

Enunciado

En esta práctica se pretende simular el juego del Mastermind con las siguientes características:

Se jugará con 5 valores numéricos, comprendidos entre 1 y 9, **sin valores repetidos**.

El usuario deberá acertar cuál ha sido la combinación generada por el programa, combinación "secreta", para ello el programa le solicitará una combinación de 5 valores y comprobará si coincide o no con la combinación secreta, no se dispondrá de más de X oportunidades para acertarla, X es una constante definida con valor 10.

Para facilitar la tarea al usuario, el programa después de leer una propuesta, imprimirá el vector de "pistas", que tendrá un '+' por cada número acertado en el lugar correcto y un '*' por cada número acertado pero puesto en un lugar incorrecto.

Si el usuario acierta la combinación, el programa imprimirá un mensaje de felicitación, en el caso de acabar los X intentos sin acertar la combinación imprimirá un mensaje indicándolo, y le dirá cuál es la combinación secreta.

Deberán utilizarse, **al menos**, las siguientes funciones:

- **Función** que **lee** en un vector la combinación propuesta por el usuario, esta función **validará** la combinación propuesta, es decir, comprobará que los 5 valores están entre 1 y 9, y que no están repetidos, en caso de no ser así solicitará otra combinación de valores.
- **Función** que **genera e imprime** el vector de pistas a partir de la combinación secreta y la combinación propuesta, el vector generado tendrá tantos '+' como números acertados en su correcta posición, y tantos '*' como números acertados pero situados en posiciones incorrectas. Esta función **también devuelve** el total de valores acertados en su posición correcta.
- **Función** que **escribe** un vector con las características del vector que contiene la combinación secreta.

Se sugiere hacer el programa en primer lugar asignando una combinación secreta en la declaración del vector, por ejemplo: `int secreto[] = {6, 4, 1, 3, 8};`

Cuando el programa funcione correctamente se deberá crear una **función** que genere un vector con números aleatorios comprendidos entre 1 y 9 y sin números repetidos, para ello, se deberá utilizar las siguientes bibliotecas estándar y funciones:

```
#include <stdlib.h>    // libreria para el uso de rand()
#include <time.h>       // libreria para el uso de time()

srand(time(NULL));    //genera la "semilla" de valores pseudoaleatorios
num=rand()%MAX+MIN;   //almacena en num un valor comprendido entre MIN y MAX
```

Ejemplo de ejecución, si la combinación secreta es: {7, 4, 5, 3, 2}:

Escribe una combinación
(5 valores entre 1 y 9): 3 4 5 6 7
Pistas: + + * *

Escribe una combinación
(5 valores entre 1 y 9): 8 4 3 2 9
Pistas: + * *

...

Explicación:

Hay dos + por el 4 y el 5 que están en su lugar
Hay dos * por el 3 y el 7 que no están en su lugar

Hay un + por el 4 que está en su lugar
Hay dos * por el 3 y el 2 que no están en su lugar

SOLUCIÓN:

```
#include <stdio.h>
#include <stdlib.h> // libreria para el uso de rand()
#include <time.h>    // libreria para el uso de time()

#define N 5
#define INTENTOS 10
#define MAX 9
#define MIN 1

int darPistas (int propuesta[N], int secreto[N], char pista [N]);
void escribir (int v[N]);
void leerDistintos (int v[N]);
void copiarVector (int origen[N], int destino[N]);
int buscarEnVect (int v[], int nElem, int num);
void creaVector (int v[N]);

int main ()
{
    int secreto[N];
    int propuesta[N];
    int cont, aciertos;
    char pista [N];

    creaVector(secreto);

    printf("\n\n");
    printf("        Juego del Mastermind\n");
    printf("    ----- \n");
    cont=0;
    do
    {
        leerDistintos (propuesta);
        aciertos= darPistas (propuesta, secreto,  pista );
        cont ++;
    }
    while (cont<INTENTOS && aciertos!=N);

    if (aciertos == N) printf ("ENHORABUENA!!!");
    else
    {
        printf("No conseguiste acertar,  la combinacion secreta es: ");
        escribir (secreto);
    }
    return 0;
}

int darPistas (int propuesta[N], int secreto[N], char pista [N])
// genera e imprime el vector con las pistas y devuelve la cantidad de aciertos
{
    int i, j, valor, cont1=0, cont2=0;
```

```
// cont1 = contador de aciertos
// cont2 = contador de numeros en el vector en distinto lugar
for (i=0; i<N; i++)
{
    valor = buscarEnVect(secreto, N, propuesta[i]);
    if (valor==i)
        cont1++;
    else if (valor != -1) cont2++;
}

for (i=0; i<cont1; i++)
    pista[i]='+';
for (i=cont1; i<cont1+cont2; i++)
    pista[i]='*';

printf("\t\t Pistas: ");
for (i=0; i<cont1+cont2; i++ )
    printf ("%c ", pista[i]);
printf("\n\n");
return cont1;
}

void escribir (int v[N])
// escribe el vector secreto o el propuesto por el usuario
{
    int i;
    printf("{ ");
    for (i=0; i<N; i++ )
        printf ("%d ", v[i]);
    printf(" }\n");
}

void leerDistintos (int v[N])
// lee la combinacion del usuario validando los datos
{
    int i, j, esValida;
    do
    {
        esValida = 1;
        printf("Escribe tu combinacion \n(%d valores entre %d y %d): ",N,MIN,MAX);
        i=0;
        while (i<N && esValida)
        {
            scanf ("%d", &v[i]);
            esValida= (v[i]>=MIN && v[i]<=MAX);
            esValida= esValida && (buscarEnVect(v,i,v[i])!= -1);
            i++;
        }
        //solo se lee hasta el primer valor incorrecto
        //ponemos el fflush para ignorar los valores no leídos
        fflush(stdin);
        if (!esValida) printf ("\tCombinacion incorrecta\n\n");
    }
    while (!esValida);
}
```

```
void creaVector (int v[N])
// genera la combinación secreta validando los datos
{
    /* Rellena el vector V con valores distintos aleatorios */
    int i=0, num;
    srand(time(NULL));    //genera la "semilla" de valores pseudoaleatorios
    while (i<N)
    {
        num = rand()%MAX+MIN;
        if (buscarEnVect (v,i,num)==-1)
        {
            v[i]=num;
            i++;
        }
    }
}

int buscarEnVect (int v[], int nElem, int num)
// Busca num en el vector v hasta la componente nElem
//devuelve el lugar del vector deonde lo encuentra 0
//si no lo encuentra devuelve -1
{
    int i=0, encont=0;
    while (i<nElem && !encont)
    {
        if (v[i] != num)
            i++;
        else encont =1;
    }
    if (encont)
        return i;
    else return -1;
}
```