

## Teoricos

```
1) printf("%li\n",sizeof(cadena));      // 10
   printf("%li\n",strlen(cadena));       // 6
   printf("%li\n",sizeof(p));           // 8
   printf("%li\n",strlen(p));           // 6
   printf("%li\n",sizeof(*p));          // 1
```

2) Una unión ocupa lo que ocupa el mayor de sus elementos, una estructura ocupa la suma de lo que ocupan sus elementos. Además, en una unión se puede acceder al mismo dato llamándolo de distinta forma.

➤ Declaración:

- union Un{  
    char U\_letra;  
    char U\_cadena[10];  
};

- struct Es{  
    char E\_letra;  
    char E\_cadena[10];  
};

➤ Tamaños

- union Un: 10Bytes
- estructura Es: 11Bytes

➤ Ejemplo con nombres

- Definición.  
union Un U1;  
struct Es E1;
- Acceso  
U1.U\_letra  
U1.Ucadena  
E1.E\_letra  
E1.E\_cadena

➤ Ejemplo con punteros

- Definición.  
union Un \*PU;  
struct Es \*PE;
- Asignación  
PU=&U1;  
PE=&E1;
- Acceso  
PU->U\_letra  
PU->Ucadena  
PE->E\_letra  
PE->E\_cadena

3) A75Dh-F363h

- a) En el caso de tener simplemente números en base hexadecimal, el A75Dh es el 42845 decimal y el F363h es el 62307 y como  $42845 - 62307 = -19462$  tenemos que en hexadecimal queda -4C06h.
- b) En el caso de trabajar en 16, los dos datos son números negativos por tener su bit más significativo por lo tanto tenemos que, aplicando complemento a 2, el A75Dh es el -58A3h que en decimal es -22691 y el F363h es -0C9Dh correspondiente al -3229 en decimal quedando  $-22691 - (-3229) = -19462$  en decimal que resulta ser -4C06h luego, por ser un número negativo, hay que usar complemento a 2 en 16 bit para el resultado final en hexadecimal: el 62307 y como  $42845 - 62307 = -19462$  tenemos en hexadecimal B3FAh.
- c) En el caso de trabajar en 32 bit lo que tenemos realmente es 0000A75Dh y 0000F363h, que son dos números positivos, de esta forma tenemos, igual que en el caso a que el A75Dh es el 42845 decimal y el F363h es el 62307 y como  $42845 - 62307 = -19462$  tenemos que en hexadecimal queda -4C06h, que al ser negativo hay que aplicarle complemento a dos para obtener su representación en 32 bits: FFFFB3FAh.