

# Fonaments de Programació (104337)

Curso 2019 - 2020 - Examen Parcial (29 de octubre de 2019)

Nombre estudiante: Paul Winter Gibson NIU: 9793642

**Importante:** Recuerda que hay que dar las mejores soluciones posibles en cada ejercicio. Además de funcionar correctamente, los procedimientos y funciones deben estar bien programados (utilizando las instrucciones más adecuadas, sin operaciones ni variables innecesarias, etc.)

## Los ejercicios 1 a 7 se refieren a un mismo tema que introducimos a continuación:

Un hospital materno-infantil nos pide hacer un programa para gestionar algunos datos médicos de los bebés nacidos en el hospital. Más concretamente, se quiere guardar y gestionar la información referente a la altura de los bebés al nacer.

El rango de alturas a considerar es el que va desde 31 cm hasta 60 cm. En caso de que el bebé mida menos de 31 cm o más de 60 cm la altura introducirá como 31 o 60 respectivamente.

Las anotaciones siempre se hacen en centímetros (cm) y se guardarán como valores enteros.

Cada bebé está identificado con un código numérico de 5 cifras. La información que se quiere guardar de cada bebé es su código y su altura.

Además, cada bebé se le asigna un grupo (A, B, C, D) en función de la altura que tiene. Este grupo, también se ha de guardar para cada bebé.

Todos estos datos se guardarán en tres listas (una para los identificadores de los bebés, otra por las alturas y otra por el grupo al que pertenece cada bebé). El orden en el que se guardan los datos es el mismo en las tres listas, es decir, el identificador de un bebé, su altura y el grupo al que pertenece, están en el mismo índice de la lista correspondiente.

### Ejercicio 1 (1 punto)

Hacer una función llamada `buscar_elemento` que reciba como parámetro una lista de enteros y un elemento a buscar, y compruebe si el elemento se encuentra en la lista. Si el elemento está en la lista, la función devolverá el índice de la posición donde se ha encontrado el elemento. Si no está, devolverá -1.

**Importante:** NO SE PUEDEN UTILIZAR FUNCIONES DE MANIPULACIÓN DE LISTAS, SALVO LA FUNCIÓN `len()`

$$20 \times 10 = 200$$

$$200 + 50 = \boxed{250}$$

### Ejercicio 2 (1 punto)

Hacer una función llamada `calcular_grupo` que calcule el grupo (A, B, C, D) que se le debe asignar a un bebé en función de su altura. La función recibirá la altura en centímetros de un bebé como parámetro y devolverá el carácter correspondiente al grupo que corresponde a aquella altura según el siguiente criterio:

- 'A' si la altura del bebé es inferior a 35cm
- 'B' si la altura del bebé se encuentra en el intervalo [35cm, 40cm)
- 'C' si la altura del bebé se encuentra en el intervalo [40cm, 50cm)
- 'D' si la altura del bebé es igual o superior a 50cm

**Nota:** Considere que la altura nunca será negativa (no hay que hacer la comprobación).

$$4 \times 3 = 12$$

$$12 \times 2 = 36$$

$$\text{Resultat: } 36$$

### Ejercicio 3 (1 punto)

Hacer un procedimiento llamado `gestionar_datos` que permita introducir los datos de un bebé nuevo. El procedimiento recibirá como parámetros las listas con los identificadores de los bebés, la lista de las alturas de los bebés y la lista del grupo al que pertenece cada bebé.

El procedimiento pedirá el identificador numérico de un bebé y comprobará si se encuentra en la lista de identificadores (con la función `buscar_elemento` del ejercicio 1).

- Si no se encuentra el identificador en la lista, se añadirá el identificador a la lista de identificadores y se pedirá la altura del bebé. Si el valor introducido es menor o igual que cero, se mostrará un mensaje de error ("ERROR: Valor incorrecto por la altura") y se volverá a pedir de introducir la altura del bebé. Este proceso se repetirá hasta que se introduzca un valor correcto.
- Una vez leída una altura correcta, se calculará el grupo de altura (A, B, C, o D) que se le debe asignar (con la función `calcular_grupo` del ejercicio 2) y se añadirán los datos a las listas correspondientes.
- Si el identificador introducido ya se encuentra en la lista de identificadores, se informará al usuario de que ya están los datos de este bebé, y las mostrará por pantalla (altura y grupo).

### Ejercicio 4 (1 punto)

Hacer una función llamada `histograma` que reciba la lista de las alturas de los bebés y haga el recuento de cuántos bebés tienen cada una de las alturas entre 31 y 60. Este recuento se devolverá en una lista que contendrá el número de bebés con altura 31 a la primera posición, el número de bebés con altura 32 en la segunda, y así sucesivamente.

La lista donde se guardará el recuento, se debe crear al principio de la función con ceros en todas las posiciones.

### Ejercicio 5 (1 punto)

Hacer una función llamada `estadistica` que permita obtener algunas medidas estadísticas sobre los datos guardados en una lista. La función recibirá una lista con valores numéricos, y calculará la media y la varianza de los datos.

Para calcular la media, suponga que disponemos de una FUNCIÓN ya implementadas:

`media(lista) →` calcula y devuelve la media de la lista pasada como parámetro

Recuerde que, para calcular la varianza de un conjunto de datos, debe seguir la fórmula:

$$varianza = \frac{1}{N-1} \sum_i (x_i - \mu)^2$$

donde  $x_i$  son los elementos del conjunto de datos,  $\mu$  es la media de los datos y  $N$  es el número de elementos del conjunto de datos.

La función devolverá los dos valores correspondientes a la media y la varianza.

### Ejercicio 6 (1 punto)

Hacer un procedimiento llamado `consultar_grupo` que permita realizar consultas de los datos de los bebés que pertenecen a un determinado grupo (A, B, C, D). El procedimiento recibirá como parámetros la lista con los identificadores de los bebés, la lista con las alturas de los bebés y la lista con el grupo al que pertenecen los bebés.

El procedimiento preguntará qué grupo (A, B, C o D) se quiere consultar ya continuación mostrará los datos (identificador y altura) de todos los bebés que pertenezcan al grupo indicado. Si se introduce un grupo inexistente, se mostrará el mensaje de error: "Error: Grupo inexistente" y se sale del procedimiento (NO se vuelve a preguntar el grupo).

### Ejercicio 7 (2 puntos)

Hacer un programa que siga los siguientes pasos:

1. inicializaciones de variables y constantes
2. Utilizar (NO IMPLEMENTAR) el procedimiento `menu_principal()` que permite imprimir por pantalla el siguiente menú:

```
----- MENU -----
1.- Introducir datos
2.- Histograma de alturas
3.- Estadísticas
4.- Consultar por grupo
5.- Finalizar
```

3. Pedir al usuario que introduzca una de las opciones.
4. Si la opción es 1, llamar la función del ej. 3, `gestionar_datos()`, para introducir los datos de un nuevo bebé.
5. Si la opción es 2:
  - 5.1. llamar la función del ej. 4, `histograma`, para generar el histograma de las alturas de los bebés.
  - 5.2. mostrar cuántos bebés tienen cada una de las alturas con mensajes con el siguiente formato:

```
31 centímetros - X bebés
32 centímetros - Y bebés
...
60 centímetros - Z bebés
```

donde X es el número de bebés con 31cm, Y el número de bebés con 32cm, etc.

6. Si la opción es 3:
  - 6.1. comprobar si ya se han introducido los datos de algún bebé. En caso afirmativo:
    - 6.1.1. llamar la función del ej. 5, `estadistica`, para calcular las medidas estadísticas sobre las alturas de los bebés.
    - 6.1.2. mostrar un mensaje con la media y la varianza de las alturas de los bebés.
  - 6.2. si todavía no se han introducido datos, mostrar un mensaje de error.
7. Si la opción es 4, llamar al procedimiento del ej. 6, `consultar_grupo()`, para hacer la consulta de los bebés que pertenecen a un determinado grupo.
8. Si la opción es 5, mostrar el mensaje: `Saliendo del programa ...`
9. Cualquier otra opción, escribir el mensaje: `Opción no permitida.`
10. Repetir los pasos 2 a 8, hasta que la opción del menú escogida sea la 5.

c7

**Ejercicio 8 (1 punto)**

Hacer un programa que pida una cadena y luego muestre por pantalla cada una de las palabras que la forman en una línea diferente. Considera que todas las palabras de la cadena están separadas entre ellas por un carácter espacio ". No hay que tener en cuenta los signos de puntuación.

Por ejemplo, si la cadena que se entra es "Hola, como estás?", por pantalla debería mostrar:

Hola,  
como  
estás?

**Ejercicio 9 (1 punto)**

¿Cuáles serán los valores de x, y y z al finalizar el programa?

```
def procesar (x, w, lista):  
    for i in range (w):  
        lista.append (x)  
        x += w  
    return x  
  
def f (x, y, z):  
    lista = []  
    z = procesar (x, y, lista)  
    x = len (lista)  
    y += 1  
    return (z, x, y)  
  
x, y, z = f (2,3,2)
```

Valor x	Valor y	Valor z

# Fonaments de Programació (104337)

Curso 2019 - 2020 - Examen Parcial (29 de octubre de 2019)

Nombre estudiante: KASSAPORA UGDAHS NIU: 7282065

**Important:** Recuerda que hay que dar las mejores soluciones posibles en cada ejercicio. Además de funcionar correctamente, los procedimientos y funciones deben estar bien programados (utilizando las instrucciones más adecuadas, sin operaciones ni variables innecesarias, etc.)

## Los ejercicios 1 a 7 se refieren a un mismo tema que introducimos a continuación:

Un hospital materno-infantil nos pide hacer un programa para gestionar algunos datos médicos de los bebés nacidos en el hospital. Más concretamente, se quiere guardar y gestionar la información referente a la altura de los bebés al nacer.

El rango de alturas a considerar es el que va desde 31 cm hasta 60 cm. En caso de que el bebé mida menos de 31 cm o más de 60 cm la altura introducirá como 31 o 60 respectivamente.

Las anotaciones siempre se hacen en centímetros (cm) y se guardarán como valores enteros.

Cada bebé está identificado con un código numérico de 5 cifras. La información que se quiere guardar de cada bebé es su código y su altura.

Además, cada bebé se le asigna un grupo (A, B, C, D) en función de la altura que tiene. Este grupo, también se ha de guardar para cada bebé.

Todos estos datos se guardarán en tres listas (una para los identificadores de los bebés, otra por las alturas y otra por el grupo al que pertenece cada bebé). El orden en el que se guardan los datos es el mismo en las tres listas, es decir, el identificador de un bebé, su altura y el grupo al que pertenece, están en el mismo índice de la lista correspondiente.

### Ejercicio 1 (1 punto)

Hacer una función llamada `buscar_elemento` que reciba como parámetro una lista de enteros y un elemento a buscar, y compruebe si el elemento se encuentra en la lista. Si el elemento está en la lista, la función devolverá el índice de la posición donde se ha encontrado el elemento. Si no está, devolverá -1.

**Importante:** NO SE PUEDEN UTILIZAR FUNCIONES DE MANIPULACIÓN DE LISTAS, SALVO LA FUNCIÓN `len()`

$$\begin{aligned} 50 \times 15 &= 750 \\ 750 - 45 &= 705 \end{aligned}$$

### Ejercicio 2 (1 punto)

Hacer una función llamada `calcular_grupo` que calcule el grupo (A, B, C, D) que se le debe asignar a un bebé en función de su altura. La función recibirá la altura en centímetros de un bebé como parámetro y devolverá el carácter correspondiente al grupo que corresponde a aquella altura según el siguiente criterio:

- 'A' si la altura del bebé es inferior a 35cm
- 'B' si la altura del bebé se encuentra en el intervalo [35cm, 40cm)
- 'C' si la altura del bebé se encuentra en el intervalo [40cm, 50cm)
- 'D' si la altura del bebé es igual o superior a 50cm

**Nota:** Considere que la altura nunca será negativa (no hay que hacer la comprobación).

$$3 + 3 = 6$$

$$6 \times 10 = 60$$

$$\text{Resultado} = 66$$

### Ejercicio 3 (1 punto)

Hacer un procedimiento llamado `gestionar_datos` que permita introducir los datos de un bebé nuevo. El procedimiento recibirá como parámetros las listas con los identificadores de los bebés, la lista de las alturas de los bebés y la lista del grupo al que pertenece cada bebé.

El procedimiento pedirá el identificador numérico de un bebé y comprobará si se encuentra en la lista de identificadores (con la función `buscar_elemento` del ejercicio 1).

- Si no se encuentra el identificador en la lista, se añadirá el identificador a la lista de identificadores y se pedirá la altura del bebé. Si el valor introducido es menor o igual que cero, se mostrará un mensaje de error ("ERROR: Valor incorrecto por la altura") y se volverá a pedir de introducir la altura del bebé. Este proceso se repetirá hasta que se introduzca un valor correcto.
- Una vez leída una altura correcta, se calculará el grupo de altura (A, B, C, o D) que se le debe asignar (con la función `calcular_grupo` del ejercicio 2) y se añadirán los datos a las listas correspondientes.
- Si el identificador introducido ya se encuentra en la lista de identificadores, se informará al usuario de que ya están los datos de este bebé, y las mostrará por pantalla (altura y grupo).

### Ejercicio 4 (1 punto)

Hacer una función llamada `histograma` que reciba la lista de las alturas de los bebés y haga el recuento de cuántos bebés tienen cada una de las alturas entre 31 y 60. Este recuento se devolverá en una lista que contendrá el número de bebés con altura 31 a la primera posición, el número de bebés con altura 32 en la segunda, y así sucesivamente.

La lista donde se guardará el recuento, se debe crear al principio de la función con ceros en todas las posiciones.

### Ejercicio 5 (1 punto)

Hacer una función llamada `estadistica` que permita obtener algunas medidas estadísticas sobre los datos guardados en una lista. La función recibirá una lista con valores numéricos, y calculará la media y la varianza de los datos.

Para calcular la media, suponga que disponemos de una FUNCIÓN ya implementadas:

`media(lista) →` calcula y devuelve la media de la lista pasada como parámetro

Recuerde que, para calcular la varianza de un conjunto de datos, debe seguir la fórmula:

$$varianza = \frac{1}{N-1} \sum_i (x_i - \mu)^2$$

donde  $x_i$  son los elementos del conjunto de datos,  $\mu$  es la media de los datos y  $N$  es el número de elementos del conjunto de datos.

La función devolverá los dos valores correspondientes a la media y la varianza.

### Ejercicio 6 (1 punto)

Hacer un procedimiento llamado `consultar_grupo` que permita realizar consultas de los datos de los bebés que pertenecen a un determinado grupo (A, B, C, D). El procedimiento recibirá como parámetros la lista con los identificadores de los bebés, la lista con las alturas de los bebés y la lista con el grupo al que pertenecen los bebés.

El procedimiento preguntará qué grupo (A, B, C o D) se quiere consultar y a continuación mostrará los datos (identificador y altura) de todos los bebés que pertenezcan al grupo indicado. Si se introduce un grupo inexistente, se mostrará el mensaje de error: "Error: Grupo inexistente" y se sale del procedimiento (NO se vuelve a preguntar el grupo).

### Ejercicio 7 (2 puntos)

Hacer un programa que siga los siguientes pasos:

1. inicializaciones de variables y constantes
2. Utilizar (NO IMPLEMENTAR) el procedimiento `menu_principai()` que permite imprimir por pantalla el siguiente menú:

```
----- MENU -----
1.- Introducir datos
2.- Histograma de alturas
3.- Estadísticas
4.- Consultar por grupo
5.- Finalizar
```

3. Pedir al usuario que introduzca una de las opciones.
4. Si la opción es 1, llamar la función del ej. 3, `gestionar_datos`, para introducir los datos de un nuevo bebé.
5. Si la opción es 2:

5.1. llamar la función del ej. 4, `histograma`, para generar el histograma de las alturas de los bebés.

5.2. mostrar cuántos bebés tienen cada una de las alturas con mensajes con el siguiente formato:

```
31 centímetros - X bebés
32 centímetros - Y bebés
...
60 centímetros - Z bebés
```

donde X es el número de bebés con 31cm, Y el número de bebés con 32cm, etc.

6. Si la opción es 3:
  - 6.1. comprobar si ya se han introducido los datos de algún bebé. En caso afirmativo:
    - 6.1.1. llamar la función del ej. 5, `estadistica`, para calcular las medidas estadísticas sobre las alturas de los bebés.
    - 6.1.2. mostrar un mensaje con la media y la varianza de las alturas de los bebés.
  - 6.2. si todavía no se han introducido datos, mostrar un mensaje de error.
7. Si la opción es 4, llamar al procedimiento del ej. 6, `consultar_grupo`, para hacer la consulta de los bebés que pertenecen a un determinado grupo.
8. Si la opción es 5, mostrar el mensaje: `Saliendo del programa ...`
9. Cualquier otra opción, escribir el mensaje: `Opción no permitida.`
10. Repetir los pasos 2 a 8, hasta que la opción del menú escogida sea la 5.

**Ejercicio 8 (1 punto)**

Hacer un programa que pida una cadena y luego muestre por pantalla cada una de las palabras que la forman en una línea diferente. Considera que todas las palabras de la cadena están separadas entre ellas por un carácter espacio ". No hay que tener en cuenta los signos de puntuación.

Por ejemplo, si la cadena que se entra es "Hola, como estás?", por pantalla debería mostrar:

Hola,  
como  
estás?

**Ejercicio 9 (1 punto)**

¿Cuáles serán los valores de x, y y z al finalizar el programa?

```
def procesar (x, w, lista):  
    for i in range (w):  
        lista.append (x)  
        x += w  
    return x  
  
def f (x, y, z):  
    lista = []  
    z = procesar (x, y, lista)  
    x = len (lista)  
    y += 1  
    return (z, x, y)  
  
x, y, z = f (2,3,2)
```

Valor x	Valor y	Valor z

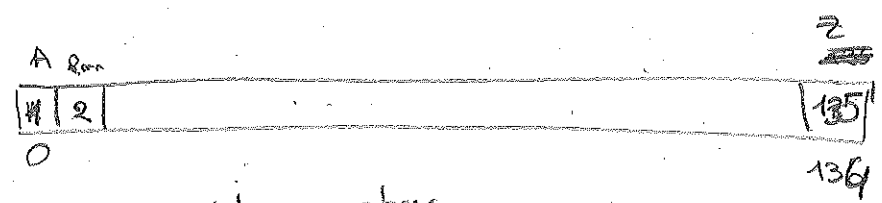


Periodistes del Parlament

Fuella → millor { intervenció d'un parlamentari.  
Pelliscada → pitjor

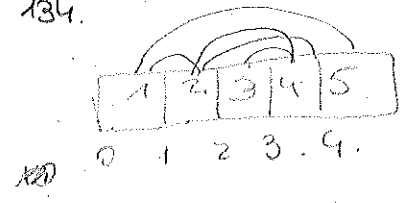
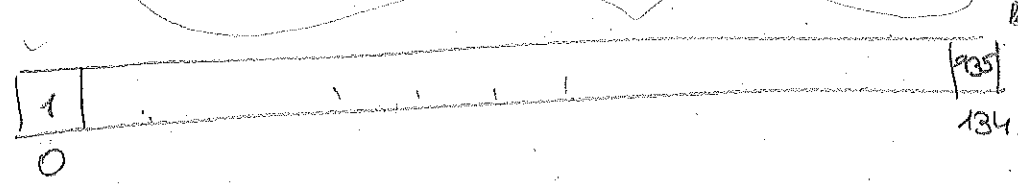
Diputats → 135

Ordenats alfabèticament → assignant valors de l'1 - 135.

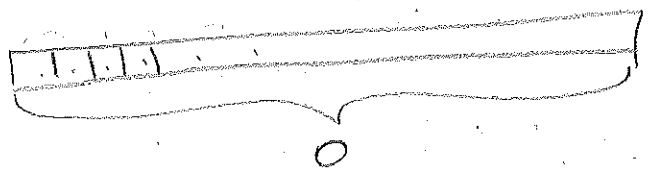


Todos deben votar

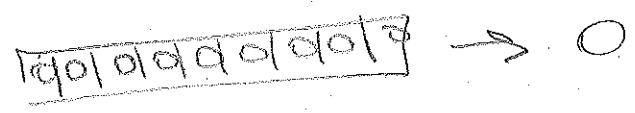
Alfabèticament ordenats → Alfabet 28.



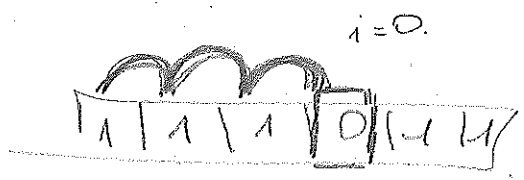
<  
→ valor →



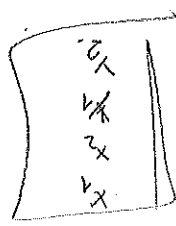
for (i=0; i < dimV; i++)



11111 → 1.



1	2	3	4	5
0	1	2	3	4



- $(x_1, y_1)$
- $(x_2, y_2)$

$5 \rightarrow 50\% \cdot P$   
 $3 \rightarrow 5 [25\% \rightarrow 50\%] P \rightarrow \text{valor} + (Prest / 2) \cdot 2 \times Prest$   
 $1 \rightarrow 5 [10\% \rightarrow 25\%] P$

if ( sedan >= (Prest x 50 / 100) ) || ( valor x 2 ) > Prest ) {

valor = P1 + P2  
 cout << valor

else if ( ( sedan >= (Prest x 25 / 100) ) || ( valor / 2 ) > Prest ) {

valor = P1 + P2  
 cout << valor

else if ( ( sedan >= (Prest x 10 / 100) )

valor = 1;

else return 0;

Menu →

int opción;

int votos [10];

int voto;

if (voto &lt; min) || (voto &gt; max)

{ cout &lt;&lt; "

amen&gt; voto;

{ do (~~while (voto < min) || (voto > max)~~)

do

while

int

salario;

cin >> ~~salario~~

if (salario &lt; 0)

if (salario == 100%)

~~100~~  
~~10~~

int voto;

opcion 2 == do

{ cout &lt;&lt; "Introduce el voto" &lt;&lt; endl;

cin &gt;&gt; voto;

{ while ((voto &gt; min) || (voto &lt; max))

votar (int min, int max);

for (int i = 0; i &lt; min; i++)

{

votos [i];

}

opcion 2 = int votos [10];

No cero Array (min, max);

if (votos &lt; N-votos)

{ Inicializar Array (min, max);

{ cout &lt;&lt; votos [i];

}

{ &lt;&lt; cout

}

opcion 3 =

do

{ indiceMinimo No Cero Array (Array [i], min);

int indiceMaximo Array (i);

cout &lt;&lt; mensaje

while (escrutinio == True)

if (escrutinio == false)

{ cout &lt;&lt; mensaje }

{

3 wh

funcion → int  
 Precedimiento → void  
 int vector (int mny, int max)

do  
 cout << "Entrar un valor entre < min < " << " y " << max << endl;  
 cin >> num;

while (num < min) // (min < max)

for (i=0; i < num; i++)

{ cout

void Ini

int NoCeroArray (Array [i], DIM)  
 return 1 → if [valores array] > 0  
 return 0 → else

{ int i;  
 for (i=0; i < DIM; i++)  
 Bool taboat;  
 taboat = true;

Array [i]  
 if (Array [i] > 0)

{ return 1;

else

{ return 0;

void Escrutinio (T votos [1, DIM, T resultados [1])  
 { int i, int votos, int resultados;  
 for (i=0; i < DIM; i++)

{  
 {  
 { int i;  
 int valor min;  
 int pos min;  
 for (i=0; i < DIM; i++)

# Fonaments de Programació (104337)

Curso 2019 - 2020 - Examen Parcial (29 de octubre de 2019)

Nombre estudiante: Raul Wilson MIU: 8645833

Important: Recuerda que hay que dar las mejores soluciones posibles en cada ejercicio. Además de funcionar correctamente, los procedimientos y funciones deben estar bien programados (utilizando las instrucciones más adecuadas, sin operaciones ni variables innecesarias, etc.)

## Los ejercicios 1 a 7 se refieren a un mismo tema que introducimos a continuación:

Un hospital materno-infantil nos pide hacer un programa para gestionar algunos datos médicos de los bebés nacidos en el hospital. Más concretamente, se quiere guardar y gestionar la información referente a la altura de los bebés al nacer.

El rango de alturas a considerar es el que va desde 31 cm hasta 60 cm. En caso de que el bebé mida menos de 31 cm o más de 60 cm la altura introducirá como 31 o 60 respectivamente.

Las anotaciones siempre se hacen en centímetros (cm) y se guardarán como valores enteros.

Cada bebé está identificado con un código numérico de 5 cifras. La información que se quiere guardar de cada bebé es su código y su altura.

Además, cada bebé se le asigna un grupo (A, B, C, D) en función de la altura que tiene. Este grupo, también se ha de guardar para cada bebé.

Todos estos datos se guardarán en tres listas (una para los identificadores de los bebés, otra por las alturas y otra por el grupo al que pertenece cada bebé). El orden en el que se guardan los datos es el mismo en las tres listas, es decir, el identificador de un bebé, su altura y el grupo al que pertenece, están en el mismo índice de la lista correspondiente.

### Ejercicio 1 (1 punto)

Hacer una función llamada `buscar` que reciba como parámetro una lista de enteros y un elemento a buscar, y compruebe si el elemento se encuentra en la lista. Si el elemento está en la lista, la función devolverá el índice de la posición donde se ha encontrado el elemento. Si no está, devolverá -1.

Important: NO SE PUEDEN UTILIZAR FUNCIONES DE MANIPULACIÓN DE LISTAS, SALVO LA FUNCIÓN `len()`

$$\begin{aligned} 15 \times 20 &= 300 \\ 150 + 10 &= 160 \end{aligned}$$

### Ejercicio 2 (1 punto)

Hacer una función llamada `calcular_grupo` que calcule el grupo (A, B, C, D) que se le debe asignar a un bebé en función de su altura. La función recibirá la altura en centímetros de un bebé como parámetro y devolverá el carácter correspondiente al grupo que corresponde a aquella altura según el siguiente criterio:

- 'A' si la altura del bebé es inferior a 35cm
- 'B' si la altura del bebé se encuentra en el intervalo [35cm, 40cm)
- 'C' si la altura del bebé se encuentra en el intervalo [40cm, 50cm)
- 'D' si la altura del bebé es igual o superior a 50cm

Nota: Considere que la altura nunca será negativa (no hay que hacer la comprobación).

$$\begin{aligned} 7 \times 3 &= 21 \\ 10 \times 3 &= 30 \\ \text{Resultat} &: 30 \end{aligned}$$

**Ejercicio 3 (1 punto)**

Hacer un procedimiento llamado `gestionar_datos` que permita introducir los datos de un bebé nuevo. El procedimiento recibirá como parámetros las listas con los identificadores de los bebés, la lista de las alturas de los bebés y la lista del grupo al que pertenece cada bebé.

El procedimiento pedirá el identificador numérico de un bebé y comprobará si se encuentra en la lista de identificadores (con la función `buscar_elemento` del ejercicio 1).

- Si no se encuentra el identificador en la lista, se añadirá el identificador a la lista de identificadores y se pedirá la altura del bebé. Si el valor introducido es menor o igual que cero, se mostrará un mensaje de error ("ERROR: Valor incorrecto por la altura") y se volverá a pedir de introducir la altura del bebé. Este proceso se repetirá hasta que se introduzca un valor correcto.
- Una vez leída una altura correcta, se calculará el grupo de altura (A, B, C, o D) que se le debe asignar (con la función `calcular_grupo` del ejercicio 2) y se añadirán los datos a las listas correspondientes.
- Si el identificador introducido ya se encuentra en la lista de identificadores, se informará al usuario de que ya están los datos de este bebé, y las mostrará por pantalla (altura y grupo).

**Ejercicio 4 (1 punto)**

Hacer una función llamada `histograma` que reciba la lista de las alturas de los bebés y haga el recuento de cuántos bebés tienen cada una de las alturas entre 31 y 60. Este recuento se devolverá en una lista que contendrá el número de bebés con altura 31 a la primera posición, el número de bebés con altura 32 en la segunda, y así sucesivamente.

La lista donde se guardará el recuento, se debe crear al principio de la función con ceros en todas las posiciones.

**Ejercicio 5 (1 punto)**

Hacer una función llamada `estadistica` que permita obtener algunas medidas estadísticas sobre los datos guardados en una lista. La función recibirá una lista con valores numéricos, y calculará la media y la varianza de los datos.

Para calcular la media, suponga que disponemos de una FUNCIÓN ya implementadas:

`media(lista)` → calcula y devuelve la media de la lista pasada como parámetro

Recuerde que, para calcular la varianza de un conjunto de datos, debe seguir la fórmula:

$varianza = \frac{1}{N-1} \sum_i (x_i - \mu)^2$	donde $x_i$ son los elementos del conjunto de datos, $\mu$ es la media de los datos y $N$ es el número de elementos del conjunto de datos.
---	--

La función devolverá los dos valores correspondientes a la media y la varianza.

**Ejercicio 6 (1 punto)**

Hacer un procedimiento llamado `consultar_grupo` que permita realizar consultas de los datos de los bebés que pertenecen a un determinado grupo (A, B, C, D). El procedimiento recibirá como parámetros la lista con los identificadores de los bebés, la lista con las alturas de los bebés y la lista con el grupo al que pertenecen los bebés.

El procedimiento preguntará qué grupo (A, B, C o D) se quiere consultar ya continuación mostrará los datos (identificador y altura) de todos los bebés que pertenezcan al grupo indicado. Si se introduce un grupo inexistente, se mostrará el mensaje de error: "Error: Grupo inexistente" y se sale del procedimiento (NO se vuelve a preguntar el grupo).

### Ejercicio 7 (2 puntos)

Hacer un programa que siga los siguientes pasos:

1. inicializaciones de variables y constantes
2. Utilizar (NO IMPLEMENTAR) el procedimiento `menu_principal()` que permite imprimir por pantalla el siguiente menú:

```
----- MENU -----
1.- Introducir datos
2.- Histograma de alturas
3.- Estadísticas
4.- Consultar por grupo
5.- Finalizar
```

3. Pedir al usuario que introduzca una de las opciones.
4. Si la opción es 1, llamar la función del ej. 3, `gestionar_datos`, para introducir los datos de un nuevo bebé.

5. Si la opción es 2:

5.1. llamar la función del ej. 4, `histograma`, para generar el histograma de las alturas de los bebés.

5.2. mostrar cuántos bebés tienen cada una de las alturas con mensajes con el siguiente formato:

```
31 centímetros - X bebés
32 centímetros - Y bebés
...
60 centímetros - Z bebés
```

donde X es el número de bebés con 31cm, Y el número de bebés con 32cm, etc.

6. Si la opción es 3:

6.1. comprobar si ya se han introducido los datos de algún bebé. En caso afirmativo:

6.1.1. llamar la función del ej. 5, `estadistica`, para calcular las medidas estadísticas sobre las alturas de los bebés.

6.1.2. mostrar un mensaje con la media y la varianza de las alturas de los bebés.

6.2. si todavía no se han introducido datos, mostrar un mensaje de error.

7. Si la opción es 4, llamar al procedimiento del ej. 6, `consultar_grupo`, para hacer la consulta de los bebés que pertenecen a un determinado grupo.

8. Si la opción es 5, mostrar el mensaje: `Saliendo del programa ...`

9. Cualquier otra opción, escribir el mensaje: `Opción no permitida.`

10. Repetir los pasos 2 a 9 hasta que la opción del menú escogida sea la 5.

**Ejercicio 8 (1 punto)**

Hacer un programa que pida una cadena y luego muestre por pantalla cada una de las palabras que la forman en una línea diferente. Considera que todas las palabras de la cadena están separadas entre ellas por un carácter espacio ". No hay que tener en cuenta los signos de puntuación.

Por ejemplo, si la cadena que se entra es "Hola, como estás?", por pantalla debería mostrar:

Hola,  
como  
estás?

**Ejercicio 9 (1 punto)**

¿Cuáles serán los valores de x, y y z al finalizar el programa?

```
def procesar (x, w, lista):  
    for i in range (w):  
        lista.append (x)  
        x += w  
    return x  
  
def f (x, y, z):  
    lista = []  
    z = procesar (x, y, lista)  
    x = len (lista)  
    y += 1  
    return (z, x, y)  
  
x, y, z = f (2,3,2)
```

Valor x	Valor y	Valor z
---------	---------	---------



Ex. 2

```
int Votar (int min, int max)
{
    int vot;
    cout << "entre valor entre " << min << " " << max << " : ";
    cin >> vot;

    while ((vot < min) || (vot > max))
    {
        cout << "Error ";
        cout << "entre valor entre ";
        cin >> vot;
    }

    return vot;
}
```

Ex. 5

```
void Escrutini (int Vots [], int dm, int Resultats)
{
    for (int i = 0; i < dm; i++)
    {
        Resultats[Vots[i]]++;
    }
}
```

Ex. 4

```
int NoZeroArray (int a [], int dm)
{
    bool trobat = false;
    int i = 0;
    while ((i < dm) && (!trobat))
    {
        if (a[i] == 0)
            trobat = true;
        else
            i++;
    }

    return !trobat;
}
```

Ex. 6

```
int IndexMinimNoZeroArray (int a [], int dm)
{
    int valmin = a[0];
    int posmin = 0;
    int i = 0;
    while for (i = 0; i = 1; i < dm; i++)
    {
        if ((a[i] < valmin) && (a[i] != 0))
        {
            valmin = a[i];
            posmin = i;
        }
    }

    return posmin;
}
```



# Fonaments de Programació (104337)

Curso 2019 - 2020 - Examen Parcial (29 de octubre de 2019)

Nombre estudiante: JANE GREENE

NIU: 3558008

Important: Recuerda que hay que dar las mejores soluciones posibles en cada ejercicio. Además de funcionar correctamente, los procedimientos y funciones deben estar bien programados (utilizando las instrucciones más adecuadas, sin operaciones ni variables innecesarias, etc.)

## Los ejercicios 1 a 7 se refieren a un mismo tema que introducimos a continuación:

Un hospital materno-infantil nos pide hacer un programa para gestionar algunos datos médicos de los bebés nacidos en el hospital. Más concretamente, se quiere guardar y gestionar la información referente a la altura de los bebés al nacer.

El rango de alturas a considerar es el que va desde 31 cm hasta 60 cm. En caso de que el bebé mida menos de 31 cm o más de 60 cm la altura introducirá como 31 o 60 respectivamente.

Las anotaciones siempre se hacen en centímetros (cm) y se guardarán como valores enteros.

Cada bebé está identificado con un código numérico de 5 cifras. La información que se quiere guardar de cada bebé es su código y su altura.

Además, cada bebé se le asigna un grupo (A, B, C, D) en función de la altura que tiene. Este grupo, también se ha de guardar para cada bebé.

Todos estos datos se guardarán en tres listas (una para los identificadores de los bebés, otra por las alturas y otra por el grupo al que pertenece cada bebé). El orden en el que se guardan los datos es el mismo en las tres listas, es decir, el identificador de un bebé, su altura y el grupo al que pertenece, están en el mismo índice de la lista correspondiente.

### Ejercicio 1 (1 punto)

Hacer una función llamada `buscar_elemento` que reciba como parámetro una lista de enteros y un elemento a buscar, y compruebe si el elemento se encuentra en la lista. Si el elemento está en la lista, la función devolverá el índice de la posición donde se ha encontrado el elemento. Si no está, devolverá -1.

Importante: NO SE PUEDEN UTILIZAR FUNCIONES DE MANIPULACIÓN DE LISTAS, SALVO LA FUNCIÓN `len()`

$$9 \times 9 = 81$$

$$81 + 9 = 90$$

### Ejercicio 2 (1 punto)

Hacer una función llamada `calcular_grupo` que calcule el grupo (A, B, C, D) que se le debe asignar a un bebé en función de su altura. La función recibirá la altura en centímetros de un bebé como parámetro y devolverá el carácter correspondiente al grupo que corresponde a aquella altura según el siguiente criterio:

- 'A' si la altura del bebé es inferior a 35cm
- 'B' si la altura del bebé se encuentra en el intervalo [35cm, 40cm)
- 'C' si la altura del bebé se encuentra en el intervalo [40cm, 50cm)
- 'D' si la altura del bebé es igual o superior a 50cm

Nota: Considere que la altura nunca será negativa (no hay que hacer la comprobación).

$$3 \times 2 = 6$$

$$6 \times 6 = 36$$

$$\text{Resultat} = 36$$

### Ejercicio 3 (1 punto)

Hacer un procedimiento llamado `gestionar_datos` que permita introducir los datos de un bebé nuevo. El procedimiento recibirá como parámetros las listas con los identificadores de los bebés, la lista de las alturas de los bebés y la lista del grupo al que pertenece cada bebé.

El procedimiento pedirá el identificador numérico de un bebé y comprobará si se encuentra en la lista de identificadores (con la función `buscar_elemento` del ejercicio 1).

- Si no se encuentra el identificador en la lista, se añadirá el identificador a la lista de identificadores y se pedirá la altura del bebé. Si el valor introducido es menor o igual que cero, se mostrará un mensaje de error ("ERROR: Valor incorrecto por la altura") y se volverá a pedir de introducir la altura del bebé. Este proceso se repetirá hasta que se introduzca un valor correcto.
- Una vez leída una altura correcta, se calculará el grupo de altura (A, B, C, o D) que se le debe asignar (con la función `calcular_grupo` del ejercicio 2) y se añadirán los datos a las listas correspondientes.
- Si el identificador introducido ya se encuentra en la lista de identificadores, se informará al usuario de que ya están los datos de este bebé, y las mostrará por pantalla (altura y grupo).

### Ejercicio 4 (1 punto)

Hacer una función llamada `histograma` que reciba la lista de las alturas de los bebés y haga el recuento de cuántos bebés tienen cada una de las alturas entre 31 y 60. Este recuento se devolverá en una lista que contendrá el número de bebés con altura 31 a la primera posición, el número de bebés con altura 32 en la segunda, y así sucesivamente.

La lista donde se guardará el recuento, se debe crear al principio de la función con ceros en todas las posiciones.

### Ejercicio 5 (1 punto)

Hacer una función llamada `estadistica` que permita obtener algunas medidas estadísticas sobre los datos guardados en una lista. La función recibirá una lista con valores numéricos, y calculará la media y la varianza de los datos.

Para calcular la media, suponga que disponemos de una FUNCIÓN ya implementadas:

`media(lista) →` calcula y devuelve la media de la lista pasada como parámetro

Recuerde que, para calcular la varianza de un conjunto de datos, debe seguir la fórmula:

$$varianza = \frac{1}{N-1} \sum_i (x_i - \mu)^2$$

donde  $x_i$  son los elementos del conjunto de datos,  $\mu$  es la media de los datos y  $N$  es el número de elementos del conjunto de datos.

La función devolverá los dos valores correspondientes a la media y la varianza.

### Ejercicio 6 (1 punto)

Hacer un procedimiento llamado `consultar_grupo` que permita realizar consultas de los datos de los bebés que pertenecen a un determinado grupo (A, B, C, D). El procedimiento recibirá como parámetros la lista con los identificadores de los bebés, la lista con las alturas de los bebés y la lista con el grupo al que pertenecen los bebés.

El procedimiento preguntará qué grupo (A, B, C o D) se quiere consultar ya continuación mostrará los datos (identificador y altura) de todos los bebés que pertenezcan al grupo indicado. Si se introduce un grupo inexistente, se mostrará el mensaje de error: "Error: Grupo inexistente" y se sale del procedimiento (NO se vuelve a preguntar el grupo).

### Ejercicio 7 (2 puntos)

Hacer un programa que siga los siguientes pasos:

1. inicializaciones de variables y constantes
2. Utilizar (NO IMPLEMENTAR) el procedimiento `menu_principal()` que permite imprimir por pantalla el siguiente menú:

```
----- MENU -----
1.- Introducir datos
2.- Histograma de alturas
3.- Estadísticas
4.- Consultar por grupo
5.- Finalizar
```

3. Pedir al usuario que introduzca una de las opciones.
4. Si la opción es 1, llamar la función del ej. 3, `gestionar_datos`, para introducir los datos de un nuevo bebé.
5. Si la opción es 2:

5.1. llamar la función del ej. 4, `histograma`, para generar el histograma de las alturas de los bebés.

5.2. mostrar cuántos bebés tienen cada una de las alturas con mensajes con el siguiente formato:

```
31 centímetros - X bebés
32 centímetros - Y bebés
...
60 centímetros - Z bebés
```

donde X es el número de bebés con 31cm, Y el número de bebés con 32cm, etc.

6. Si la opción es 3:
  - 6.1. comprobar si ya se han introducido los datos de algún bebé. En caso afirmativo:
    - 6.1.1. llamar la función del ej. 5, `estadistica`, para calcular las medidas estadísticas sobre las alturas de los bebés.
    - 6.1.2. mostrar un mensaje con la media y la varianza de las alturas de los bebés.
  - 6.2. si todavía no se han introducido datos, mostrar un mensaje de error.
7. Si la opción es 4, llamar al procedimiento del ej. 6, `consultar_grupo`, para hacer la consulta de los bebés que pertenecen a un determinado grupo.
8. Si la opción es 5, mostrar el mensaje: `Saliendo del programa ...`
9. Cualquier otra opción, escribir el mensaje: `Opción no permitida.`
10. Repetir los pasos 2 a 8, hasta que la opción del menú escogida sea la 5.

**Ejercicio 8 (1 punto)**

Hacer un programa que pida una cadena y luego muestre por pantalla cada una de las palabras que la forman en una línea diferente. Considera que todas las palabras de la cadena están separadas entre ellas por un carácter espacio ". No hay que tener en cuenta los signos de puntuación.

Por ejemplo, si la cadena que se entra es "Hola, como estás?", por pantalla debería mostrar:

Hola,  
como  
estás?

**Ejercicio 9 (1 punto)**

¿Cuáles serán los valores de x, y y z al finalizar el programa?

```
def procesar (x, w, lista):  
    for i in range (w):  
        lista.append (x)  
        x + = w  
    return x  
  
def f (x, y, z):  
    lista = []  
    z = procesar (x, y, lista)  
    x = len (lista)  
    y + = 1  
    return (z, x, y)  
  
x, y, z = f (2,3,2)
```

	Valor x	Valor y	Valor z

```
int Valor (int min, int max)
```

```
int vot;
```

```
cout << " ";
```

```
cin >> vot;
```

```
While (vot < min || vot > max)
```

```
    cout << " ";
```

```
    cout << " ";
```

```
    cin >> vot;
```

```
return vot;
```

```
Void InicialitzarArray (int V[], int dim, int valor)
```

```
int i;
```

```
for (i=0; i < dim; i++)
```

```
    V[i] = valor;
```

```
int NoZeroArray (int V[], int dim)
```

```
int i=0;
```

```
bool trobat = false;
```

```
While ((i < dim) && (!trobat))
```

```
    if (V[i] == 0)
```

```
        trobat = true;
```

```
    else
```

```
        i++;
```

```
if (trobat == true)
```

```
    return 0;
```

```
else
```

```
    return 1;
```

void Esercizio1 (int Vets[], int dim, int Risultats[])

0	7
2	3

y = 7

res

j = 0

do {

for (i = 0; i < dim; i++)

if (Vets[i] == y)

res++;

Risultats[j] = res;

j++;

while (j < Risultats);

for (j = 0; j < dim; j++)

for (i = 0; i < dim; i++)

if (Vets[i] == y)

res++

int i, min, ind;

ind = 0;

min = Vets[0];

for (i = 1; i < dim; i++)

if (Vets[i] < min || Vets[i] == 0)

min = Vets[i];

ind = i;

return ind;