FONAMENTS D'ORDINADORS

TEMA3: Tipus estructurats de dades

Manel Guerrero

[H1] Struct

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

typedef struct {
    float x;
    float y;
} t_punt;
t_punt p = {1,1};
p.x = 2; p.y = 3;
```

- [Abans fer el programa aniversaris sense structs.]
- De la mateixa manera que els àtoms es poden combinar en molècules, les variables atòmiques es poden combinar en estructures.
- Struct permet declarar una única variable amb diversos elements (camps).
- Exemples:
  - Data: int dia, mes, any
  - Punt: float x, y

Structs niats e inicialització

```
typedef struct {
    char c;
    int  n;
} t_edifici;
typedef struct {
    char c; // 'E', 'S', ' '
    int  n;
} t_planta;
typedef struct {
    t_edifici e;
    t_planta  p;
    int     num;
} t_despatx;
t_despatx 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;
} t_punt;
t_punt p1 = {1,1};
t_punt 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 (si dona temps):
  - #4 (suma de dos números complexes)
  - #5: (Introduir els 3 cantons d'un triangle i que retorni el perímetre i el tipus de triangle (e'Q'uilàter, 'I'sòsceles, 'E'scalè) totes aquestes variables són de t_triangle)

[H2] Interludi matemàtic festiu

```
#include <math.h>
/* potencia, arrel quadrada, valor absolut */
double pow(double x, double y)
double sqrt(double x)
double fabs(double x)
/* Trigonomètriques (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
- a^b (caret) és bitwise XOR!!!

Exercicis de Structs

- Dir si un punt està dins d'un cercle de centre 'c' i radi 'r'.
- El de dir si un punt pertany a una recta millor no.

[H3] 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];
t_persona 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 index 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!

Inicialització de vectors

```
#define ARRAY_SIZE 10
/* Initialize all the elements. */
int array_all[ARRAY_SIZE] = {1, 2,
3, 4, 5, 6, 7, 8, 9, 10};
/* If only the first elements are
initialized, the remaining elements
are all set to zero. */
int array2[ARRAY_SIZE] = { 1, 2 };
/* So, if first element is set to
0, then everything else is, too. */
int array_zero[ARRAY_SIZE] = { 0 };
/* If nothing is initialized, the
array contains random junk. */
int array_random[ARRAY_SIZE];
```

- Podem inicialitzar tots els elements d'un vector en el moment de la declaració.
- Després de la declaració haurà de ser un a un amb un bucle.
- Si no els inicialitzem contindran valors al atzar.
- http://www.lemoda.net/c/array-initialization/ [font]

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

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

[H4] 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).

[H5] Interludi haiku iteratiu

- t9_coseno.c
- t9_coseno_haiku.c:
  - Haiku de 17 mores (5,7,5) en castellà:
    - Bucles vacios
    - de repente iteran
    - ... es un coseno.
    -
  - http://ca.wikipedia.org/wiki/Haiku

$$cos\,(x) = \sum_{i=0}^{n} \left(\frac{x^{2i}}{(2i)!} (-1)^i \right)$$

[Els interludis són intercanviables]

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 introdueix 2 vectors de 'n' enters i fes el producte escalar) p=v1*w1+v2*w2+...+vn*wn
  - #3: (Introdueix un número 'n'. Ara introdueix 1 vectors de 'n' enters i inverteix-lo)

[H6] Matrius

```
VECTORS:
tipus nom[mida];
int v[10]
v[0] = 1; v[9] = 0;

MATRIUS:
tipus nom[files][cols];
float m[FILES][COLS]={{0}};
m[0][0] = 2.10;
m[FILES-1][COLS-1] = 0;
```

- Una matriu funciona exactament de la mateixa manera que un vector.
- Peró en lloc de tenir un únic index, en té dos. En matrius bidimensionals els anomenem fila i columna.

Exercicis de Matrius

- Suma de matrius.
- Producte de matrius.
- Intercanviar files i columnes d'una matriu sobre si mateixa.

(En els exemples a la web s'utilitza una funció anomenada imprimir_matriu() perquè el codi sigui mes net. En el següent tema veurem com programar funcions).

[H7] Algorismes bàsics de vectors

- https://en.wikipedia.org/wiki/Sorting_algorithm
- https://en.wikipedia.org/wiki/Bubble_sort
- t3v_sort_bubble.c Ordenació vectors: bubble sort.
- https://en.wikipedia.org/wiki/Insertion_sort
- t3v_sort_insertion.c Ordenació vectors: insertion sort.

[H8] Algorismes bàsics de vectors 2

Conjunts:

- Quina és l'estructura de dades més adient per guardar un conjunt d'elements?
- Implementeu t3v_conjunts_resolt.pdf a casa.
- El resoldrem a classe.