

Práctica 8 de programación en C

- Trabajo obligatorio realizado por dos alumnos.
- Sirve como base para la realización de la prueba en la décima sesión (acto 2 de evaluación).
- En esa sesión se pedirá añadir una nueva funcionalidad al programa realizado previamente por el grupo (a esa fecha debe funcionar correctamente).

Estudio del ranking de universidades

```
D:\Mega_Gmail\Docencia\Infr  X  +  v

----- Menu de opciones -----
1. Universidad mejor valorada.
2. Universidades que superan una valoracion minima.
3. Tabla de medias de universidades publicas y privadas.
4. Datos por continente.
0. Finalizar.
-----
Elige una opcion (0-4): |
```

Fichero que contiene la información: ***universidades.txt*** (bajarlo de Poliformat)

1. **Identificador o código:** cada universidad tiene un identificador único entre 1 y 1000.

2. **Tipo: 1: Pública, 2: Privada)**

3. **Continente:**

1. América.
2. Europa.
3. Asia.
4. África.
5. Oceanía.

Ejemplo de una línea:

472 2 5 53 21999 6159 2328 139 24.30 40.20 41.40 7.70 64.50

La universidad de identificador 472 es privada (2) y está situada en Oceanía (5). Tiene 53 escuelas. Tiene 21999 alumnos nacionales y 6159 internacionales. Tiene 2328 profesores nacionales y 139 internacionales. Sus índices son: académico 24.30, de empresas 40.20, de alumnos 41.40, de sostenibilidad 7.70 e investigador 64.50

4. **Número de escuelas (o facultades) de la universidad:** valor entero.

5. **Número de alumnos nacionales de la universidad:** valor entero.

6. **Número de alumnos internacionales de la universidad:** valor entero.

7. **Número de profesores nacionales de la universidad:** valor entero.

8. **Número de profesores internacionales de la universidad:** valor entero.

9. **Índice académico:** valor float entre 1 y 100. Este valor indica la percepción que sobre la universidad tienen los académicos de otras universidades. A mayor valor, mayor prestigio de la universidad.

10. **Índice empresas:** valor float entre 1 y 100. Este valor indica la percepción que sobre la formación de los alumnos tienen las empresas que han contratado alumnos de la universidad. A mayor valor, mayor satisfacción con la formación de los alumnos de la universidad.

11. **Índice alumnos:** valor float entre 1 y 100. Este valor indica la percepción que sobre su formación tienen los alumnos de la universidad. A mayor valor, mayor satisfacción.

12. **Índice sostenibilidad:** valor float entre 1 y 100. Este valor aúna diversos factores que indican si la universidad es más o menos sostenible. A mayor valor, mayor sostenibilidad.

13. **Índice investigador:** valor float entre 1 y 100. Este valor aúna diversos factores que indican si la universidad es puntera en investigación. A mayor valor, mayor impacto de la investigación realizada por su personal investigador.

Pasos a seguir:

- Implementa las funciones **menu** y **leer_datos**
- Ejecuta el código para comprobar que no hay errores de compilación
- Implementa la función **calcular_valoración** y comprueba que no se producen errores de compilación
- Desarrolla cada una de las opciones, declarando las funciones necesarias e implementando el código correspondiente del switch
- Prueba el correcto funcionamiento cuando hayas implementado todo lo relacionado con cada una de las opciones

```
#include <stdio.h>
#include <stdlib.h>
#define N_UNIVERSIDADES 1000
#define PUBLICA 1
#define PRIVADA 2
#define AMERICA 1
#define EUROPA 2
#define ASIA 3
#define AFRICA 4
#define OCEANIA 5

int menu() {
    int opcion;
    /*implementar menu*/
    return opcion;
}

// Función que Lee los datos: rellena todos los vectores para ser usados posteriormente
int lee_datos(int vTipo[], int vContinente[], int vEscuelas[], int vAlumnosNacionales[], int vAlumnosInternacionales[], int vProfesoresNacionales[], int vProfesoresInternacionales[], float vIndiceAcademico[], float vIndiceEmpresas[], float vIndiceAlumnos[], float vIndiceSostenibilidad[], float vIndiceInvestigador[]) {
    /*Poner código*/
    return 0;
}

void calcular_valoracion(int vAlumnosNacionales[], int vAlumnosInternacionales[], int vProfesoresNacionales[], int vProfesoresInternacionales[], float vIndiceAcademico[], float vIndiceEmpresas[], float vIndiceAlumnos[], float vIndiceSostenibilidad[], float vIndiceInvestigador[], float vValoracion[]) {
    /*Poner código*/
}

/* Poner aquí función universidad_mejor_valorada (opción 1)*/
/* Poner aquí función universidades_minimo (opción 2)*/
/* Poner aquí función medias_tipo (opción 3)*/
void medias_tipo(int vTipo[], int vEscuelas[], int vAlumnosNacionales[], int vAlumnosInternacionales[], int vProfesoresNacionales[], int vProfesoresInternacionales[], float *ratio_publicas, float *ratio_privadas) {
    /*Código de la función medias tipo*/
}

/* Poner aquí función universidades_continente (opción 4)*/
```

```
int main() {
    /*Declaración de vectores*/
    int vTipo[N_UNIVERSIDADES];
    int vContinente[N_UNIVERSIDADES];
    int vEscuelas[N_UNIVERSIDADES];
    int vAlumnosNacionales[N_UNIVERSIDADES];
    int vAlumnosInternacionales[N_UNIVERSIDADES];
    int vProfesoresNacionales[N_UNIVERSIDADES];
    int vProfesoresInternacionales[N_UNIVERSIDADES];
    float vIndiceAcademico[N_UNIVERSIDADES];
    float vIndiceEmpresas[N_UNIVERSIDADES];
    float vIndiceAlumnos[N_UNIVERSIDADES];
    float vIndiceSostenibilidad[N_UNIVERSIDADES];
    float vIndiceInvestigador[N_UNIVERSIDADES];
    float vValoracion[N_UNIVERSIDADES];
    int opcion;
    /*poner otras variables que necesiteis

    /*Invocar a la función Lee_datos*/
    if (res == -1) {
        printf("No se ha podido abrir el fichero.\n");
        return 1;
    }
    /*Invocar a la función calcular_valoracion:*/
    calcular_valoracion(vAlumnosNacionales, vAlumnosInternacionales, vProfesoresNacionales, vProfesoresInternacionales,
        vIndiceAcademico, vIndiceEmpresas, vIndiceAlumnos,
        vIndiceSostenibilidad, vIndiceInvestigador, vValoracion);

    do {
        opcion = menu(); /*Invoca a la opción menú*/
        switch (opcion) {
            case 1: {
                /*Código para la primera opción del menú: invoca a la función calcular_valoracion y realiza operaciones adicionales*/
                break;
            }
            case 2: {
                /*Código para la segunda opción del menú: invoca a la función calcular_valoracion y realiza operaciones adicionales*/
                break;
            }
            case 3: {
                /*Código para la tercera opción del menú: invocar a la función medias_tipo y realiza operaciones adicionales*/
                break;
            }
            case 4: {
                /*Código para la cuarta opción del menú: invocar a la función universidades_continente y realiza operaciones adicionales*/
                break;
            }
            case 5: {
                printf("Finalizando programa...\n");
                break;
            }
        }
    } while (opcion != 5);

    return 0;
}
```

```
// Función que lee los datos: rellena todos los vectores para ser usados posteriormente
```

```
int lee_datos(int vTipo[], int vContinente[], int vEscuelas[], int vAlumnosNacionales[], int vAlumnosInternacionales[], int vProfesoresNacionales[], int vProfesoresInternacionales[], float vIndiceAcademico[], float vIndiceEmpresas[], float vIndiceAlumnos[], float vIndiceSostenibilidad[])
/*Poner código*/
return 0;
}
```

Lee cada línea del fichero universidades.txt y almacena los datos en el correspondiente **vector**. Ten en cuenta que si lee el código **id** de la universidad, lo almacena en la posición **id-1** del vector

1. **Identificador o código:** cada universidad tiene un identificador único entre 1 y 1000.
2. **Tipo (1: Pública, 2: Privada)**
3. **Continente**
4. **Número de escuelas (o facultades) de la universidad:** valor entero.
5. **Número de alumnos nacionales de la universidad:** valor entero.
6. **Número de alumnos internacionales de la universidad:** valor entero.
7. **Número de profesores nacionales de la universidad:** valor entero.
8. **Número de profesores internacionales de la universidad:** valor entero.
9. **Índice académico:** valor float entre 1 y 100. Este valor indica la percepción que sobre la universidad tienen los académicos de otras universidades. A mayor valor, mayor prestigio de la universidad.
10. **Índice empresas:** valor float entre 1 y 100. Este valor indica la percepción que sobre la formación de los alumnos tienen las empresas que han contratado alumnos de la universidad. A mayor valor, mayor satisfacción con la formación de los alumnos de la universidad.
11. **Índice alumnos:** valor float entre 1 y 100. Este valor indica la percepción que sobre su formación tienen los alumnos de la universidad. A mayor valor, mayor satisfacción.
12. **Índice sostenibilidad:** valor float entre 1 y 100. Este valor aúna diversos factores que indican si la universidad es más o menos sostenible. A mayor valor, mayor sostenibilidad.
13. **Índice investigador:** valor float entre 1 y 100. Este valor aúna diversos factores que indican si la universidad es puntera en investigación. A mayor valor, mayor impacto de la investigación realizada por su personal investigador.

int vTipo[N_UNIVERSIDADES];	Vector que contiene en la posición n el tipo de la universidad $n+1$ (pública o privada).
int vContinente[N_UNIVERSIDADES];	Vector que contiene en la posición n el continente donde se sitúa la universidad $n+1$.
int vEscuelas[N_UNIVERSIDADES];	Vector que contiene en la posición n , el número de escuelas de la universidad $n+1$.
int vAlumnosNacionales[N_UNIVERSIDADES];	Vector que contiene en la posición n , el número de alumnos nacionales de la universidad $n+1$.
int vAlumnosInternacionales[N_UNIVERSIDADES];	Vector que contiene en la posición n , el número de alumnos internacionales de la universidad $n+1$.
int vProfesoresNacionales[N_UNIVERSIDADES];	Vector que contiene en la posición n , el número de profesores nacionales de la universidad $n+1$.
int vProfesoresInternacionales[N_UNIVERSIDADES];	Vector que contiene en la posición n , el número de profesores internacionales de la universidad $n+1$.
float vIndiceAcademico[N_UNIVERSIDADES];	Vector que contiene en la posición n , el índice académico de la universidad $n+1$.
float vIndiceEmpresas[N_UNIVERSIDADES];	Vector que contiene en la posición n , el índice de empresas de la universidad $n+1$.
float vIndiceAlumnos[N_UNIVERSIDADES];	Vector que contiene en la posición n , el índice de alumnos de la universidad $n+1$.
float vIndiceSostenibilidad[N_UNIVERSIDADES];	Vector que contiene en la posición n , el índice de sostenibilidad de la universidad $n+1$.
float vIndiceInvestigador[N_UNIVERSIDADES];	Vector que contiene en la posición n , el índice investigador de la universidad $n+1$.

Función *menu*

Muestra por pantalla el menú con las opciones y solicita al usuario una opción (número entero), controlando que el valor introducido sea válido, es decir, entre 0 y 4 (mediante un bucle do-while). Si el valor no es válido, volverá a solicitarlo al usuario tantas veces como sea necesario para garantizar que el valor sea correcto.

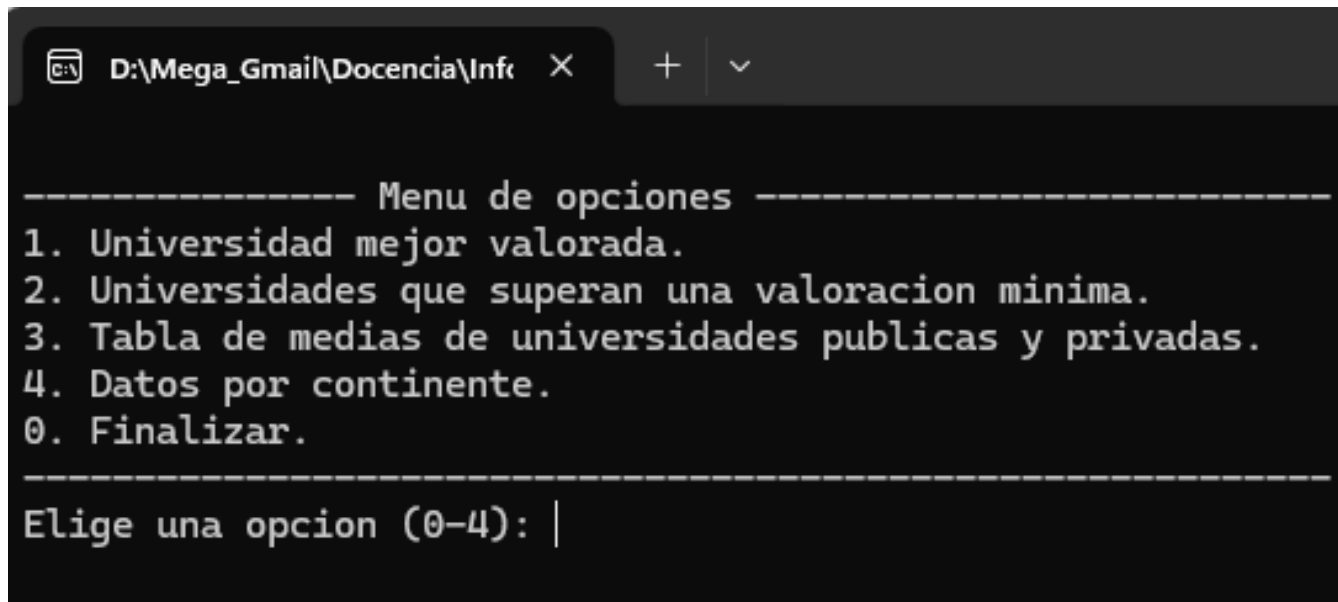
Finalmente, la función retorna a la función principal la opción seleccionada por el usuario.

- **Parámetros:** ninguno.
- **Valor de retorno:** valor entero (entre 0 y 4) que se corresponde con la opción elegida por el usuario.

```
int menu() {  
    int opcion;  
    /*implementar menu*/  
    return opcion;  
}
```

Función *principal*

La función principal llama a la función *menu* dentro del bucle do-while.



```
D:\Mega_Gmail\Docencia\Infr  x  +  v  
----- Menu de opciones -----  
1. Universidad mejor valorada.  
2. Universidades que superan una valoracion minima.  
3. Tabla de medias de universidades publicas y privadas.  
4. Datos por continente.  
0. Finalizar.  
-----  
Elige una opcion (0-4): |
```

Función: calcular_valoración

La valoración de una universidad se calcula como la media ponderada de los siguientes datos:

- Índice académico 20%.
- Índice de empresa 20%.
- Índice de estudiantes 20%.
- Índice investigador 20%.
- Índice de sostenibilidad 10%.
- Porcentaje de profesores internacionales 5%. El porcentaje se calcula con el número de profesores internacionales de esa universidad con respecto al número de profesores totales de esa universidad.
- Porcentaje de estudiantes internacionales 5%. El porcentaje se calcula con el número de estudiantes internacionales de esa universidad con respecto al número de estudiantes totales de esa universidad.

Función calcular_valoración

La función obtendrá la valoración de cada una de las universidades, siguiendo la fórmula anterior. Las valoraciones se almacenan en el vector **vValoracion**. La valoración de la universidad $n+1$ se almacenará en la posición n del vector **vValoracion**.

La función no mostrará nada por pantalla.

- **Parámetros:** vectores:
 - vAlumnos
 - vAlumnosInternacionales
 - vProfesores
 - vProfesoresInternacionales
 - vIndiceAcademico
 - vIndiceEmpresas
 - vIndiceAlumnos
 - vIndiceSostenibilidad
 - vIndiceInvestigador
 - vValoracion: este vector lo rellena la función con la valoración de cada universidad
- **Valor de retorno:** ninguno.

Función principal

La función principal no mostrará nada por pantalla como resultado de esta opción.

La opción se implementa mediante la función universidad mejor valorada. La función obtiene:

- Universidad mejor valorada (mayor valoración).
- Universidad más sostenible (mayor índice de sostenibilidad).
- Universidad más internacional (mayor valor de la suma de profesores internacionales y de alumnos internacionales).

Además, la opción mostrará por pantalla si hay (o no), alguna universidad que sea la mejor valorada en las tres categorías.

Función universidad mejor valorada

La función calculará cuál es la universidad mejor valorada, la más sostenible y la más internacional. Los valores obtenidos los escribirá la función por pantalla.

La salida por pantalla de esta función deberá tener el formato que se muestra posteriormente en el ejemplo de funcionamiento.

- **Parámetros:** vectores:
 - yAlumnos
 - yAlumnosInternacionales
 - yProfesores
 - yProfesoresInternacionales
 - yIndiceSostenibilidad
 - yValoracion
- **Valor de retorno:**
 - Si hay alguna universidad que sea la mejor en las tres categorías, retorna su identificador.
 - Si no hay ninguna universidad que sea la mejor en las tres categorías, retorna -1.

Función principal

La función principal llama a la función universidad mejor valorada, y dependiendo del valor de retorno de la función, muestra un mensaje u otro. La función principal muestra por pantalla:

- Si la función retorna -1: muestra el mensaje “No hay ninguna universidad que sea mejor en las 3 categorías”.
- Si la función retorna otro valor (n) significa que hay una universidad que es la mejor en las tres categorías. En ese caso, muestra el mensaje “La universidad n es la mejor valorada en las 3 categorías”.

```
----- Menu de opciones -----
. Universidad mejor valorada.
. Universidades que superan una valoración mínima.
. Tabla de medias de universidades públicas y privadas.
. Datos por continente.
. Finalizar.
-----
Lige una opción (0-4): 1

universidad mejor valorada: 161 (79.76 puntos)
universidad mas sostenible: 839 (99.90 puntos)
universidad mas internacional: 349 (247 profesores internacionales y 9516 alumnos internacionales)

c hay ninguna universidad que sea mejor en las 3 categorías
```

Opción 2: Universidades que superan una valoración mínima

La opción obtiene las universidades de un tipo (pública o privada), de un determinado continente y que superan un valor mínimo de valoración. Además, escribirá en sendos ficheros, las universidades que cumplen las condiciones, pero no superan el mínimo y en otro las que cumplen las condiciones, pero superan el mínimo. La función principal será el encargado de pedir al usuario los datos necesarios (tipo, continente y valoración mínima) y pasarlos como parámetro a la función *universidades_minimo*.

Función *universidades_minimo*

La función comienza abriendo dos ficheros: *valoracion_menor.txt* y *valoracion_mayor.txt*. Si alguno de estos ficheros no puede abrirse, la función retorna -1 a la función principal y termina.

La función recibe como parámetro, además de los vectores, tres variables: tipo de universidad, continente y valoración mínima.

La función escribe en el fichero *valoracion_menor.txt*, aquellas universidades que son del tipo pasado como parámetro, que están en el continente pasado como parámetro y cuya valoración es menor que la valoración mínima pasada como parámetro.

La función escribe en el fichero *valoracion_mayor.txt*, aquellas universidades que son del tipo pasado como parámetro, que están en el continente pasado como parámetro y cuya valoración es mayor o igual que la valoración mínima pasada como parámetro.

En ambos casos, se debe escribir, en el fichero correspondiente, el identificador de la universidad y su valoración. El formato de ambos ficheros se muestra posteriormente en el ejemplo.

Además, la función calculará y mostrará por pantalla:

- El porcentaje de universidades que cumplen las siguientes condiciones: son del tipo y continente pasado como parámetro y su valoración es mayor o igual a la valoración pasada como parámetro, respecto a las **universidades del mismo tipo y continente**.
- El porcentaje de universidades que cumplen las siguientes condiciones: son del tipo y continente pasado como parámetro y su valoración es mayor o igual a la valoración pasada como parámetro, respecto al **total de universidades** del fichero.

El formato del mensaje se muestra en el ejemplo de funcionamiento posterior.

• Parámetros:

- Vectores:
 - vTipo
 - vContinente
 - vValoracion
- Tipo de universidad (valor entero entre 1 y 2)
- Continente (valor entero entre 1 y 5)
- Valoración mínima (valor float entre 1 y 100)

• Valor de retorno:

- -1 si se produce un error al abrir algún fichero. La función no mostrará ningún mensaje si no se abre algún fichero. Será la función principal quien lo haga.
- 0 si se leen los datos correctamente.

Función principal

Cuando se selecciona la opción 2 del menú, la función principal, antes de llamar a la función *universidades_minimo*, solicita al usuario los siguientes datos:

- Tipo de universidad (1-pública, 2-privada): se debe comprobar que el valor leído es correcto mediante un bucle do-while.
- Continente (1-América, 2-Europa, 3-Asia, 4-Africa, 5-Oceania): se debe comprobar que el valor leído es correcto mediante un bucle do-while.
- Valoración mínima (valor float entre 1 y 100): se debe comprobar que el valor leído es correcto mediante un bucle do-while.

Después de leer los datos, se llamará a la función. Si la función retorna el valor -1, la **función principal** escribirá el mensaje de error "No es posible abrir el fichero."

Un ejemplo de ejecución puede ser el siguiente:

```
----- Menu de opciones -----
1. Universidad mejor valorada.
2. Universidades que superan una valoración mínima.
3. Tabla de medias de universidades publicas y privadas.
4. Datos por continente.
0. Finalizar.
-----
Elige una opción (0-4): 2

Escribe el tipo de universidad (1-pública, 2-privada): 1
Escribe un continente (1-América, 2-Europa, 3-Asia, 4-Africa, 5-Oceania): 4
Escribe la valoración mínima (1..100): 50

De las 92 universidades publicas de Africa, el 36.96% estan valoradas por encima de 50.00 puntos
Estas universidades son el 3.40% del total de universidades del mundo
```

Un ejemplo de las 10 primeras líneas de los ficheros para los datos de la ejecución anterior, sería:

```
Universidad 45: 62.44 puntos
Universidad 50: 60.61 puntos
Universidad 62: 54.62 puntos
Universidad 95: 60.08 puntos
Universidad 96: 56.74 puntos
Universidad 110: 62.44 puntos
Universidad 149: 56.53 puntos
Universidad 161: 79.76 puntos
Universidad 165: 51.61 puntos
```

valoración_mayor.txt

```
Universidad 3: 26.73 puntos
Universidad 9: 32.95 puntos
Universidad 43: 38.69 puntos
Universidad 56: 41.63 puntos
Universidad 64: 41.69 puntos
Universidad 98: 36.79 puntos
Universidad 109: 44.32 puntos
Universidad 124: 48.98 puntos
Universidad 127: 36.89 puntos
```

valoración_menor.txt

Opción 3: Tabla de medias de universidades públicas y privadas



La opción obtiene una tabla comparativa, entre las universidades privadas y públicas. Los datos que se muestran en la tabla son: número de alumnos, número de alumnos internacionales, número de profesores, número de profesores internacionales y número de escuelas. Estos datos se muestran tanto para las universidades públicas, como privadas. Además, obtiene el ratio de alumnos por profesor en ambos tipos de universidades.

La opción se implementa mediante la función *medias_tipo*.

Función *medias_tipo*

La función escribe por pantalla una tabla donde aparecen los siguientes datos para las universidades públicas y para las universidades privadas:

- Número medio de alumnos.
- Número medio de alumnos internacionales.
- Número medio de profesores.
- Número medio de profesores internacionales.
- Número medio de escuelas.

La tabla mostrará en cada fila cada uno de estos datos, tanto de las universidades públicas como de las privadas, además mostrará la palabra "Pública" si el valor de la universidad pública es mayor que el valor de la privada, y "Privada", en caso contrario. El formato de la tabla se muestra en el ejemplo de funcionamiento.

La función también obtendrá el número medio de alumnos por profesor, tanto para las universidades públicas como privadas. Se deben tener en cuenta todos los alumnos, tanto nacionales como internacionales, y todos los profesores, tanto nacionales como internacionales. Estos valores se pasan a la función principal como parámetros por referencia. La función principal será el encargado de escribirlos por pantalla.

• Parámetros: vectores:

- vTipo
- vEscuelas
- vAlumnos
- vAlumnosInternacionales
- vProfesores
- vProfesoresInternacionales

• Parámetros por referencia:

- Número medio de alumnos por profesor en las universidades públicas.
- Número medio de alumnos por profesor en las universidades privadas.

• Valor de retorno: ninguno.

Función principal

La función principal llama a la función *medias_tipo* pasándole como parámetro, además de los vectores necesarios, dos parámetros por referencia: ratio de alumnos por profesor en las universidades públicas y en las privadas.

La función principal escribirá por pantalla estos valores siguiendo el formato que se muestra en el ejemplo de ejecución.

```
----- Menu de opciones -----
1. Universidad mejor valorada.
2. Universidades que superan una valoracion minima.
3. Tabla de medias de universidades publicas y privadas.
4. Datos por continente.
0. Finalizar.
```

Elige una opcion (0-4): 3

Medias	Publica	Privada	Mayor
Alumnos	16939.83	17063.35	Privada
Alumnos internacionales	2682.70	2633.31	Publica
Profesores	1983.12	1944.41	Publica
Profesores internacionales	112.58	110.49	Publica
Escuelas	42.83	42.90	Privada

En las universidades publicas hay 9.4 alumnos por profesor
En las universidades privadas hay 9.6 alumnos por profesor

Declaración de la función:

```
void medias_tipo(int vTipo[], int vEscuelas[], int vAlumnosNacionales[], int vAlumnosInternacionales[],
                 int vProfesoresNacionales[], int vProfesoresInternacionales[], float *ratio_publicas, float *ratio_privadas) {
    /*Código de la función medias_tipo*/
    *ratio_publicas = .....
    *ratio_privadas = .....
}
```

Invocación desde el main:

```
medias_tipo(vTipo, vEscuelas, vAlumnosNacionales, vAlumnosInternacionales,
            vProfesoresNacionales, vProfesoresInternacionales,
            &ratio_publicas, &ratio_privadas);
```

Opción 4: Datos por continente

En esta opción se calculan una serie de datos sobre las universidades de los 5 continentes contenidas en el fichero.

La opción se implementa mediante la función *universidades_continente*.

Función *universidades_continente*

La función comienza abriendo el fichero *mejor_continente.txt*. Si el fichero no se abre, la función retornará -1 a la función principal y finalizará.

La función calculará y mostrará los siguientes valores:

- El porcentaje de universidades en cada continente, respecto al total de universidades.
- El porcentaje de alumnos (tanto nacionales como internacionales) de las universidades de cada continente, respecto al total de universidades.
- La valoración media de las universidades de cada continente.

La función mostrará estos valores por pantalla siguiendo el formato que se muestra en el ejemplo de funcionamiento.

Para el cálculo de dichos valores deben usarse los siguientes vectores (que se declaran dentro de la función):

- **float valoracion_contiente[5];** En este vector se irán sumando las valoraciones de las universidades de cada uno de los continentes (en la posición 0, las valoraciones de América,...).
- **int alumnos_contiente[5];** En este vector se irán sumando los alumnos de las universidades de cada continente. Se debe sumar los alumnos nacionales e internacionales.
- **int publicas_continente[5], privadas_continente[5];** En estos vectores se cuenta cuantas universidades públicas hay en cada continente, y cuantas privadas.

Además, la función calculará cual es el continente cuya valoración media es mayor.

Para ese continente, escribirá en el fichero *mejor_continente.txt*:

- En la primera línea, el nombre del continente.
- En las líneas siguientes, todas las universidades del continente. Para cada universidad escribirá su identificador y su valoración.

El formato del fichero se muestra en el ejemplo de funcionamiento.

• Parámetros de entrada: vectores:

- vTipo
- vContinente
- vAlumnos
- vAlumnosInternacionales
- vValoracion

• Valor de retorno:

- -1 si se produce un error al abrir algún fichero. La función no mostrará ningún mensaje si no se abre algún fichero. Será la función principal quien lo haga.
- 0 si se leen los datos correctamente.

Función principal

Cuando se seleccione la opción 4 del menú, la función principal llama a la función *universidades_continente*. Si la función *universidades_continente* retorna el valor -1, la función principal escribirá el mensaje de error "No es posible abrir el fichero."

Un ejemplo de ejecución puede ser el siguiente:

```
----- Menu de opciones -----
1. Universidad mejor valorada.
2. Universidades que superan una valoracion minima.
3. Tabla de medias de universidades publicas y privadas.
4. Datos por continente.
0. Finalizar.
-----
Elige una opcion (0-4): 4

Porcentaje de universidades en cada continente:
America: 21.60% universidades
Europa: 17.90% universidades
Asia: 22.90% universidades
Africa: 17.90% universidades
Oceania: 19.70% universidades

Porcentaje de alumnos en cada continente:
America: 21.71% alumnos
Europa: 18.61% alumnos
Asia: 22.26% alumnos
Africa: 16.98% alumnos
Oceania: 20.44% alumnos

Valoracion media de las universidades en cada continente:
America: 44.59 puntos
Europa: 46.21 puntos
Asia: 45.74 puntos
Africa: 45.38 puntos
Oceania: 45.52 puntos
```

Un ejemplo de las 10 primeras líneas del fichero sería:

```
Universidades de Europa:
Universidad 4: 51.49
Universidad 7: 62.15
Universidad 11: 36.38
```

```
Universidad 14: 66.59
Universidad 15: 62.49
Universidad 18: 34.44
Universidad 24: 39.31
Universidad 26: 57.97
```

Fichero *mejor_continente.txt*

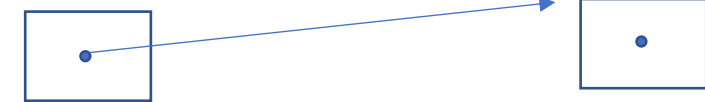
El paso por referencia se usa para devolver valores en los argumentos (parámetros) de entrada

- En la función se declara el argumento de entrada como un puntero. Para ello, se escribe el tipo de datos, el símbolo * y el nombre de la variable puntero (en el ejemplo, las variables área y perímetro)
- Dentro de la función se modifica adecuadamente la variable a la que apunta (en el ejemplo, *área y *perímetro)
- En la invocación a la función se escribe en el lugar correspondiente el símbolo & y el nombre de la variable

```
#include <stdio.h>
float cuadrado(float x){
    return x*x;
}
void area_perimetro(float b, float h, float *area, float *perimetro){
    *area=b*h;
    *perimetro=2*(b+h);
}
int main(){
    float x, y, base, altura, Area, Perimetro;
    printf("x? ");
    scanf("%f",&x);
    y=cuadrado(x);
    printf("x*x=%f\n",y);
    printf("introduce la base y la altura: ");
    scanf("%f%f",&base,&altura);
    area_perimetro(base,altura,&Area,&Perimetro);
    printf("Area=%f \t Perímetro=%f\n",Area,Perimetro);
}
```

area (puntero)

*area

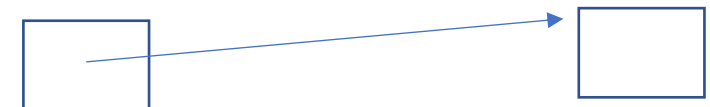


El contenido de la variable area (puntero) es la dirección de la variable *area

En la invocación se da la dirección de la variable Area, mediante el uso de & (igual que sucede con scanf)

&area

Area



Muchas gracias
y
¡ a programar !

