

Pràctica 7 de programació en C

Objectius

- Aplicar les estructures de selecció i repetició a la resolució de problemes, incloent bucles anitats.
- Conèixer la programació modular: funcions en C.
- Utilitzar vectors unidimensionals en la resolució de problemes.

Funcions en C

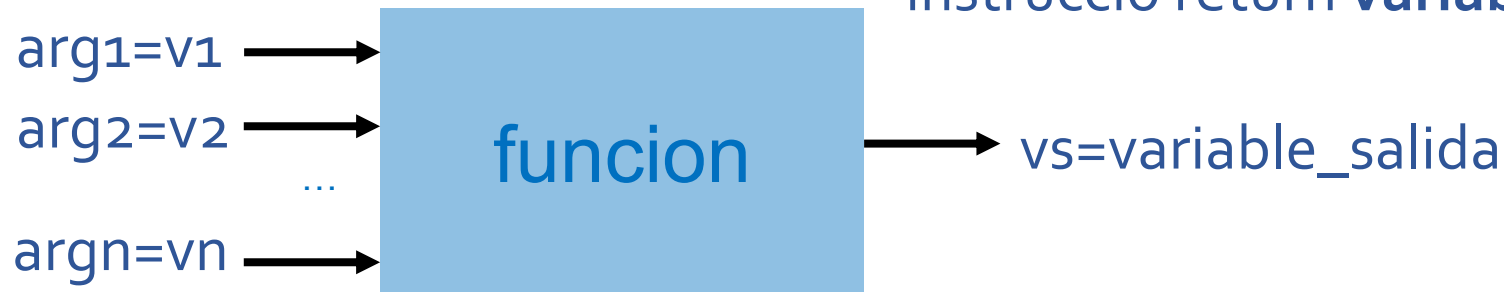
Implementació de la funció (es declara abans del programa principal (funció **main**)):

```
Tipo_variable_salida funcion(tipus1 arg1, tipus2 arg2,..., tipon argn){  
....  
return variable_salida  
}
```

- Invocació des d' una altra funció, en particular des del programa principal:

```
....  
vs =funcion(v1, v2, .....,vn);  
....
```

- **vs**=variable de sortida (emmagatzemarà el resultat)
- **vi** = variables o dades d' entrada
- Totes les variables han de ser del mateix tipus que les declarades en la funció
- Si en el **Tipo_variable_salida** es posa **void**, indica que no es retorna cap valor, i per tant, no apareix la instrucció return **variable_salida**



Exemple: càlcul de l' àrea d' un rectangle

```
#include <stdio.h>
float area_rectangulo(float base, float altura){
    float area;
    area=base*altura;
    return area;
}
int main(){
    float A, b, h;
    printf("Introduce la base del rectángulo: ");
    scanf("%f", &b);
    printf("Introduce la altura del rectángulo: ");
    scanf("%f", &h);
    A=area_rectangulo(b, h);
    printf("El área del rectángulo es igual a %f\n ",A);
    return 0;
}
```



Exercici 4

- Desenvolupar un programa en C que llegeixi d' un fitxer informació d' una flota d' avions i calculi algunes dades estadístiques dels avions.
- Es partirà d'un fitxer anomenat **vuelos.txt** (es troba a **Poliformat**) amb dades sobre els vols de diferents aeronaus durant un determinat període de temps. Cada aeronau ve identificada per un nombre sencer entre 0 i $N-1$, essent **N** el nombre total d' aeronaus de la flota.
- En el fitxer esmentat es guarden tres valors sencers per a cada vol realitzat: identificador de l'avió (que es repetirà per a tots els vols del mateix avió), distància del vol (en quilòmetres) i quantitat de combustible emprat (en litres). Per a cada avió hi ha tantes línies al fitxer (desordenades) com viatges ha realitzat. Els avions que no han realitzat cap trajecte no apareixen al fitxer.

Exemple:

41 6297 88159

62 850 14450

47 2965 47441

La primera línia informa d'un vol de l'avió amb 41 en el qual ha recorregut 6297 quilòmetres i per al qual ha necessitat 88159 litres de combustible. La segona línia indica que l'avió 62 ha fet un vol de 850 km gastant 14450 litres,

Cal completar aquest codi

Exercici 4



```
1 #include <stdio.h>
2 #include <float.h>
3
4 #define N 82 // Número total de aeronaves
5
6
7 int menu() {
8     int opc;
9
10    /* muestra un menú al usuario y solicita una opción. Si la opción no
11     está entre las del menú, indica un mensaje de error y vuelve a
12     mostrar menú / pedir opción, repitiéndose todo hasta que se
13     escoge una opción correcta */
14
15    return opc;
16 }
17
18 int cargarDatos(int num_vuelos[], int distancias[], int consumos[]) {
19     FILE *file;
20     int id, distancia, consumo;
21
22     file = fopen("vuelos.txt", "r");
23     if (file == NULL)
24         return 1;
25
26     while (0 /*sustituir el 0 por la condición que corresponda -lectura de una línea
27         num_vuelos[id]++;
28         distancias[id] += distancia;
29         consumos[id] += consumo;
30     })
31
32     fclose(file);
33     return 0;
34 }
35
36 void mostrarInformacionAvion(int num_vuelos[], int distancias[], int consumos[]) {
37     int id;
38
39     do {
40         printf("Avion (0-%d): ", N-1);
41         scanf("%d", &id);
42
43         if (id < 0 || id >= N)
44             printf("Error. Valor fuera de rango.\n");
45     } while (id < 0 || id >= N);
46
47     if (num_vuelos[id] == 0) {
48         printf("El avion %d no ha realizado vuelos.\n", id);
49     } else {
50         /* Falta el cálculo del consumo medio y mostrar por pantalla
51         el número total de vuelos, la distancia total recorrida, el
52         consumo total de combustible y el consumo medio por kilómetro. */
53     }
54 }
```

```
56 void mostrarEficienciaAviones(int num_vuelos[], int distancias[], int consumos[]) {
57     int i, id_mas_eficiente = -1, id_menos_eficiente = -1;
58     float consumo_medio_mas_eficiente = FLT_MAX, consumo_medio_menos_eficiente = 0, consumo_medio;
59
60     for (i = 0; i < N; i++) {
61         /* Falta completar el bloque del for: recorrido de los vectores
62         para encontrar el avión más eficiente y el menos eficiente en
63         términos de consumo medio por kilómetro */
64     }
65
66     if (id_menos_eficiente != -1) {
67         printf("Avion menos eficiente: %d (consumo: %.1f l/km)\n", id_menos_eficiente, consumo_medio_menos_eficiente);
68         printf("Avion mas eficiente: %d (consumo: %.1f l/km)\n", id_mas_eficiente, consumo_medio_mas_eficiente);
69     }
70     else
71         printf("Ningún avión ha volado todavía.\n");
72 }
73
74 int main() {
75     int num_vuelos[N] = {0};
76     int distancias[N] = {0};
77     int consumos[N] = {0};
78     int i, opcion;
79
80     if (cargarDatos(num_vuelos, distancias, consumos) == 1){
81         printf("Error al abrir el fichero.\n");
82         return 1;
83     }
84
85     do {
86         opcion = 0 /*cambiar el 0 por la llamada a la función menu */;
87
88         switch(opcion) {
89             case 1:
90                 mostrarInformacionAvion(num_vuelos, distancias, consumos);
91                 break;
92             case 2:
93                 mostrarEficienciaAviones(num_vuelos, distancias, consumos);
94                 break;
95             case 0:
96                 printf("Bye!\n");
97                 break;
98         }
99     } while(opcion != 0);
100
101     return 0;
102 }
```

Exercici 4

En executar el codi, el programa mostrarà per pantalla un menú amb 3 opcions i sol·licitarà a l'usuari que triï una d'elles. Després de comprovar la validesa de l'opció seleccionada, calcularà els valors indicats per l'usuari, els mostrarà per pantalla i tornarà a mostrar el menú per pantalla. El programa acabarà quan l'usuari seleccioni l'opció 0.

Opció 1: Informació d' un avió: Aquesta opció mostra dades estadístiques sobre els vols realitzats per un avió en concret.

Opció 2: Avions més i menys eficients: Aquesta opció busca els avions més i menys eficients, entenent que un avió és més eficient que un altre si el seu consum mitjà per quilòmetre ha estat menor.

```
AVIONES
-----
1. Informacion de un avion.
2. Aviones mas y menos eficientes.
0. Salir.
Elige una opcion:
```

Exercici 4 (funció main)

Invoca a la funció **carregarDatos** per emmagatzemar en tres vectors la informació continguda en els tres fitxers:

- **num_vuelos** vols: emmagatzema el nombre total de vols realitzat per cada avió (podent ser zero si encara no ha volat).
- **distàncies**: emmagatzema la distància total (en quilòmetres) recorreguda per cada avió.
- **consums**: emmagatzema el consum total (en litres) que ha tingut cada avió.

Dins d' un bucle **do-while** invoca la funció menú per obtenir un número 0, 1 i 2. En funció de l' opció escollida executarà una o altra opció.

```
74 int main() {
75     int num_vuelos[N] = {0};
76     int distancias[N] = {0};
77     int consumos[N] = {0};
78     int i, opcion;
79
80     if (cargarDatos(num_vuelos, distancias, consumos) == 1){
81         printf("Error al abrir el fichero.\n");
82         return 1;
83     }
84
85     do {
86         opcion = 0 /*cambiar el 0 por la llamada a la función menu */;
87
88         switch(opcion) {
89             case 1:
90                 mostrarInformacionAvion(num_vuelos, distancias, consumos);
91                 break;
92             case 2:
93                 mostrarEficienciaAviones(num_vuelos, distancias, consumos);
94                 break;
95             case 0:
96                 printf("Bye!\n");
97                 break;
98         }
99     } while(opcion != 0);
100
101     return 0;
102 }
```

Declaració de variables

Invocació a **cargarDatos**

Es canvia un cop s' hagi implementat la funció **menu()**
Obtenció d' un número 0, 1 o 2

num_vuelos: almacena el número total de vuelos realizado por cada avión (puede ser 0)
distancias: almacena la distancia total (kilómetros) recorrida por cada avión.
consumos: almacena el consumo total (litros) que ha tenido cada avión

Desenvolupament de l' exercici 4

- Completar la funció carregarDats: aquesta funció emmagatzema en tres vectors la informació continguda en els tres fitxers.
 - num_vuelos vols: emmagatzema el nombre total de vols realitzat per cada avió (podent ser zero si encara no ha volat).
 - distàncies: emmagatzema la distància total (en quilòmetres) recorreguda per cada avió.
 - consumsconsums: emmagatzema el consum total (en litres) que ha tingut cada avió.

```
18 int cargarDatos(int num_vuelos[], int distancias[], int consumos[]) {
19     FILE *file;
20     int id, distancia, consumo;
21
22     file = fopen("vuelos.txt", "r");
23     if (file == NULL)
24         return 1;
25
26     while (0 /*sustituir el 0 por la condición que corresponda -lectura de una línea-*/) {
27         num_vuelos[id]++;
28         distancias[id] += distancia;
29         consumos[id] += consumo;
30     }
31
32     fclose(file);
33     return 0;
34 }
```

41	6297	88159
62	850	14450
47	2965	47441
81	1053	13689
1	3395	33950
0	8705	156693

Nota: només cal canviar la línia 26 per la lectura de línies del fitxer file fins a assolir final de fitxer

- Completar la funció menú

```
int menu() {  
    int opc;
```

```
/* muestra un menú al usuario y solicita una opción. Si la opción no  
está entre las del menú, indica un mensaje de error y vuelve a  
mostrar menú / pedir opción, repitiéndose todo hasta que se  
escoge una opción correcta */
```

```
    return opc;
```

```
}
```

```
AVIONES  
-----  
1. Informacion de un avion.  
2. Aviones mas y menos eficientes.  
0. Salir.  
Elige una opcion:
```

- Completar la funció `mostrarInformacionAvion`

```
void mostrarInformacionAvion(int num_vuelos[], int distancias[], int consumos[]) {  
    int id;  
  
    do {  
        printf("Avion (0-%d): ", N-1);  
        scanf("%d", &id);  
  
        if (id < 0 || id >= N)  
            printf("Error. Valor fuera de rango.\n");  
    } while (id < 0 || id >= N);  
  
    if (num_vuelos[id] == 0) {  
        printf("El avion %d no ha realizado vuelos.\n", id);  
    } else {  
        /* Falta el cálculo del consumo medio y mostrar por pantalla  
        el número total de vuelos, la distancia total recorrida, el  
        consumo total de combustible y el consumo medio por kilómetro. */  
    }  
}
```

```
AVIONES  
-----  
1. Informacion de un avion.  
2. Aviones mas y menos eficientes.  
0. Salir.  
Elige una opcion: 1  
  
Avion (0-81): 82  
Error. Valor fuera de rango.  
Avion (0-81): 18  
El avion 18 ha realizado 26 vuelos (75223 km) usando 1062236 l de combustible.  
Consumo medio por Km: 14.1 l/km
```

- Per provar el correcte funcionament d'aquesta opció, haureu de canviar en la funció `main()` la línia `opcion = 0`, per la línia `opcion = menu();`

- Completar la funció `mostrarEficienciaAviones`

```
void mostrarEficienciaAviones(int num_vuelos[], int distancias[], int consumos[]) {  
    int i, id_mas_eficiente = -1, id_menos_eficiente = -1;  
    float consumo_medio_mas_eficiente = FLT_MAX, consumo_medio_menos_eficiente = 0, consumo_medio;  
  
    for (i = 0; i < N; i++) {  
        /* Falta completar el bloque del for: recorrido de los vectores  
        para encontrar el avión más eficiente y el menos eficiente en  
        términos de consumo medio por kilómetro */  
    }  
  
    if (id_menos_eficiente != -1) {  
        printf("Avion menos eficiente: %d (consumo: %.1f l/km)\n", id_menos_eficiente, consumo_medio_menos_eficiente);  
        printf("Avion mas eficiente: %d (consumo: %.1f l/km)\n", id_mas_eficiente, consumo_medio_mas_eficiente);  
    }  
    else  
        printf("Ningún avión ha volado todavía.\n");  
}
```

```
AVIONES  
-----  
1. Informacion de un avion.  
2. Aviones mas y menos eficientes.  
0. Salir.  
Elige una opcion: 2  
  
Avion menos eficiente: 46 (consumo: 15.5 l/km)  
Avion mas eficiente: 4 (consumo: 13.0 l/km)
```

Exercici 1

Desenvolupa un programa amb una única funció (la funció main) que demani un número sencer a l'usuari (el número ha d'estar entre 1 i 10) i mostri per pantalla el factorial d'aquest número.

Exemple de funcionament (en cursiva l'introduït per l'usuari):

Dóna'm un número: 4

El factorial de 4 és 24

Nota: $n! = 1 * 2 * 3 * \dots * n$

Exercici 2

Desenvolupa un programa que implementi dues funcions:

Una funció, anomenada **factorial**, que rebi un valor sencer i retorni el factorial d'aquest valor (excepte si el valor rebut no és positiu, cas en què retornarà el valor 0):

int factorial(int n)

Una funció principal (**main**) que demani un número sencer qualsevol i usi l'anterior funció per calcular la seva factorial.

Exercici 3

Desenvolupa un programa que calculi i mostri:

$$1! + 2! + 3! + \dots + (n-2)! + (n-1)! + n!$$

on n és un valor que s'ha de demanar a l'usuari i ha d'estar entre 1 i 10.
Utilitza en el programa la funció factorial desenvolupada en l'exercici 2.

Exemple de funcionament:

```
Calculo de 1! + 2! + 3! + ... + (n-1)! + n!
```

```
Dame el valor de n: 0
```

```
Dame el valor de n: 22
```

```
Dame el valor de n: 5
```

```
El resultado es: 153
```

```
#include <stdio.h>
```

```
Incloure la funció factorial de l' exercici 2
```

```
int main(){
```

```
    int n, f, total=0;
```

```
    printf("Càlcul de 1! + 2! + 3! + ... + (n-1)! + n!\n\n");
```

```
Llegir una dada n pel teclat fins que el seu valor es trobi entre 1 i 10
```

```
printf("El resultat es: %d\n", total);
```

```
return 0;
```

```
Bucle for que recorri els valors  $i = 1, 2, \dots, n$ 
```

```
    total = total + factorial( i);
```

```
}
```

Moltes gràcies
y
a programar !

