

# Grado en Ingeniería Informática

## Relaciones de Estructuras de Datos.

J. Fdez-Valdivia

*Departamento de Ciencias de la Computación e I.A. ETS Ingeniería Informática  
Universidad de Granada. 18071 Granada. Spain.  
Email: jfv@decsai.ugr.es*

### 1. Problemas sobre Punteros.

**Problema 1.1** *Escriba un trozo de código donde se declara un vector de 100 enteros y, mediante un bucle, se asigne el valor 1 a todas las posiciones, sin usar el “operador []”.*

**Problema 1.2** *Declare una variable `numeros` como un vector de 1000 enteros. Escriba un trozo de código que recorra el vector y modifique todos los enteros negativos cambiándolos de signo. No se debe usar el operador ‘[]’, es decir, se deberá usar aritmética de punteros. El bucle se controlará mediante un contador entero.*

**Problema 1.3** *Modifique el código del problema anterior para controlar el final del bucle con un puntero a la posición siguiente a la última.*

**Problema 1.4** *Considere la figura 1. Se presentan gráficamente un conjunto de estructuras de datos. Se puede observar que las matrices se representan indicando los índices y las estructuras indicando los nombres de los campos. Escriba los trozos de código que corresponden a su creación. Nota: No se debe usar memoria dinámica (para cada caso se incluye el nombre de las variables necesarias).*

**Problema 1.5** *Supongamos tres vectores `v1`, `v2`, `res` de valores reales. En `v1`, `v2` se almacenan, respectivamente, `n`, `m` valores ordenados de menor a mayor. Escribir un trozo de código para mezclar, de manera ordenada, los valores en el vector `res` que tiene capacidad para almacenar al menos `n+m` valores. No se debe usar el operador ‘[]’, es decir, se debe usar aritmética de punteros.*

---

## 2. Problemas sobre Funciones.

**Problema 2.1** Indique el efecto de la siguiente función

```
void intercambia (int *p, int *q)
{
    int temp= p;
    p=q;
    q=temp;
}
```

**Problema 2.2** Indique el efecto de la siguiente función

```
void intercambia (int *p, int *q)
{
    int *temp= p;
    p=q;
    q=temp;
}
```

**Problema 2.3** Implemente una función `swap_puntero` que intercambie el valor de dos punteros usando paso por referencia. Muestre un ejemplo de llamada.

**Problema 2.4** Implemente la función del problema anterior sin usar paso por referencia. Deberá usar un paso de parámetros al “estilo C”, es decir, por medio de un puntero al valor a modificar. Reescriba el ejemplo de llamada.

**Problema 2.5** Consideremos que en un proyecto, un programador tiene la tarea de crear una función que determine si un valor entero, indicado por un puntero, es par. Crea la siguiente función

```
bool par(int *p)
{
    return (*p %2==0);
}
```

Considere otra función, realizada por otro programador, que usa la primera.

```
int numero_pares (const int *v, int n)
{
    int res=0;
    for (int i=0;i<n;i++)
        if (par(v+i)) res++;
    return res;
}
```

¿Qué problema se va a encontrar la persona encargada de integrar ambos códigos? Si es necesario, incluya las dos funciones en un fichero fuente e intente compilarlo. ¿Qué error es de esperar?

**Problema 2.6** Considere la siguiente función

```
void copiar (float **m, int f, int c, float **res)
{
    // modifica res, para apuntar a una nueva zona y
    // copia la matriz fxc m a res
}
```

- ¿Qué posible problema nos podemos encontrar al ejecutar `Copiar(mat,f,c,mat)`?

- ¿Cómo podríamos asegurarnos de resolver ese problema?

**Problema 2.7** Las cadenas de caracteres representan un ejemplo clásico en el uso de punteros. El tipo correspondiente para almacenarlas es un vector de caracteres <sup>1</sup>. Implemente las siguientes funciones:

- Función `copia_cadena`. Copia una cadena de caracteres en otra.
- Función `encadenar_cadena`. Añade una cadena de caracteres al final de otra.
- Función `longitud_cadena`. Devuelve un entero con la longitud (número de caracteres sin contar el nulo) de la cadena.
- Función `comparar_cadena`. Compara dos cadenas. Devuelve un valor negativo si la primera es más “pequeña”, positivo si es más “grande” y cero si son “iguales”.

Teniendo en cuenta que se supone que hay suficiente memoria en las cadenas de destino y no es necesario pasar el tamaño de las cadenas (controlado por la terminación en carácter nulo).

**Problema 2.8** En la tabla 1 se muestran posibles pasos de parámetros. Indique si son correctos y por qué.

	Declaración	Parámetro Actual	Parámetro formal
1	int v	v	float v
2	int m[]	m	int *mat
3	float mat[5]	mat	float *& mat
4	const int v[10]	v	int *mat
5	int m[]	m	int mat[10]
6	int m[3][5][7]	m	int mat[][5][7]
7	float v[5]	v+2	const float mat[]
8	int m[]	m	int mat[][5]
9	float f	f	double f
10	float f	&f	double& f
11	bool mat[5][7]	&mat[3][2]	const bool mat[]
12	char mat[3][5]	mat[0]	char *mat
13	int *m[10]	m	int **mat
14	const double v	&v	double v[]
15	int **m	m	int mat[][]
16	int mat[5][7][9][11]	&mat[0][0][0][0]	int *p
17	int *p	&p	int *mat[]
18	float *p	p	float *& p
19	int m[3][5][7]	m	int *mat[5][7]
20	int mat[5][7][9][11]	mat[2][4][3]	const int *m
21	float *p	p+5	float *& p
22	double *p	p+2	float *p
23	int m[5][10]	m	int *mat[]
24	int *mat[5]	&mat[5]	const int **p
25	const bool *mat	mat+4	bool *p
26		5	int& val

Cuadro 1: Problema 2.8. Lista de posibles pasos de parámetros

<sup>1</sup>Recordemos que un literal de cadena, es una secuencia de caracteres entre comillas. Por ejemplo, "Hola". El tipo de este ejemplo es `const char [5]` (incluye el `'\0'`).

---

**Problema 2.9** Considere las siguientes funciones

```
int primera ()  
{ int local; ... return local; }  
int segunda ()  
{ static int local=0; ... return local; }
```

¿Cree que son correctas? Razone la respuesta.

**Problema 2.10** Suponga que ejecutamos el siguiente programa

```
#include <iostream>  
using namespace std;  
  
struct Par{  
    int *p1,*p2;  
};  
void intercambia(Par p)  
{  
    int *paux;  
    paux=p.p1; p.p1=p.p2; p.p2=paux;  
  
    int aux;  
    aux=*p.p1; *p.p1=*p.p2; *p.p2=aux;  
}  
  
int main()  
{  
    Par p;  
    int x=0,y=1;  
    p.p1=&x;  
    p.p2=&y;  
    intercambia(p);  
    if (p.p1==&x)  
        cout << "Los punteros están igual" << endl;  
    if (*p.p1==0)  
        cout << "El primero sigue teniendo 0" << endl;  
}
```

¿Cuál cree que será la salida? Razone la respuesta.

---

**Problema 2.11** *Suponga que ejecutamos el siguiente programa*

```
#include <iostream>
using namespace std;

struct Par{
    int vec[2];
};
void intercambia(Par p)
{
    int aux;
    aux=p.vec[0]; p.vec[0]=p.vec[1]; p.vec[1]=aux;
}

int main()
{
    Par p;
    p.vec[0]=0;
    p.vec[1]=1;
    intercambia(p);
    if (p.vec[0]==0)
        cout << "El primero sigue teniendo 0" << endl;
}
```

*¿Cuál cree que será la salida? Razone la respuesta.*

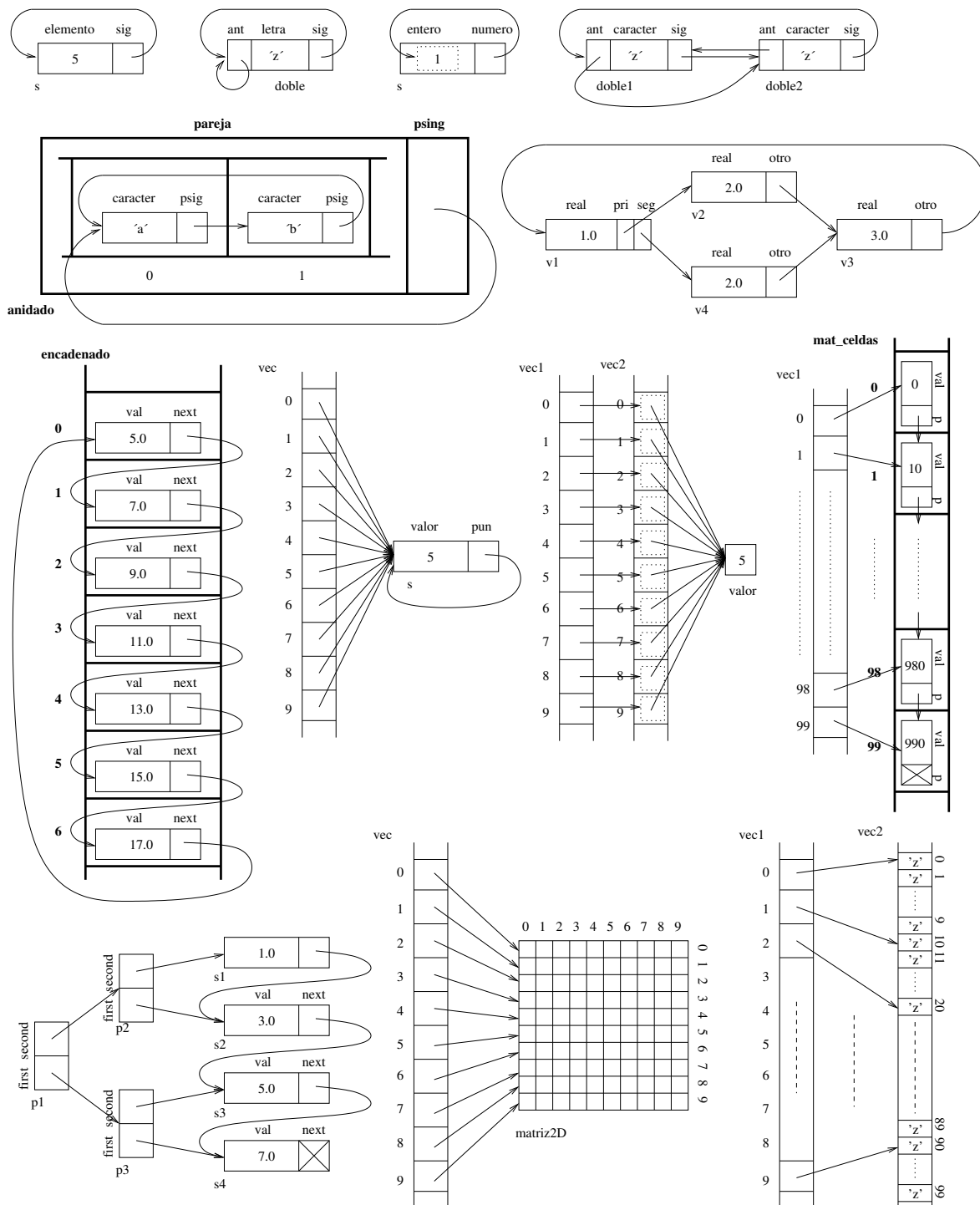


Figura 1: Ejemplos de estructuras.