

INICIO

{%codepage UTF8}%//Permite usar acenhastas y ñ en consola.

uses crt;

constanteante

filasfacil=8;

columnasfacil=8;

minasfacil=10;

filasintermedio=16;

columnasintermedio=16;

minasintermedio=40;

filasdificil=16;

columnasdificil=30;

minasdificil=99;

type

tableroacil = arreglo[1..filasfacil, 1..columnasfacil] de entero;

mostrarceldasf= arreglo [1..8,1..8] de booleano;

tablerointermedio = arreglo[1..filasintermedio, 1..columnasintermedio] de entero;

mostrarceldasI= arreglo [1..16,1..16] de booleano;

tablerodificil = arreglo[1..filasdificil, 1..columnasdificil] de entero;

mostrarceldasD= arreglo [1..16,1..30] de booleano;

variable

visiblef: mostrarceldasf;

celdas1: tableroacil;

visibleI: mostrarceldasI;

```
celdas2: tablerointermedio;  
visibleD: mostrarceldasD;  
celdas3: tablerodificil;
```

Funcion max(num1,num2:entero):entero;

variable

num:entero;

INICIO

si num1>num2 entonces

INICIO

num:=num1;

max:=num

FIN

si no

INICIO

num:=num2;

max:=num

FIN;

FIN;

Funcion min(num1,num2:entero):entero;

variable

num:entero;

INICIO

si num1<num2 entonces

INICIO

num:=num1;

min:=num

```

    FIN
    si no
    INICIO
        num:=num2;
        min:=num
    FIN;
FIN;

procedimiento colocarminasF;
variable
    f2,c2,i, x, y: Entero;

INICIO
    randomizar;
    para i := 1 hasta minasfacil hacer
    INICIO
        repetir
            x := Random(filasfacil) + 1;
            y := Random(columnasfacil) + 1;
            hasta celdas1[x, y] <> 8;
            celdas1[x, y] := 8;
        para f2 := max(1, x - 1) hasta min(8, x + 1) hacer
        INICIO
            para c2 := max(1, y - 1) hasta min(8, y + 1) hacer
            INICIO
                si celdas1[f2, c2] <> 8 entonces

```

```
        celdas1[f2, c2]:=celdas1[f2, c2]+1;
    FIN;
FIN;
FIN;
FIN;
```

procedimiento colocarminasI;

variable

f2,c2,i, x, y: Entero;

INICIO

randomizar;

para i := 1 hasta minasintermedio hacer

INICIO

repetir

 x := Random(filasintermedio) + 1;

 y := Random(columnasintermedio) + 1;

 hasta celdas2[x, y] <> 16;

 celdas2[x, y] := 16;

para f2 := max(1, x - 1) hasta min(16, x + 1) hacer

INICIO

 para c2 := max(1, y - 1) hasta min(16, y + 1) hacer

INICIO

 si celdas2[f2, c2] <> 16 entonces

 celdas2[f2, c2]:=celdas2[f2, c2]+1;

 FIN;

 FIN;

FIN;

FIN;

procedimiento colocarminasD;

variable

f2,c2,i, x, y: Entero;

INICIO

randomizar;

para i := 1 hasta minasdifícil hacer

INICIO

repetir

x := Random(filasdifícil) + 1;

y := Random(columnasdifícil) + 1;

hasta celdas3[x, y] <> 30;

celdas3[x, y] := 30;

para f2 := max(1, x - 1) hasta min(16, x + 1) hacer

INICIO

para c2 := max(1, y - 1) hasta min(30, y + 1) hacer

INICIO

si celdas3[f2, c2] <> 30 entonces

celdas3[f2, c2]:=celdas3[f2, c2]+1;

FIN;

FIN;

FIN;

FIN;

procedimiento inicializarTableroF();

variable

c,f: entero;

INICIO

para f:=1 hasta 8 hacer

INICIO

Para c:=1 hasta 8 hacer

INICIO

celdas1[f,c]:=0;

FIN;

FIN;

colocarminasF;

para f:=1 hasta 8 hacer

INICIO

Para c:=1 hasta 8 hacer

INICIO

visiblef[f,c]:= falso;

FIN;

FIN;

FIN;

procedimiento inicializarTableroI();

variable

c,f: entero;

INICIO

para f:=1 hasta 16 hacer

INICIO

Para c:=1 hasta 16 hacer

INICIO

celdas2[f,c]:=0;

```
        FIN;
    FIN;
    colocarminasI;
    para f:=1 hasta 16 hacer
    INICIO
        Para c:=1 hasta 16 hacer
        INICIO
            visibleI[f,c]:= falso;
        FIN;
    FIN;
FIN;
```

```
procedimiento inicializarTableroD();
variable
c,f: entero;
INICIO
    para f:=1 hasta 16 hacer
    INICIO
        Para c:=1 hasta 30 hacer
        INICIO
            celdas3[f,c]:=0;
        FIN;
    FIN;
    colocarminasD;
    para f:=1 hasta 16 hacer
    INICIO
        Para c:=1 hasta 30 hacer
```

```

        INICIO
            visibleD[f,c]:= falso;
        FIN;
    FIN;
FIN;

procedimiento destapar_celdasF( f,c: entero);
variable
f2,c2:entero;
INICIO
    si visiblef[f,c]=falso entonces
        INICIO
            visiblef[f,c]:= verdadero;
            si celdas1[f,c]=0 entonces
                INICIO
                    para f2 := max(1, f - 1) hasta min(8, f + 1) hacer
                        INICIO
                            para c2 := max(1, c - 1) hasta min(8, c + 1) hacer
                                INICIO
                                    si celdas1[f2, c2] <> 8 entonces
                                        destapar_celdasF(f2, c2);
                                FIN;
                            FIN;
                        FIN;
                    FIN;
                FIN;
            FIN;
        FIN;
    FIN;

procedimiento destapar_celdasI( f,c: entero);

```


variable

f2,c2:entero;

INICIO

si visibleI[f,c]=falso entonces

INICIO

visibleI[f,c]:= verdadero;

si celdas2[f,c]=0 entonces

INICIO

para f2 := max(1, f - 1) hasta min(16, f + 1) hacer

INICIO

para c2 := max(1, c - 1) hasta min(16, c + 1) hacer

INICIO

si celdas2[f2, c2] <> 16 entonces

destapar_celdasI(f2, c2);

FIN;

FIN;

FIN;

FIN;

FIN;

procedimiento destapar_celdasD(f,c: entero);

variable

f2,c2:entero;

INICIO

si visibleD[f,c]=falso entonces

INICIO

visibleD[f,c]:= verdadero;

si celdas3[f,c]=0 entonces

```

    INICIO
        para f2 := max(1, f - 1) hasta min(16, f + 1) hacer
            INICIO
                para c2 := max(1, c - 1) hasta min(30, c + 1) hacer
                    INICIO
                        si celdas3[f2, c2] <> 30 entonces
                            destapar_celdasD(f2, c2);
                    FIN;
                FIN;
            FIN;
        FIN;
    FIN;

```

```

funcion celdas_destapadasF(f,c:entero):entero;

```

```

variable

```

```

num: entero;

```

```

INICIO

```

```

    num:=0;

```

```

    para f:=1 hasta 8 hacer

```

```

        INICIO

```

```

            para c:= 1 hasta 8 hacer

```

```

                INICIO

```

```

                    si visiblef[c,f] entonces

```

```

                        INICIO

```

```

                            num:=num+1;

```

```

                            celdas_destapadasF:=num

```

```

        FIN;
    FIN;
FIN;
FIN;
```

```
funcion celdas_destapadasI(f,c:entero):entero;
```

```
variable
```

```
num: entero;
```

```
INICIO
```

```
    num:=0;
```

```
    para f:=1 hasta 16 hacer
```

```
        INICIO
```

```
            para c:= 1 hasta 16 hacer
```

```
                INICIO
```

```
                    si visibleI[c,f] entonces
```

```
                        INICIO
```

```
                            num:=num+1;
```

```
                            celdas_destapadasI:=num
```

```
                        FIN;
```

```
                    FIN;
```

```
                FIN;
```

```
            FIN;
```

```
funcion celdas_destapadasD(f,c:entero):entero;
```

```
variable
```

```
num: entero;
```

INICIO

num:=0;

para f:=1 hasta 16 hacer

INICIO

para c:= 1 hasta 30 hacer

INICIO

si visibleD[c,f] entonces

INICIO

num:=num+1;

celdas_destapadasD:=num

FIN;

FIN;

FIN;

FIN;

funcion revisar_tablero (c, f: entero): entero;

variable

resultado:entero;

INICIO

resultado:=0;

si celdas1[c,f]=8 entonces

INICIO

visiblef[c,f]:= verdadero;

resultado:=-1;

FIN

si no

INICIO

```
        destapar_celdasF(c,f);
        resultado:=celdas_destapadasF(c,f);
    FIN;
    revisar_tablero:=resultado;
FIN;
```

```
funcion revisar_tableroI (c, f: entero): entero;
variable
resultado:entero;
```

```
INICIO
    resultado:=0;
    si celdas2[c,f]=16 entonces
        INICIO
            visibleI[c,f]:= verdadero;
            resultado:=-1;
        FIN
    si no
        INICIO
            destapar_celdasI(c,f);
            resultado:=celdas_destapadasI(c,f);
        FIN;
    revisar_tableroI:=resultado;
FIN;
```

```
funcion revisar_tableroD (c, f: entero): entero;
variable
resultado:entero;
```

INICIO

 resultado:=0;

 si celdas3[c,f]=30 entonces

 INICIO

 visibleD[c,f]:= verdadero;

 resultado:=-1;

 FIN

 si no

 INICIO

 destapar_celdasD(c,f);

 resultado:=celdas_destapadasD(c,f);

 FIN;

 revisar_tableroD:=resultado;

FIN;

procedimiento mostrar_minasf;

variable

 f, c: entero;

INICIO

 para f := 1 hasta 8 hacer

 INICIO

 para c := 1 hasta 8 hacer

 INICIO

 si celdas1[f, c] = 8 entonces

 visiblef[f, c]:= verdadero;

 FIN;

```

        FIN;
FIN;

procedimiento mostrar_minasI;
variable
    f, c: entero;

INICIO
    para f := 1 hasta 16 hacer
        INICIO
            para c := 1 hasta 16 hacer
                INICIO
                    si celdas2[f, c] = 16 entonces
                        visibleI[f, c]:= verdadero;
                FIN;
            FIN;
        FIN;
FIN;

procedimiento mostrar_minasD;
variable
    f, c: entero;

INICIO
    para f := 1 hasta 16 hacer
        INICIO
            para c := 1 hasta 30 hacer
                INICIO
                    si celdas3[f, c] = 30 entonces

```

```

    FIN;

FIN;

FIN;

procedimiento marco_facil;

variable

    i, j: entero;

    colum: cadena = (' 1 2 3 4 5 6 7 8');

INICIO

    escribir linea(colum);

    para i := 1 hasta filasfacil hacer

        INICIO

            escribir(i, ' ');

            para j := 1 hasta columnasfacil hacer

                INICIO

                    si visiblef[i,j] entonces

                        INICIO

                            escribir(' ');

                        FIN

                    si no

                        INICIO

                            si celdas1[i,j] = 8 entonces

                                INICIO

                                    mostrar_minasf;

                                escribir('* ');

```



```

        FIN
    si no
        INICIO
            escribir(celdas1[i,j], ' ');
        FIN;
    FIN;
FIN
si no
    INICIO
        escribir('# ');
    FIN;
FIN;
escribir linea();
FIN;
FIN;

procedimiento marco_intermedio;
variable
    i, j: entero;
    colum: cadena = (' 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16');
INICIO
    escribir linea(colum);
    para i := 1 hasta filasintermedio hacer
        INICIO
            si i>9 entonces
                INICIO
                    escribir(i, ' ');
                FIN
            FIN
        FIN
    FIN

```

```

        si no
        INICIO
            escribir(' ',i,' ');
        FIN;
para j := 1 hasta columnasintermedio hacer
INICIO
    si visibleI[i,j] entonces
    INICIO
        si celdas2[i,j] = 0 entonces
        INICIO
            escribir(' ');
        FIN
    si no
    INICIO
        si celdas2[i,j] = 16 entonces
        INICIO
            mostrar_minasI;
        escribir(' * ');
        FIN
    si no
    INICIO
        escribir(' ',celdas2[i,j],' ');
        FIN;
    FIN;
FIN
si no
INICIO
    escribir(' # ');

```

```

        FIN;
    FIN;
    escribir linea();
    FIN;
FIN;

procedimiento marco_dificil;
variable
    i, j: entero;
    colum: cadena = (' 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
25 26 27 28 29 30');
INICIO
    escribir linea(colum);
    para i := 1 hasta filasdificil hacer
        INICIO
            si i>9 entonces
                INICIO
                    escribir(i, ' ');
                FIN
            si no
                INICIO
                    escribir(' ', i, ' ');
                FIN;
        para j := 1 hasta columnasdificil hacer
            INICIO
                si visibleD[i,j] entonces
                    INICIO
                        si celdas3[i,j] = 0 entonces
                            INICIO

```

```

        escribir(' ');
    FIN
    si no
    INICIO
        si celdas3[i,j] = 30 entonces
            INICIO
                mostrar_minasD;
                escribir(' * ');
            FIN
        si no
        INICIO
            escribir(' ',celdas3[i,j],');
        FIN;
    FIN;
FIN
si no
INICIO
    escribir(' # ');
    FIN;
FIN;
    escribir linea();
    FIN;
FIN;

procedimiento Facil();
    Variable
        f,c,resultado: entero;
    INICIO

```

```

inicializarTableroF;

repetir
INICIO
    marco_facil;
    //Pedir fila y columna de casilla a destapar
    repetir
    INICIO
        Escribir linea('Indica fila entre el 1 al 8:');
        leer linea(f);
    FIN;
    hasta (f>=1) and (f<=8);
    repetir
    INICIO
        Escribir linea('Indica una columna entre el 1 al 8:');
        leer linea(c);
    FIN;
    hasta (c>=1) and (c<=8) ;

    //Comprobamos el tablero

    //resultado:=ComprobarTablero(celdas,portada,f,c);
    resultado:=revisar_tablero(f,c);
    Limpiar pantalla;
FIN;
hasta (resultado=-1) or (resultado=54);
marco_facil;
//Has destapado una mina
si resultado=-1 entonces

```

INICIO

escribir linea('Has pisahacer una mina');

Escribir linea('¡¡¡HAS PERDIHACER!!!');

FIN

//has destapahacer hastadas las casillas

si no

INICIO

Escribir linea('Has completahacer el juego sin pisar una mina');

Escribir linea('¡¡¡FELICIDADES!!!');

FIN;

FIN;

procedimiento intermedio();

Variable

f,c,resultado: entero;

INICIO

inicializarTableroI;

repetir

INICIO

marco_intermedio;

//Pedir fila y columna de casilla a destapar

repetir

INICIO

Escribir linea('Indica fila entre el 1 al 16:');

leer linea(f);

FIN;

hasta (f>=1) and (f<=16);

```

    repetir
    INICIO
        Escribir linea('Indica una columna entre el 1 al 16:');
        leer linea(c);
    FIN;
    hasta (c>=1) and (c<=16) ;

    //Comprobamos el tablero

    resultado:=revisar_tableroI(f,c);
    Limpiar pantalla;
    FIN;
    hasta (resultado=-1) or (resultado=216);
    marco_intermedio;
//Has destapahacer una mina
    si resultado=-1 entonces
    INICIO
        mostrar_minasI;
        escribir linea('Has pisahacer una mina');
        Escribir linea('¡¡¡HAS PERDIHACER!!!');
    FIN
    //has destapahacer hastadas las casillas
    si no
    INICIO
        Escribir linea('Has completahacer el juego sin pisar una mina');
        Escribir linea('¡¡¡FELICIDADES!!!');
    FIN;

```

FIN;

procedimiento dificil();

Variable

f,c,resultado: entero;

INICIO

inicializarTableroD;

repetir

INICIO

marco_dificil;

//Pedir fila y columna de casilla a destapar

repetir

INICIO

Escribir linea('Indica fila entre el 1 al 16:');

leer linea(f);

FIN;

hasta (f>=1) and (f<=16);

repetir

INICIO

Escribir linea('Indica una columna entre el 1 al 30:');

leer linea(c);

FIN;

hasta (c>=1) and (c<=30) ;

//Comprobamos el tablero

resultado:=revisar_tableroD(f,c);

Limpiar pantalla;


```

    FIN;

    hasta (resultado=-1) or (resultado=351);

    marco_dificil;

//Has destapahacer una mina

    si resultado=-1 entonces

        INICIO

            mostrar_minasD;

            escribir linea('Has pisahacer una mina');

            Escribir linea('¡¡¡HAS PERDIHACER!!!');

        FIN

    //has destapahacer hastadas las casillas

    si no

        INICIO

            Escribir linea('Has completahacer el juego sin pisar una mina');

            Escribir linea('¡¡¡FELICIDADES!!!');

        FIN;

FIN;

variable

opcion, decision, dificultad: caracter;

INICIO

    Repetir

        INICIO

            Repetir

                INICIO

```

```

    escribir linea('Bienvenido al juego de Buscaminas');

    Escribir linea('Presione 1 para Jugar');
    Escribir linea('Presione 2 para salir');
    Leer linea(opcion);
    Limpiar pantalla;

FIN;

hasta (opcion>'1') or (opcion<'3');//evita que se elija una opcion que no existe
si(opcion='1') entonces
    INICIO
        Repetir
            INICIO
                Escribir linea('Usted ha elegido la opcion de jugar');
                Escribir linea('Elija la dificultad que quiere jugar');
                Escribir linea('Presione 1 para modo Facil');
                Escribir linea('presiones 2 para modo Intermedio');
                escribir linea('Presiones 3 para modo Dificil');
                leer linea(dificultad);
                Limpiar pantalla;
            FIN;
            hasta (dificultad>'1') o (dificultad<'4');//evita que se elija una
dificultad que no existe, pero te devuelve al inicio del programa
            case dificultad de
                '1': INICIO
                    Escribir linea('Usted ha elegido la dificultad
fácil');

                    escribir linea('Diviertase, presione cualquier
tecla para iniciar');

                    readkey;
                    Limpiar pantalla;

```

```

repetir
    INICIO
        Facil;
        escribir linea('Ha terminado la
partida');
        escribir linea('Presione 1 si
quiere volver a la pantalla de inicio');
        escribir linea('Presione cualquier
otra tecla para seguir jugando en esta dificultad');
        leer linea(decision);
        Limpiar pantalla;
    FIN;
    hasta (decision='1');//para que el usuario decida
si quiere volver a jugar o quiere reiniciar el programa
    FIN;
    '2': INICIO
        Escribir linea('Usted ha elegihacer la dificultad
Intermedio');
        escribir linea('Diviertase, presione cualquier
tecla para iniciar');
        readkey;
        Limpiar pantalla;
        repetir
            INICIO
                intermedio;
                escribir linea('Ha terminado la partida');
                escribir linea('Presione 1 si
quiere volver a la pantalla de inicio');
                escribir linea('Presione cualquier
otra tecla para seguir jugando en esta dificultad');

```

```

leer linea(decision);
Limpiar pantalla;
FIN;
hasta (decision='1');//para que el usuario decida
si quiere volver a jugar o quiere reiniciar el programa
FIN;
'3': INICIO
Escribir linea('Usted ha elegido la
dificultad difcill');
escribir linea('Diviertase, presione
cualquier tecla para iniciar');
readkey;
Limpiar pantalla;
repetir
INICIO
dificil;
escribir linea('Ha terminado la
partida');
escribir linea('Presione 1 si
quiere volver a la pantalla de inicio');
escribir linea('Presione cualquier
otra tecla para seguir jugando en esta dificultad);
leer linea(decision);
Limpiar pantalla;
FIN;
hasta (decision='1');//para que el usuario decida
si quiere volver a jugar o quiere reiniciar el programa
FIN;
FIN;
FIN;
FIN;

```

hasta (opcion='2');

FIN.