ? ? ? ?

? ? ? ? ? ? ? ? ?

? ? ? ? ? ? ? ? ?

? ? ? ? ? ? ? ? ?

? ? ? ? ? ? ? ? ?

? ? ? ? ? ? ? ? ?

3 3

- - - - - - - - -

- - - - - - - - -

- - - - - - - - -

- - - - - - - - -

- - - - - - - - -

- - - - - - - - -

- 1 1 1 - - - - -

- 1 ? 1 - - - - -

- 1 ? 1 - - - - -