GRUPO DE PRÁCTICAS: **8.**

ALUMNO: **VÍCTOR MONSERRAT VILLATORO.**

7. En un curso se han realizado dos exámenes diferentes, Ay B, entre sus 50 alumnos (alumnos pares, examen A; alumnos impares, examen B. Implementa una función que calcule la nota media, máxima y mínima de cada examen.

- **DATOS.**

  - **ENTRADA:**

    Notas de los exámenes: reales, entrada por teclado.

  - **SALIDA:**

    Nota media de los alumnos del examen A: real, salida por pantalla.
    Nota media de los alumnos del examen B: real, salida por pantalla.
    Nota máxima de los alumnos del examen A: real, salida por pantalla.
    Nota máxima de los alumnos del examen B: real, salida por pantalla.
    Nota mínima de los alumnos del examen A: real, salida por pantalla.
    Nota mínima de los alumnos del examen B: real, salida por pantalla.

  - **OTROS:**

    Constante: DIM_VECTOR (dimensión del vector) = 50.

- **RESTRICCIONES.**

  Las notas introducidas tienen que ser positivas y menores que 10.

- **OBTENCIÓN DEL RESULTADO.**

  La nota media del examen A será la suma de todos los elementos pares del vector de notas dividido entre la mitad del tamaño del vector (25), de la misma manera pero con los impares, se obtiene la nota media del examen B.
  La nota máxima del examen A se obtendrá comprobando dos a dos a través de un bucle los elementos pares del vector notas, de la misma manera pero con los impares, se obtiene la nota máxima del examen B.
  La nota mínima del examen A se obtendrá comprobando dos a dos a través de un bucle los elementos pares del vector notas, de la misma manera pero con los impares, se obtiene la nota mínima del examen B.

- **EJEMPLO.**

  Vector notas [50] = {5, 8, 3, 4, 7, 6, ...}
  Nota media del examen A: 5 + 3 + 7 + … / 25.
  Nota media del examen B: 8 + 4 + 6 + … /25.
  Nota máxima del examen A: 7 (sin tener en cuenta las notas no especificadas).
  Nota máxima del examen B: 8 (sin tener en cuenta las notas no especificadas).
  Nota máxima del examen A: 3 (sin tener en cuenta las notas no especificadas).
  Nota máxima del examen B: 4 (sin tener en cuenta las notas no especificadas).

- **PSEUDOCÓDIGO.**

```
[1]  Función leer_vector (vector [])
[2]  Inicio
[3]     Para contador de 1 hasta (DIM_VECTOR)
[4]          Escribir "Nota del alumno (contador): "
[5]          Leer (vector [contador])
[6]     Mientras (vector [contador] < 0) o (vector [contador] > 10)
[7]          Escribir "Error: La nota debe ser un número entre 0 y 10. Vuelva a
     introducir la nota del alumno (contador): "
[8]          Leer (vector [contador])
        Finmientras
[9]  Fin
[10]
[11] Función media (vector [], media_pares, media_impares)
[12] Inicio
[13]    Para contador de 0 hasta DIM_VECTOR
[14]         suma_pares ← suma_pares + vector [contador]
[15]         contador ← contador +1
[16]    Finpara
[17]    media_pares ← suma_pares/(DIM_VECTOR/2)
[18]    Para contador de 1 hasta DIM_VECTOR
[19]         suma_impares ← suma_impares + vector [contador]
[20]         contador ← contador +1
[21]    Finpara
[22]    media_impares ← suma_impares/(DIM_VECTOR/2)
[23] Función maximo (vector [], maximo_pares, maximo_impares)
[24] Inicio

[25]    maximo_pares ← vector [0]
[26]    Para contador de 2 hasta DIM_VECTOR
[27]         Si vector [contador] > maximo_pares
[28]              maximo_pares ← vector [contador]
[29]         Finsi
[30]         contador ← contador + 1
[31]    Finpara
[32]    Para contador de 3 hasta DIM_VECTOR
[33]         Si vector [contador] > maximo_impares
[34]              maximo_impares ← vector [contador]
[35]         Finsi
[36]         contador ← contador + 1
[37]    Finpara
[38] Fin
[39]
[40] Función minimo (vector [], minimo_par, minimo_impares)
[41] Inicio
[42]    minimo_pares ← vector [0]
[43]    Para contador de 2 hasta DIM_VECTOR
[44]         Si vector [contador] < minimo_pares
[45]              minimo_pares ← vector [contador]
[46]         Finsi
             contador ← contador + 1
```

```
[47]    Finpara
[48]    minimo_impares ← vector [1]
[49]    Para contador 3 hasta DIM_VECTOR
[50]            Si vector [contador] < minimo_impares
[51]                    minimo_impares ← vector [contador]
[52]            Finsi
[53]            contador ← contador + 1
[54]    Finpara
[55] Fin
[56]
[57] Inicio
[58]    Llamada a la función leer_vector (vector)
[59]    Llamada a la función media (vector, media_pares, media_impares)
[60]    Llamada a la función maximo (vector, maximo_pares, maximo_impares)
[61]    Llamada a la función minimo (vector, minimo_pares, minimo_impares)
[62]    Escribir “La nota media de los alumnos del examen A es: (media_pares).”
[63]    Escribir “La nota media de los alumnos del examen B es: (media_impares).”
[64]    Escribir “La nota máxima de los alumnos del examen A es: (maximo_pares).”
[65]    Escribir “La nota máxima de los alumnos del examen B es: (maximo_impares).”
[66]    Escribir “La nota mínima de los alumnos del examen A es: (minimo_pares).”
[67]    Escribir “La nota mínima de los alumnos del examen B es: (minimo_pares).”
```

- **CÓDIGO EN C.**

```
#include <stdio.h>
#define DIM_VECTOR 50

void leer_vector (float vector [])
{
        int i;

        for (i=0; i<DIM_VECTOR; i++)
        {
                printf("\nNota del alumno %i: ", i);
                scanf("%f", &vector [i]);
                while (vector [i] < 0) || (vector [i] > 10)
                {
                        printf("\nError: La nota debe ser un número entre 0 y 10. Vuelva
a introducir la nota del alumno %i: ", i);
                        scanf("%f", &vector [i]);
                }
        }
}

void media (float vector [], float* media_pares, float* media_impares)
{
        int i;
        float suma_pares = 0, suma_impares = 0;

        for (i=0; i<DIM_VECTOR; i++)
        {
                suma_pares = suma_pares + vector [i];
```

```
            i++;
        }
        *media_pares = suma_pares/(DIM_VECTOR/2);

        for (i=1; i<DIM_VECTOR; i++)
        {
            suma_impares = suma_impares + vector [i];
            i++;
        }
        *media_impares = suma_impares/(DIM_VECTOR/2);
}

void maximo (float vector [], float* maximo_pares, float* maximo_impares)
{
        int i;

        *maximo_pares = vector [0];
        for (i=2; i<DIM_VECTOR; i++)
        {
            if ((vector [i]) > (*maximo_pares))
            {
                *maximo_pares = vector [i];
            }
            i++;
        }
        *maximo_impares = vector [1];
        for (i=3; i<DIM_VECTOR; i++)
        {
            if ((vector [i]) > (*maximo_impares))
            {
                *maximo_impares = vector [i];
            }
            i++;
        }
}

void minimo (float vector [], float* minimo_pares, float* minimo_impares)
{
        int i;

        *minimo_pares = vector [0];
        for (i=2; i<DIM_VECTOR; i++)
        {
            if ((vector [i]) < (*minimo_pares))
            {
                *minimo_pares = vector [i];
            }
            i++;
        }

        *minimo_impares = vector [1];
        for (i=3; i<DIM_VECTOR; i++)
```

```
        {
                if ((vector [i]) < (*minimo_impares))
                {
                        *minimo_impares = vector [i];
                }
                i++;
        }
}

int main ()
{

        float vector [DIM_VECTOR], media_pares, media_impares, maximo_pares,
maximo_impares, minimo_pares, minimo_impares;

        leer_vector (vector);
        media (vector, &media_pares, &media_impares);
        maximo (vector, &maximo_pares, &maximo_impares);
        minimo (vector, &minimo_pares, &minimo_impares);
        printf("\nLa nota media de los alumnos del examen A es: %f.", media_pares);
        printf("\nLa nota media de los alumnos del examen B es: %f.",
media_impares);
        printf("\nLa nota máxima de los alumnos del examen A es: %f.",
maximo_pares);
        printf("\nLa nota máxima de los alumnos del examen B es: %f.",
maximo_impares);
        printf("\nLa nota mínima de los alumnos del examen A es: %f.",
minimo_pares);
        printf("\nLa nota mínima de los alumnos del examen B es: %f.\n\n",
minimo_impares);
        return (0);
}
```

9. Implementa una función que responda al siguiente prototipo: *int es_prefijo (char *cadena, char *prefijo)*, que compruebe si una cadena es prefijo de otra. La función devolverá 1 si es prefijo y 0 en otro caso. Utiliza la función *strstr* de la bibliotecas <*string.h*>.

- **DATOS.**
    - **ENTRADA:**

      Cadena: cadena de caracteres, entrada por teclado.
      Prefijo: cadena de caracteres, entrada por teclado.

    - **SALIDA:**

      Se mostrará en pantalla si el prefijo lo es de la cadena o no.

    - **OTROS:**

      Constante: DIM_CHAR (dimensión de la cadena de caracteres) = 100.

- **RESTRICCIONES.**

- **OBTENCIÓN DEL RESULTADO.**

  A través de la función strstr que devuelve un puntero NULL si no encuentra el prefijo en la cadena y un valor si lo encuentra. Esta función la igualamos a un puntero y comprobamos si a la dirección de memoria de este puntero restándole la propia dirección de memoria de la cadena es igual a 0, entonces será un prefijo de la cadena porque serán la misma dirección ya que devuelve un puntero a la primera aparición de prefijo en cadena, sino no.

- **EJEMPLO.**

  Cadena cadena [DIM_CHAR] = "deshacer"
  Cadena prefijo [DIM_CHAR] = "des"
  El programa imprimirá que "des" es un prefijo de "deshacer".

  Aunque:

  Cadena cadena [DIM_CHAR] = "deshacer"
  Cadena prefijo [DIM_CHAR] = "hacer"
  El programa imprimirá que "hacer" NO es un prefijo de "deshacer".

- **PSEUDOCÓDIGO.**

```
[1]  es_prefijo (*cadena, *prefijo)
[2]  Inicio
[3]      ptr ← strstr (cadena, prefijo)
[4]      Si ptr – cadena = 0
[5]              Devuelve 1
[6]      Sino
```

[7] Devuelve 0
[8] Finsi
[9] Fin
[10]
[11] Inicio
[12] Escribir “Introduzca la palabra para ver si contiene un prefijo: “
[13] Leer (cadena)
[14] Escribir “Introduzca el prefijo: “
[15] Leer (prefijo)
[16] resultado ← es_prefijo (cadena, prefijo)
[17] Si resultado = 0
[18] Escribir “El prefijo “(prefijo)” NO es prefijo de la palabra “(cadena)”.
[19] Finsi
[20] Sino
[21] Escribir “El prefijo “(prefijo)” es prefijo de la palabra “(cadena)”.
[22] Fin

- **CÓDIGO EN C.**

```
#include <stdio.h>
#include <string.h>
#define DIM_CHAR 100

int es_prefijo (char *cadena, char *prefijo)
{
        char *ptr;

        ptr = strstr (cadena, prefijo);
        if (ptr-cadena == 0)
        {
                return (1);
        }
        else
        {
                return (0);
        }
}

int main ()
{
        char cadena [DIM_CHAR], prefijo [DIM_CHAR];
        int resultado;

        printf("\nIntroduzca la palabra para ver si contiene un prefijo: ");
        fgets (cadena, DIM_CHAR, stdin);
        cadena [strlen (cadena) -1] = '\0';
        printf("\nIntroduzca el prefijo: ");
        fgets (prefijo, DIM_CHAR, stdin);
        prefijo [strlen (prefijo) -1] = '\0';
        resultado = es_prefijo (cadena, prefijo);
        if (resultado == 0)
        {
```

```
                printf("\nEl prefijo \"%s\" NO es prefijo de la palabra \"%s\".\n\n",
prefijo, cadena);
        }
        else
        {
                printf("\nEl prefijo \"%s\" es prefijo de la palabra \"%s\".\n\n", prefijo,
cadena);
        }
        return (0);
}
```