Programación

Test Autoevaluación
Tema 6

Autores:

M. Paz Sesmero Lorente
Paula de Toledo Heras
Fco. Javier Ordóñez Morales
Juan Gómez Romero
José A. Iglesias Martínez
José Luis Mira Peidro

Universidad Carlos III de Madrid
www.uc3m.es

CC BY NC SA

SOLUCIONES

1. **Dada la siguiente definición de una estructura de datos, ¿cómo se declara un vector de 10 elementos de tipo 'cancion' y se asignan valores al primer elemento?**

```
struct  cancion   {
        char interprete[20];
        float duracion;
};
```

   a. **No puede realizarse, un vector no puede almacenar estructuras de datos.** Falso. Los elementos de un vector pueden ser estructuras de datos

   b. **No puede realizarse, es necesario incluir valores en todos los elementos del vector.** Falso. Una vez declarado el vector, es posible inicializar tantas componentes como deseemos.

   c.
```
struct cancion descargas[10];
strcpy (descargas[0].interprete, "U2");
descargas[0].duracion = 20.7;
```
   Verdadero. (Ver justificación)

   d.
```
struct cancion descargas[10];
strcpy (descargas.interprete[0], "U2");
descargas.duracion[0] = 20.7;
```
   Falso. (Ver justificación)

Solución: Los elementos de un vector pueden ser estructuras de datos. La forma de declarar un vector cuyos elementos son del tipo canción es:

```
struct cancion descargas[10];
```

Una vez declarado el vector, es posible inicializar tantas componentes como deseemos ajustándonos a la estructura genérica:

```
vector[componente].campo_estructura1=<valor de tipo campo_estructura1>
vector[componente].campo_estructura2=<valor de tipo campo_estructura2>
……
```

En el caso particular que nos ocupa, y teniendo presente que el primer miembro de la estructura es una cadena de caracteres, la forma correcta de inicializar esta componente es la indicada en la opción c.

2. **Dado el siguiente fragmento de código, indique cuál de las siguientes afirmaciones es cierta:**

```
int main(void){
   int lista[5]={1, 3, 5, 7, 9};
   int *p1, *p2;
   int x;
   x= *(lista);
   p1=&lista[0];
   p2=lista;
   printf("%d, %d, %d\n", x, *p1, *p2);
   return 0;
}
```


BY NC SA

a. **La sentencia `x= *(lista);` genera un error de compilación porque su sintaxis es incorrecta.** Falso. `*(lista)` es un puntero al primer elemento de un vector de enteros. Por tanto la sentencia es correcta y permite asignar a x la primera componente de dicho vector. En este caso el valor 1.
b. **Tras ejecutar `p1=&lista[0];` el valor almacenado en `p1` es `1`.** Falso. Tras ejecutar esta sentencia el valor almacenado en p1 es la dirección de memoria en la que se encuentra almacenado el primer elemento del vector.
c. **La sentencia `p2=lista;` genera un error en tiempo de ejecución.** Falso. Al igual que la sentencia incluida en la opción c, esta sentencia almacena en p2 la dirección de memoria la que se encuentra almacenado el primer elemento del vector.
d. **Tras ejecutar la sentencia `printf("%d, %d, %d\n", x, *p1, *p2);` se imprimen tres unos (1, 1, 1).** Cierto. Según lo indicado en la opción a, x almacena el valor 1. Asimismo y según lo dicho en las opciones b y c, tanto p1 como p2 almacenan la dirección de memoria en la que está almacenado el primer elemento del vector lista. Por tanto, tras ejecutar la sentencia indicada se visualizarán tres unos.

3. **Indique cuál de las siguientes afirmaciones respecto a un array es incorrecta:**
   a. **Los datos de los arrays deben procesarse elemento a elemento.** Cierto.
   b. **Un array completo no puede imprimirse directamente como si fuera un bloque.** Cierto. Para mostrar por pantalla los elementos de un vector hay que hacerlo elemento a elemento.
   c. **Un array no puede contener elementos que sean a su vez arrays.** Falso. Los elementos de un array pueden ser de cualquier tipo, incluido un array. Es más, una matriz de dos dimensiones puede verse como un vector cuyos elementos son a su vez vectores.
   d. **Un array puede comprarse directamente con otro si los dos contienen el mismo tipo de datos y son del mismo tamaño.** Falso. Al igual que el resto de operaciones, la comparación de un vector debe hacerse elemento a elemento.

BY NC SA

4. Indique qué afirmación en relación con el siguiente programa es correcta:

```
#include <stdio.h>
    struct persona{
           char nombre[50];
           int edad;
    };
    struct rodaje{
           char lugar[256];
           float presupuesto;
      };
    struct pelicula{
       struct persona director;
       struct persona actor1;
       struct persona actor2;
       struct rodaje datos;
    };

int main(void){

    struct pelicula mi_pelicula;
    strcpy (mi_pelicula.director.nombre, "Almodovar");
    printf ("%s \n", mi_pelicula.director.nombre);
    return 0;
}
```

a. **Se producirá un error de compilación porque la estructura persona está repetida en tres miembros de la estructura película.** Falso. La definición de la estructura película incluye tres datos (director, actor1 y actor2) diferentes que son del mismo tipo lo cual no genera ningún tipo de incompatibilidad.

b. **Se produce un error de compilación porque un miembro de una estructura no puede ser otra estructura.** Falso. No hay nada que impida que los miembros de una estructura sean de una estrucutra previamente definida.

c. **La sentencia** `strcpy (mi_pelicula.director.nombre, "Almodovar");` **genera un error en tiempo de compilación.** Falso. La sintaxis de esta sentencia es correcta.

d. **Todas las afirmaciones anteriores son falsas.** Correcto.

5. Dado el siguiente fragmento de código de un programa principal, ¿cuál de los siguientes encabezados de función es compatible con la llamada?

```
int main(void){
    float n1,n2;
    float r1[2],r2[2];
    //.................
    n1=Calcular(r1,r2,n2);
    //.................
    return 0;
}
```

a. `float Calcular (float a,float b,float c[ ])`

b. `float Calcular (float a[ ],float b[ ],float c[ ])`

c. `float Calcular (float a,float b,float c)`

d. `float Calcular (float a[ ],float b[ ],float c)`

CC BY NC SA

Solución: Los parámetros incluidos en la llamada a la función `Calcular` son un vector de tipo `float` (`r1`), otro vector de tipo `float` (`r2`) y una variable simple de tipo `float` (`n2`). Por tanto, la respuesta correcta es la opción d.

6. Indique la línea de código con la que se debe completar el siguiente programa en lenguaje C, para que la llamada a la función se haga correctamente.

```
#include <stdio.h>
struct fraccion {
      int num;
      int den;
};
void producto (struct fraccion f1, struct fraccion *resul,
struct fraccion f2);
int main(void)
{     struct fraccion a={2,5}, b={3,6}, prod;
      //En este punto se añade la línea indicada
      return 0;
}
void producto (struct fraccion f1, struct fraccion *resul,
struct fraccion f2)
{
     (*resul).num = f1.num * f2.num;
     (*resul).den = f1.den * f2.den;
       return;
}
```

a. `producto(a, &prod, b);` Cierto.

b. `producto(a, prod, b);` Falso. Se corresponde con una función en el que ninguno de los parámetros se pasa por referencia.

c. `producto(&a, prod, &b);` Falso. Se corresponde con uma función en el que el primer y tercer parámetros se pasan por referencia.

d. Con cualquiera de las opciones anteriores se produce un error de compilación. Falso. La opción a no produce ningún error de compilación.

7. Dado el siguiente programa en C, indique cuál de las siguientes afirmaciones es correcta:

```
#include <stdio.h>

int main(void){

   int a[10]={1,2,3,4,5,6,7,8,9,10};
   int i;

   for(i=1; i<=10; i++) {
      printf("%i ", a[i]);
   }
   return 0;
}
```


cc BY NC SA

a. **El programa no es correcto. Se accede a una posición de memoria que no ha sido reservada para el array.**
b. **El programa es correcto. Imprime en pantalla todos los valores del array.**
c. **El programa no es correcto. La sentencia `printf("%i ", a[i]);` debería cambiarse por `printf("%i ", &a[i]);`**
d. **El programa no es correcto. Solo se imprimen los primeros 9 valores del array.**

Solución. La primera componente del vector es `a[0]` y la última `a[9]`. Sin embargo, el índice del bucle `for` varía de 1 a 10 lo que indica que en la última iteración se intenta acceder a la componente `a[10]` que no forma parte del array. Por tanto, la opción correcta es la a.

8. **Indique cuál de las siguientes afirmaciones es cierta:**
   a. **Una estructura está compuesta por miembros que siempre son de distinto tipo.** Falso. Los miembros de una estructura pueden ser del mismo o distintos tipos.
   b. **Una estructura está compuesta por miembros que pueden ser de distinto tipo.** Cierto.
   c. **En la definición de una estructura cada miembro individual va entre llaves.** Falso. Lo que se encierra entre llaves es la declaración de todos los miembros que componen la estructura.
   d. **Para acceder a los miembros de una estructura se utiliza el operador (*).** Falso. El operador usado para acceder a los miembros de una estructura es el operador miembro (.).

9. **Dado el siguiente código en lenguaje C indique qué valor tomaría la variable X.B después de ejecutar el programa:**

```c
#include <stdio.h>

struct ejemplo
{
      int A;
      char string [10];
      float B;
};

int main (void)  {
   struct ejemplo X ={5, "string1", 3.4};
   struct ejemplo Y;
   Y.A=3;
   strcpy (Y.string, "string1");
   Y.B=5.7;
   X=Y;
   printf ("%f\n", X.B);
   system ("PAUSE");
   return 0;
}
```


cc BY NC SA

a. 3.4
b. 5.7
c. 5
d. 3

Solución. La sentencia `X=Y;` (que es una sentencia perfectamente válida) asigna a `X` los valores almacenados en `Y`. Puesto que en ese punto `Y.B=5.7`, tras ejecutar `X=Y;` el valor almacenado en `X.B` será 5.7. Por tanto, la respuesta correcta es la opción b.

10. Dado el siguiente vector de enteros, indique qué instrucción imprimirá por pantalla el número 10.

```
int vector[10]={3, 20,  4,  5, -2, 6, 7, 10, 15, 0};
```

a. `printf("%d", &( vector +7));` Falso
b. `printf("%d", *(vector+7));` Verdadero
c. `printf("%d", &vector[7]);` Falso
d. **Ninguna de las anteriores.** Falso

Solución: El valor 10 está asignado a la octava componente del vector, es decir a `vector[7]`. Por tanto, para mostrar este valor la sentencia habitual sería `printf("%d", vector[7]);` No obstante dada la relación existente entre punteros y arrays, se satisface que `vector[7]= *(vector+7)` lo que indica que las sentencias `printf("%d", vector[7]);` y `printf("%d", *(vector+7));`son totalmente equivalentes. Por tanto, la respuesta correcta es la opción b.

BY NC SA