

FONAMENTS D'ORDINADORS

TEMA3: Tipus estructurats de dades

Manel Guerrero

[H9] Struct

```
typedef struct {
    tipus_1 camp_1;
    ...
    tipus_N camp_N;
} nom_tipus;
```

```
typedef struct {
    int x;
    int y;
} tpunt;
tpunt p = {1,1};
p.x = 2; p.y = 3;
```

- Struct permet declarar una única variable amb diversos elements (camps).
- Exemples:
 - DNI: num, lletra
 - Complex: real, imag
 - Data: dia, mes, any
 - Punt: x, y

Structs niats e inicialització

```
typedef struct {
    char c;
    int n;
} tedifici;
typedef struct {
    char c; // 'E', 'S', ' '
    int n;
} tplanta;
typedef struct {
    tedifici e;
    tplanta p;
    int num;
} tdespatx;
tdespatx d = {{'D',6},{' ',2},12};
d.p.c = 'E';
```

- Un camp d'un struct pot ser de tipus struct. Exemple:
 - Persona amb DNI, despatx i data de naixement.
- Les variables de tipus struct es poden inicialitzar
- Accedim als elements d'una variable de tipus struct així: `nom_var.nom_element`

Comparar Structs

```
typedef struct {
    int x;
    int y;
} tpunt;
tpunt p1 = {1,1};
tpunt p2 = {2,2};
INCORRECTE:
if (p1 == p2)
CORRECTE:
if ((p1.x==p2.x)&&
    (p1.y==p2.y))
CORRECTE:
p1 = p2;
```

- Trampa: Per comparar si dos structs són iguals hem de comparar que tots els seus elements siguin iguals!

Problemes de Structs

- Mirar si dues persones fan anys el mateix dia.
- Col·lecció de problemes:
 - #4 (suma de dos números complexos)

[H10] Exercicis de Structs

- Dir si un punt està dins d'un cercle de centre 'c' i radi 'r'.
- Col·lecció de problemes:
 - #3: (Dir si un punt pertany a la recta determinada per dos punts)
 - #5: (Introduir els 3 cantons d'un triangle i que retorni el perímetre i el tipus de triangle (e'Q'uilàter, 'l'sòsceles, 'E'scalè) totes aquestes variables són de t_triangle)

[H11] Vectors

```
tipus nom[mida];
int v[10];
v[0] = 1;
v[9] = 0;
v[10] i v[-1] no
existeixen! Però el
compilador no us avisarà!
```

```
char paraula[8];
tpersona profe[50];
profe[3].dni.lletra = 'A';
```

- Un vector és una estructura de dades consistent en una seqüència d'elements del mateix tipus que són accedits mitjançant un índex que va de '0' la mida del vector menys 1.
- C us deixarà accedir a elements fora de rang sense queixar-se. Error. Trampa mortal!

Comparar Vectors

```
int x[10], y[10], z[10];
```

INCORRECTE:

```
z = x + y;
```

CORRECTE:

```
for (i=0; i<10; i++) {
    z[i] = x[i] + y[i];
```

```
}
```

INCORRECTE:

```
if (z == x) {
```

```
...
```

- Trampa: No es poden fer operacions entre vectors. S'han de fer entre els seus elements!

Exercicis de Vectors

- Cercar 'a's.
- Comptar 'a's.
- Contar palíndroms en una frase (si dona temps).

[H12] Més exercicis de Vectors

- Afegir i eliminar elements en un vector d'enters (versió simple sense menú).
- Qui celebra aniversari el mateix dia (o versió simplificada: comparar dos a dos tots els elements d'un vector d'enters).

[H13] Interludi matemàtic

```
#include <math.h>
/* potencia, arrel quadrada,
valor absolut */
double pow(double x, double
y)
double sqrt(double x)
double fabs(double x)
/* Trigonòmètriques (x en
radiants */
double cos(double x)
double sin(double x)
double tan(double x)
```

- Trampa mortal: No t'oblidis de compilar amb la llibreria matemàtica ("-lm"). Podria ser que compilés sense errors però que el resultat de les funcions de math.h retornessin valors incorrectes.
- Correcte:
 - gcc -o foo foo.c -lm
- Incorrecte:
 - gcc -lm -o foo foo.c

Més exercicis de Vectors

- Llegir i imprimir vectors de char acabats en '\n' amb for i while (t3v_leer_cadena.c).
- Problemes de la col·lecció:
 - #2: (Introdueix un número 'n'. Ara introduceix 2 vectors de 'n' enters i fes el producte escalar) $p=v1*w1+v2*w2+...+vn*wn$
 - #3: (Introdueix un número 'n'. Ara introduceix 1 vectors de 'n' enters i inverteix-lo)