

A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the date.

1/19/2022

Vuelta atrás

Practica Tema 3

Several thin, curved lines in dark blue and light grey originate from the bottom left corner and curve upwards and to the right.

Alejandro Colmenero Moreno
ALGORÍTMIA

ÍNDICE

APARTADO 1..... 2

 CÓDIGO 3

APARTADO 2..... 6

 CÓDIGO 6

APARTADO 3..... 9

 CÓDIGO 9

BIBLIOGRAFÍA..... 12

APARTADO 1

Instancio una matriz de 9x9. La relleno de ceros. Voy pidiendo las 81 casillas una a una por teclado. Si el número recibido no está entre 1 y 9, te lo pide otra vez dándote un error. Escribo en el archivo "tablero.DAT" todos los datos de la matriz.

```
C:\Users\pycol\OneDrive\Escritorio\Formaci3/4n\ExpertoUniversitario_Pr  
ERROR: Numero entre 1 y 9 por favor.  


|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 4 | 5 | 4 | 6 | 6 | 7 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

  
Dame el elemento de la coordenada 1/0: 98
```

```
C:\Users\pycol\OneDrive\Escritorio\Formaci3/4n\ExpertoUniv  


|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 0 |

  
Dame el elemento de la coordenada 8/8: 7
```



```

int num;
for(int x = 0; x < TAM; x++){
    for(int y = 0; y < TAM; y++){

        show_matriz(matriz);
        printf ("Dame el elemento de la coordenada %d/%d: ", x, y) ;
        scanf ("%d", &num );

        system("cls");
        if(num >= 1 && num <= 9){
            matriz[x][y] = num;

        } else {
            printf("ERROR: Numero entre 1 y 9 por favor.\n");
            y--;
        }

    }
}

show_matriz(matriz);
create_file(matriz);

return 0;
}

void create_file(int[TAM][TAM]);
void create_file(int matriz[TAM][TAM]){
    // MATRIZ A TXT
    char txt[TAM+1];

    // ESCRIBIMOS EN EL ARCHIVO BINARIO
    FILE *fileW;
    fileW = fopen(FILENAME, "a");
    if(fileW != NULL){

        for(int x = 0; x < TAM; x++){
            for(int y = 0; y < TAM; y++){
                int index = (x*TAM)+y;
                txt[y]= matriz[x][y] + '0';
            }
            txt[TAM] = '\n';
            fwrite(txt, sizeof txt, 1, fileW);
            free(txt);
        }
    }
}

```

```

        fclose(fileW);

        printf("\nArchivo creado correctamente.\n");

    } else {
        printf("Error al abrir el archivo.");
    }
}

void show_matriz(int[TAM][TAM]);
void show_matriz(int matriz[TAM][TAM]){
    int index = 0;
    for(int x = 0; x < TAM+4; x++){
        for(int y = 0; y < TAM+4; y++){
            if(x == 0 || x == TAM+3 || x == 4 || x == 8){
                // conseguir cuadrado perfecto
                if(y < 9){
                    printf("%c%c%c", 219, 219, 219);
                } else {
                    printf("%c", 219, 219);
                }
            } else if(y == 0 || y == TAM+3 || y == 4 || y == 8){
                printf("%c", 219);
            } else {

                int rx,ry;
                ry = index%TAM;
                rx = index/TAM;
                printf(" %d ", matriz[rx][ry]);
                index++;
            }
        }
        printf("\n");
    }
}

```

APARTADO 2

He quitado todas las apariciones de la variable "iter". Tras esto, en un bucle while del que no se puede salir si no escribes 2 números entre el 1 y 8, he dado valor a fila y columna con dichos números, haciendo la función de las coordenadas del caballo.

```
C:\Users\pycol\OneDrive\Escritorio\Formaci34n\ExpertoUniversitario_Programacion.
Dame la coordenada x (n entre 1 y 8): 8
Dame la coordenada y (n entre 1 y 8): 8

SOLUCION
Columna -> 7 Fila -> 7

53 | 34 | 55 | 30 | 51 | 32 | 15 | 18 |
56 | 49 | 52 | 33 | 16 | 19 | 6 | 13 |
35 | 54 | 29 | 50 | 31 | 14 | 17 | 4 |
48 | 57 | 36 | 41 | 20 | 5 | 12 | 7 |
37 | 28 | 47 | 58 | 43 | 22 | 3 | 64 |
46 | 59 | 42 | 21 | 40 | 11 | 8 | 23 |
27 | 38 | 61 | 44 | 25 | 2 | 63 | 10 |
60 | 45 | 26 | 39 | 62 | 9 | 24 | 1 |
```

CÓDIGO

```
#include <stdio.h>
#include <stdlib.h>
#define tam 8

int main (){
    int tablero [tam][tam];
    int columna = 0;
    int fila = 0;

    carga_tablero (tablero) ;

    int x = 0, y = 0;
    while(1){
        printf ("Dame la coordenada x (n entre 1 y 8): ") ;
        scanf ("%d", &x );
        printf ("Dame la coordenada y (n entre 1 y 8): ") ;
        scanf ("%d", &y );
        if(x < 1 || y < 1 || x > tam || y > tam){
            printf("ERROR\n");
        }
    }
}
```

```

        } else {
            columna = x-1;
            fila = y-1;
            break;
        }
    }

    tablero [columna] [fila]=1;
    recursivo (tablero, 2, columna, fila);

    printf ("\nSOLUCION \nColumna -> %d Fila -> %d \n\n", columna, fila);
    mostrar (tablero) ;
    return 0;
}

void carga_tablero (int [][][tam] );
void carga_tablero (int array[][tam] ){

    int j, i;
    for (j=0; j<tam; j++){
        for (i=0; i<tam; i++){
            array[j][i] = 0;
        }
    }
}

void mostrar (int[] [tam] );
void mostrar (int array [] [tam] ){
    int j, i;
    for (i=0; i<tam; i++){
        for (j=0; j<tam; j++){
            printf (" %2d |", array[j] [i]);
        }
        printf ("\n\n") ;
    }
}

int final (int array [] [tam] )
{
    int j, i;
    for (i=0; i<tam; i++){
        for (j=0; j<tam; j++){
            if (array [j] [i] == 0){
                return 0;
            }
        }
    }
}

```



```

    return 1;
}

void recursivo (int tab[] [tam] , int numero, int x, int y){
    int arrx[] = { 1, 2, 2, 1, -1, -2, -2, -1 };
    int arry[] = { -2, -1, 1, 2, 2, 1,-1,-2 };
    int newx = 0;
    int newy = 0;
    int indice = 0;

    while (indice < 8 && final(tab) != 1){
        newx = arrx [indice] + x;
        newy = arry [indice] + y;
        if ((tab [newx] [newy] == 0) && (newx < tam && newx >= 0 && newy
< tam && newy >= 0)){
            tab [newx] [newy] = numero;
            if (numero != tam*tam){

                recursivo (tab, numero+1, newx, newy) ;
            }
        }

        indice+=1;
    }

    if (final (tab) != 1){
        tab [x] [y]=0;
    }
}

```

APARTADO 3

Simplemente he cambiado todos los números mágicos “8” por una constante del mismo valor y todos los “7” por dicha constante menos 1.

C:\Users\pycol\OneDrive\Escritorio\Formaci34n\ExpertoUniversitario_ProgramacionAvanzada\Algoritmia\PracticaT3\Ejercicio3.exe

Introduce N: 12

#####-Tablero-#####

```
* 0 0 0 0 0 0 0 0 0 0 0 0
0 0 * 0 0 0 0 0 0 0 0 0 0
0 0 0 0 * 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 * 0 0 0 0
0 0 0 0 0 0 0 0 0 0 * 0 0
0 0 0 0 0 0 0 0 0 0 0 0 *
0 0 0 0 0 * 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 * 0 0
0 * 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 * 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 * 0 0 0 0
0 0 0 * 0 0 0 0 0 0 0 0 0 0
```

Coordenadas

```
0,0
1,2
2,4
3,7
4,9
5,11
6,5
7,10
8,1
9,6
10,8
11,3
```

CÓDIGO

```
#include <stdio.h>
#include <stdlib.h>

int *crearVec (int t) ;
void llenarPositivo (int *v, int t) ;
void reinas (int i, int *vec, int *puesto, int *diag2_1, int *diag1_2) ;
void imp(int *vec) ;
int reina=0;
int N = 10;

int main (){
    int a;
    printf ("\nIntroduce N: ") ;
    scanf ("%d", &N) ;
    //N = a;

    int *vec=NULL, *puesto=NULL, *diag2_1=NULL, *diag1_2=NULL;
    vec = crearVec (N) ;
    puesto = crearVec (N) ;
    diag2_1 = crearVec (15) ;
```

```

    diag1_2 = crearVec (15) ;
    llenarPositivo (puesto, N) ;
    llenarPositivo (diag2_1, 15) ;
    llenarPositivo (diag1_2, 15) ;
    reinas (0, vec, puesto, diag2_1, diag1_2) ;
    imp (vec) ;
    return 0;
}

int *crearVec (int t){
    int *v=NULL;
    if( (v=malloc (sizeof (int) *t) )==NULL){
        printf ("Sin memoria suficiente\n") ;
        return NULL;
    }
    return v;
}

void llenarPositivo (int *v, int t){
    int i;
    for (i=0; i<t; i++){
        v[i] = 1;
    }
}

void reinas (int i, int *vec, int *puesto, int *diag2_1, int *diag1_2){
    int j;
    for (j=0; j<N; j++){
        if (puesto[j] && diag2_1[i+j] && diag1_2[i-j+N-1]){
            vec [i] = j;
            puesto [j] = 0;
            diag2_1 [i+j] = 0;
            diag1_2 [i-j+N-1] = 0;
            reina++;
            if (i<N-1){
                reinas (i+1, vec, puesto, diag2_1, diag1_2) ;
                if (reina<N){
                    puesto [j] = 1;
                    diag2_1[i+j] = 1;
                    diag1_2 [i-j+N-1] = 1;
                    reina--;
                }
            }
        }
    }
}

```

```

}

void imp (int *vec){
    int i, j;
    printf ("\n#####-Tablero-#####\n") ;
    for (i=0; i<N;i++){
        for (j=0;j<N;j++){
            if (j==vec [i]){
                printf(" * ") ;
            }else{
                printf(" 0 ") ;
            }
        }
        printf("\n");
    }

    printf ("\n") ;
    printf ("\nCoordenadas\n") ;
    for (i=0; i<N; i++){
        for (j=0; j<N;j++){
            if (j==vec [i]){
                printf ("%d,%d\n", i, j) ;
            }
        }
    }
}
}

```

BIBLIOGRAFÍA

<https://www.tutorialkart.com/c-programming/c-delete-file/>

<https://www.delftstack.com/howto/c/convert-int-to-char/>