Arreglos paralelos

Los arreglos paralelos (dos o más) son aquellos que están relacionados por un mismo índice, o sea, las celdas ubicadas en la misma posición (para distintos arreglos) tienen un vínculo entre sus datos. Se utilizan generalmente para asociar datos de distintos tipos y almacenarlos en arreglos de tipos de datos básicos.

En realidad no representa un nuevo tipo de datos, sino una forma en particular de trabajar con un grupo de arreglos.

Por ejemplo:
Almacenar los datos (número de legajo, nombre y año) de un grupo de alumnos. Se crean tres arreglos del mismo tamaño, cada uno del tipo correspondiente para que pueda almacenar uno de los datos.

legajos	121	122	123	124	125
nombres	Alvarez	Perez	Lopez	Garcia	Simpson
años	2	1	2	3	3
	0	1	2	3	4

Como es de suponer, los datos de igual número de celda de cada uno de los arreglos están vinculados, por pertenecer a una misma “entidad de datos”. Esto tiene que ver con el diseño, o sea con el significado que el programador le da a los datos (semántica).

En nuestro ejemplo tenemos almacenados los datos de cinco alumnos; en la celda número uno (subíndice 0) está almacenado el alumno Alvarez cuyo legajo es 121 y cursa el 2 año, y así con los demás.

Ordenación sobre arreglos paralelos

Para ordenar arreglos paralelos se debe establecer uno de ellos como el arreglo que determinará el criterio de ordenación. Todos los arreglos deberán modificar la ubicación de sus elementos según la ordenación establecida por uno de ellos.

Por ejemplo, si se desea ordenar el conjunto de arreglos paralelos del ejemplo anterior, por orden de legajo, según el algoritmo de ordenación por selección, se tiene:

```
int posicionMenor(int leg[ ], int cantVal, int u){
    int menor = leg[u];
    int posMenor = pos;
    for(int index = u + 1; index < cantVal; index ++){
        if(menor < leg[index]){
            menor = leg[index];
            posMenor = index;
        }
    }
    return posMenor;
}
```

```c
void intercambioPalabra(char pal[ ][30], int i, int j){
    char aux[30];
    strcpy(aux, pal[i]);
    strcpy(pal[i], pal[j]);
    strcpy(pal[j], aux);
}
```

```c
void intercambioEntero(int numero[ ], int i, int j){
    int aux = numero[i];
    numero[i] = numero[j];
    numero[j] = aux;
}
```

```c
void ordenacionSeleccion(int leg[ ], char nombre[ ][30], int anio[ ],
int cantVal ) {
//los tres arreglos tienen la misma cantidad de elementos válidos
    int posMenor;
    int i=0;
    while(i<cantVal - 1){ /// llego hasta la anteúltima posición
        posMenor = posicionMenor(leg, i);
        intercambioPalabra(nombre, posMenor, i);
        intercambioEntero(leg, posMenor, i);
        intercambioEntero(anio, posMenor, i);
        i++;
    }
}
```

Queda para el alumno resolver el algoritmo de ordenación por inserción.

NOTA: como ya vimos en el apunte “Arreglos en C”, un **arreglo de palabras** se define igual que una matriz de caracteres, en donde el primer subíndice determina la cantidad de palabras del arreglo y el segundo subíndice determina la cantidad máxima de caracteres de cada palabra.

Para trabajar cada celda del arreglo de palabras simplemente se usa el nombre de la matriz de caracteres y solamente el primer subíndice, por ejemplo, sea un arreglo de 10 palabras de longitud 30 cada una:

```c
char palabras[10][30];
```

para acceder a la palabra de la celda i (0 <= i < 10) escribimos: **palabras[i]** dentro de la instrucción correspondiente.