

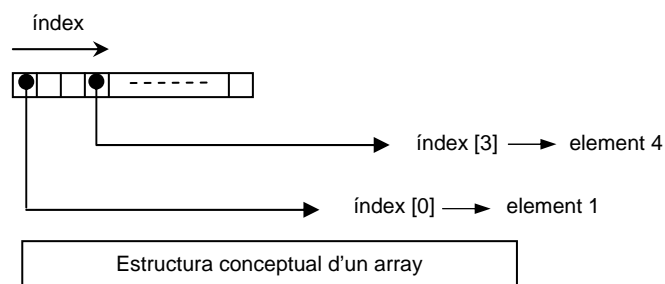
ALGORISMES I PROGRAMACIÓ (GESTIÓ AERONÀUTICA)

SESSIÓ 14/03/2005

1. Arrays 1-Dimensionals: Vectors

Com ja sabeu, en C, la informació es guarda en variables. Fins ara per a cada variable necessitàvem un identificador (nom) diferent. Ara però introduïm un nou tipus de dada, el **vector** o **array**:

Un **vector** és una col·lecció de variables del mateix tipus i que es referencien mitjançant un nom comú. A un element específic d'un array s'hi accedeix a través d'un nombre natural anomenat **índex**. Els índexs comencen a comptar des de 0.



1.1 Declaració

La forma general de declarar un array unidimensional (o vector) és la següent:

```
tipus nom_vector[longitud_vector];
```

on,

- `tipus`, declara el tipus base del vector. O sigui, tipus de les variables que constitueixen el vector (int, float, ...).
- `nom_vector`, és un nom vàlid en C.
- `longitud_vector`, és la longitud (=número d'elements) que volem que tingui el vector.

Exemple>

- `float alçades_ETSE[2664];`

Observeu que declarant una sola variable `alçades_ETSE` us permet emmagatzemar l'alçada de tots els alumnes de l'E.T.S.E. D'altra forma hauríeu de declarar 2664 variables float:

```
float A_alumne0001, A_alumne0002, . . . , A_alumne2664;
```

IMPENSABLE!!!

1.2 Accés als elements

Una vegada tenim un array declarat, la manera d'accedir a les seves posicions és a través d'un únic índex que indica la posició de l'element que volem manipular. **Els índexs comencen en zero i acaben en longitud_vector-1.**

La finalitat d'accedir a les posicions d'un array és doble: per una banda ens interessarà inicialitzar-lo (o assignar-li valors) i per altra ens interessarà agafar els valors que conté per efectuar qualsevol operació amb ells

1.3 Assignació de valors a un vector (inicialització)

Hi ha varies maneres d'assignar valors a les posicions d'un vector:

- En el moment de la declaració:

```
float v[5]={1.0,3.1416,-2.5,-3.0,7.25845};
```

En aquest cas no es estrictament necessari especificar la longitud del vector.

- Podem assignar-li des del codi

```
int v[10];  
  
v[0]=8;  
v[1]=13;  
.  
.  
.  
v[9]=666;
```

- O entrant valors per pantalla

```
char characters[10];  
  
scanf("%c",&characters[0]);  
scanf("%c",&characters[1]);  
.  
.  
.  
scanf("%c",&characters[9]);
```

```
char characters[10];  
  
characters[0]=getche();  
characters[1]=getche();  
.  
.  
.  
characters[9]=getche();
```

1.4 Accés ràpid als elements d'un vector

Una manera d'estalviar-se haver d'escriure codi és usant bucles. Recordeu l'exemple de les alçades dels alumnes de l'ETSE. Resulta completament ineficient escriure 2664 `scanf`'s per llegir les alçades.

```
int i;  
float alçades_ETSE[2664];  
  
for(i=0;i<2664;i++)  
    scanf("%f",&alçades_ETSE[i]);
```

Exercici #1> Feu un petit programa en el què creeu tres vectors enters (v1, v2 i v3) de 10 posicions cadascun i empleneu-los de la següent manera:

1. A cada posició de v1 hi col·loqueu el seu índex.
2. A cada posició de v2 hi col·loqueu el doble del seu índex.
3. A cada posició de v3 hi col·loqueu el doble d'un enter entrat per l'usuari.

1.5 Utilització dels valors d'un vector

Una vegada tenim un vector ple, és a dir, inicialitzat, segur que d'una manera o altra voldrem accedir al seu contingut (si no, no tindria sentit haver-lo creat). Per accedir-hi, simplement hem d'escriure el nom del vector seguit del seu índex. Vegem-ne alguns exemples:

- Agafem el valor d'una posició del vector per assignar-li a una altra variable (del mateix tipus!)

```
float v[10],c;  
.  
.  
.  
} Inicialitzem v  
c=v[6];
```

- Mostrem els valor d'una posició del vector per pantalla:

```
int i;  
float alçades_ETSE[2664];  
  
for(i=0;i<2664;i++)  
    scanf("%f",&alçades_ETSE[i]);  
  
for(i=0;i<2664;i++)  
    printf("%f\n",alçades_ETSE[i]);
```

Exercici #2> Feu un petit programa empleni 2 vectors numèrics de longitud 5 a través de pantalla. Creeu un tercer vector que sigui la suma dels dos anteriors (element a element) i mostreu el resultat.

1.6 Índexs

Recordeu, els índexs dels vectors sempre varien entre 0 (primera posició) i L-1 (última), on L és la longitud del vector. **Si l'índex no es troba dins d'aquests valors estem accedint a una posició de memòria que no ens pertany! i que podria contenir informació important.** És responsabilitat vostra controlar que l'índex no prengui valors fora d'aquest rang, ja que el compilador no us dóna error i serà en executar que el programa us donarà valors inesperats.

Exercici #3> Feu un programa on es declari el vector 'y' de 3 floats amb els valors:

{1.0, 2.0, 3.0}

Feu un bucle que visualitzi fins a la posició 5 i executeu el programa.

2. Vectors de caràcters: Strings

En C, una cadena es defineix com un vector de caràcters l'últim dels quals és el caràcter nul: `'\0'`.

2.1 Declaració i Inicialització

Per declarar cadenes és necessari que la longitud sigui un caràcter més del màxim previst. Per exemple, per a declarar un vector `'frase'` que contingui una cadena de 5 caràcters, escriurem:

```
char frase[6]; // Això és un String
```

Per inicialitzar-lo tenim dues opcions:

- Element a element :
frase[0] = 'Q';
frase[1]='u';
frase[2]='i';
frase[3]='n';
frase[4]='a';
frase[5]='\0';

Usant aquesta opció no cal que la longitud del string coincideixi amb la mida del vector de chars:

```
frase[0] = 'Q';  
frase[1]='u';  
frase[2]='i';  
frase[3]='\0';
```

- Usant una constant de cadena (llista de caràcters entre cometes “)
frase="Quina "; //inicialitzacio

```
char frase[]="Quina"; //inicialitzacio en la mateixa declaració
```

En aquest cas (usar una constant de cadena) el caràcter nul `'\0'` no cal (ja va inclòs en la constant de cadena) però la longitud de la constant de cadena ha de coincidir amb la de la variable string.

2.2 Entrada / Sortida de strings

Si volem entrar algun string per pantalla, podem fer-ho de la següent manera:

- Usant la funció `gets()`. Aquesta pren com a paràmetre el nom de la variable de tipus `string`. Per usar-la, necessiteu incloure la llibreria `string.h` :

```
char frase[11];  
  
gets(frase);
```

Per altra banda, si volem treure un string per pantalla, també podem fer-ho de dues maneres:

- Amb `printf`:

```
printf("%s", frase);
```

o be,

- La funció `puts()`. També pren com a paràmetre el nom de la variable de tipus `string` i, per usar-la, necessiteu incloure la llibreria `string.h`.

```
puts(frase);
puts("Un string constant");
```

Exercici #4> Feu programa en què introduïu 3 strings per pantalla (Feu-los prou grans per a que hi càpiguen frases llargues). Un quart string de longitud 7, l'ompleneu de la següent manera:

1	2	3	4	5	6	'\0'
---	---	---	---	---	---	------

1. Primer element del primer string
2. Segon element del primer string
3. Primer element del segon string
4. Segon element del segon string
5. Primer element del tercer string
6. Segon element del tercer string

Mostreu el resultat per pantalla.

2.3 Manipulació de strings

La llibreria `string.h` conté diverses funcions per a manipular strings. De moment, nosaltres utilitzarem:

- `strlen(la_cadena)`; que retorna el número de caràcters de l'string `la_cadena` fins el `'\0'`. Observeu que el que retorna `strlen` no té per que coincidir amb la longitud de l'array de chars:

Exemple>

```
//Declaració d'un array de 5 chars
char frase[5];

//Inicialització a string de longitud 2

frase[0]='m';
frase[1]='i';
frase[2]='\0';
```

- `strcmp(cadena_1,cadena_2)`; que compara les dues cadenes i si són iguals, retorna el valor enter 0.

Exercici #5> Feu un programa que demani a l'usuari que introdueixi dues cadenes de caràcters. A continuació, mostreu per pantalla la longitud de cadascuna de les cadenes i mostreu un missatge per pantalla que indiqui si les cadenes de caràcters són iguals (en contingut) o diferents.

3. Altres exercicis

Exercici #6> Feu un programa que empleni un vector v de longitud 25 on, a cada posició hi figuri la següent funció de l'índex:

$$v[i] = \frac{(i+1)(i+2)}{2}$$

Exercici #7> Feu un programa en què hagueu d'emplenar un vector de 10 enters, per pantalla i, una vegada ple, mostreu per pantalla tan sols les posicions parelles del vector.

Exercici #8> Escriure un programa que llegeixi la temperatura mitja dels dotze mesos de l'any i digui quin mes és el que ha tingut la temperatura mínima, i quins mesos han tingut una temperatura inferior a la mitja anual.

Exercici #9> Feu un programa que llegeixi una cadena de caràcters i mostri per pantalla cadascuna de les paraules que componen la cadena. Recorreu l'string i compareu cadascun dels seus caràcters amb l'espai en blanc ' '. Mentre no el trobeu aneu imprimint el caràcter sense canviar de línia. La coincidència amb ' ' marcarà el fi de paraula i, per tant, el salt de línia.

Exercici #10> Feu un programa que llegeixi una cadena de caràcters introduïda per l'usuari. Recorreu l'string i canvieu tots els caràcters 'a' pel caràcter 'e' i tots els caràcters 'i' pel caràcter 'o'. Mostreu per pantalla el resultat final.