NOMBRE Y APELLIDOS: GREGORIO CORPAS PRIETO
GRUPO: GP8

1. ENUNCIADO:
   ▶ En un curso se han realizado dos exámenes diferentes, A y 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.

2. DATOS DE ENTRADA:
   ▶ Supondremos el conjunto de exámenes como un array de 50 elementos ya relleno.
      - Nombre: examenes
      - Correspondencia: Estructura almacenadora de notas de exámenes
      - Tipo de dato: Flotante
      - Restricciones: Ninguna

3. RESULTADOS O SALIDA:
   ▶ El problema planteado requiere el retorno de 3 valores estadísticos acerca de las notas (media, máxima, mínima)
      - Nombre: media, maxima, minima
      - Correspondencia: Notas media, máxima y mínima del conjunto de exámenes
      - Tipo de dato: Flotante

   ▶ Una vez obtenidos los valores de las variables se mostrarán impresas por pantalla en la terminal

4. DATOS AUXILIARES:
   ▶ Para poder implementar las funcionalidades de media, máxima y mínima necesitaremos variables auxiliares del mismo tipo para ir almacenando y comparando tanto mínimos como máximos, y también el sumatorio de notas para calcular la media.

5. DESCRIPCIÓN DEL ALGORITMO:
   ▶ Debido a que el problema pide que en una única función o método calculemos varios valores de una vez, usaremos funciones void con parámetros de llamada y retorno usando paso por referencia.

   ▶ Por modularizar las funcionalidades he decidido implementar la solución usando métodos particulares para cada cálculo estadístico que hagan retornos tipo flotante, aunque bien podría haber hecho las modificaciones directamente en las variables pasando sus direcciones desde CalculaEstadísticas a cualquier método particular por referencia, o incluso ahorrarme dicha modularidad y calcularlo todo en el método CalculaEstadísticas.

   ▶ La idea consiste en crear un método principal al que llamaremos calculaEstadisticas que recibirá el vector de notas y las posiciones de memoria de las variables donde se servirán los valores de retorno.

   ▶ Dentro de esta función CalculaEstadísticas llamaremos individualmente a los métodos calculaMedia, calculaMaxima y calculaMinima. Dichos métodos recibirán por parámetro el vector con las notas y mediante un return devolverán el valor exigido.

   ▶ Una vez calculado todo se rellenarán las variables pasadas por

referencia en calculaEstadisticas y posteriormente desde el programa principal se imprimirán los valores finales.

6. PSEUDOCÓDIGO

```
PROGRAMA PRINCIPAL:

        FLOTANTE MEDIA, MAXIMA, MINIMA, NOTAS[50]
        ENTERO I

        PARA ( I = 0, MIENTRAS QUE I < 50, I++ )
                NOTAS[I] = RAND()%1000 / 100
        FIN PARA

        CALCULAESTADISTICAS(&NOTAS, &MEDIA, &MAXIMA, &MINIMA)

        IMPRIME( LA NOTA MEDIA ES [MEDIA] LA NOTA MAXIMA ES [MAXIMA] LA NOTA MINIMA ES [MINIMA] )

FIN PROGRAMA PRINCIPAL

CALCULAESTADISTICAS ( *NOTAS, *MEDIA, *MAXIMA, *MINIMA ):

        *MEDIA = CALCULAMEDIA(NOTAS)
        *MAXIMA = CALCULAMAXIMA(NOTAS)
        *MINIMA = CALCULAMINIMA(NOTAS)

FIN CALCULAESTADISTICAS

CALCULAMEDIA ( *NOTAS ):

        ENTERO I
        FLOTANTE AUXILIAR=0

        PARA ( I = 0, MIENTRAS QUE I < 50, I++ )
                AUXILIAR += NOTAS[I]
        FIN PARA

        RETURN AUXILIAR / 50

FIN CALCULAMEDIA

CALCULAMAXIMA( *NOTAS ):

        ENTERO I
        FLOTANTE AUXILIAR=0

        PARA ( I = 0, MIENTRAS QUE I < 50, I++ )
                SI AUXILIAR < NOTAS[I]
                        AUXILIAR = NOTAS[I]
                FIN SI
        FIN PARA

        RETURN AUXILIAR

FIN CALCULAMAXIMA

CALCULAMINIMA( *NOTAS ):

        ENTERO I
        FLOTANTE AUXILIAR=10

        PARA ( I = 0, MIENTRAS QUE I < 50, I++ )
                SI AUXILIAR > NOTAS[I]
                        AUXILIAR = NOTAS[I]
                FIN SI
        FIN PARA

        RETURN AUXILIAR

FIN CALCULAMINIMA
```

1. ENUNCIADO:
   - ▶ 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 si no lo fuera. Utiliza la función strstr de la biblioteca <string.h>

2. DATOS DE ENTRADA:
   - ▶ Tendremos 2 vectores de char, uno para la cadena y otro para la subcadena
     - Nombre: cadena, subcadena
     - Correspondencia: cadena de caracteres y prefijo que buscar
     - Tipo de dato: * char
     - Restricciones: Ninguna

3. RESULTADOS O SALIDA:
   - ▶ El retorno será una impresión por pantalla que indique si la subcadena es prefijo de la cadena

4. DATOS AUXILIARES:
   - ▶ Ninguno en particular

5. DESCRIPCIÓN DEL ALGORITMO:
   - ▶ Se pedirá por teclado al usuario introducir la cadena de texto
   - ▶ Se pedirá por teclado al usuario introducir la subcadena prefijo
   - ▶ Se pasarán ambas cadenas (vectores de caracteres) por referencia a la función esPrefijo, donde serán comparadas y analizadas.
   - ▶ Con la función strstr comprobaremos en primera instancia si la subcadena se encuentra en la cadena, y además, dicha función devuelve la posición de memoria de donde se encontró la primera coincidencia, por tanto la compararemos con la posición 0 del array de caracteres cadena, o lo que es lo mismo, con cadena. (cadena = &cadena[0])
   - ▶ Si son prefijo o no será cuestionado en el programa principal mediante el retorno de la función esPrefijo que será comparado con 1, y el resultado final se indicará mediante mensaje por pantalla en la terminal

6. PSEUDOCÓDIGO

```
PROGRAMA PRINCIPAL:

        CARACTER CADENA[30], SUBCADENA[20]

        IMPRIME(INTRODUCE UNA CADENA DE TEXTO)
        LEE(CADENA)
        IMPRIME(INTRODUCE UN PREFIJO A BUSCAR EN LA CADENA)
        LEE(SUBCADENA)

        SI (ESPREFIJO(CADENA,SUBCADENA) == 1)
                IMPRIME(COINCIDENCIA ENCONTRADA, [SUBCADENA] ES PREFIJO DE [CADENA])

        SINO
                IMPRIME(NO EXISTEN COINCIDENCIAS, UNA MALA TARDE LA TIENE CUALQUIERA)
        FIN SI

FIN PROGRAMA PRINCIPAL

ESPREFIJO (CARACTER * CADENA, CARACTER * SUBCADENA):
        ENTERO ACIERTO = 0;

        SI ((STRSTR(CADENA,SUBCADENA)) == CADENA)
                ACIERTO = 1
        FIN SI

        RETURN ACIERTO;
FIN ESPREFIJO
```