

Fundamentos de Software de Comunicaciones

E.T.S.I. Telecomunicación

Práctica 0.1

Revisión de arrays, cadenas, punteros y estructuras en C

La práctica sirve para repasar los conceptos aprendidos sobre secuencias de bytes, cadenas de caracteres, punteros y estructuras de datos.

1. Se quiere crear una cadena de caracteres con contenido "Hola"; ¿Cuál de las declaraciones es correcta?
 - a. `char cadena[] = "Hola";`
 - b. `char cadena[5] = {'H','o','l','a','\0'};`
 - c. `char cadena [4] = "Hola";`
 - d. `char cadena[512] = "Hola\0";`
 - e. `char *cadena = "Hola";`
 - f. `char cadena = 'Hola';`
2. ¿Cómo se accede al segundo elemento de una cadena definida como `char cadena[256]`?
 - a. `cadena[] + 2`
 - b. `cadena[1]`
 - c. `*(cadena + 1)`
 - d. `*(cadena + 2)`
3. Asigne el puntero `char *p` a la primera posición de la cadena anterior. ¿cómo se hace?
 - a. `p = *cadena;`
 - b. `p = cadena[];`
 - c. `p = cadena[0];`
 - d. `p = &cadena;`
 - e. `p = cadena;`
 - f. `p = &cadena[0];`
 - g. `p = *cadena[0];`
4. Utilice la función `memcpy` para copiar 5 bytes de una cadena origen (`char origen[40]`) en otra cadena destino (`char destino[20]`). ¿cómo se hace? [NOTA: prototipo de la función `memcpy`:
`int memcpy(char *direccion_destino, char *direccion_origen, int num_bytes_a_copiar);`]
 - a. `memcpy(origen, destino, 5);`
 - b. `memcpy(destino, origen, 5);`
 - c. `memcpy(&destino, &origen, 5);`
 - d. `memcpy(&destino[0], &origen[0], 5);`
 - e. `memcpy(&destino[0], &origen[0], sizeof(char)*5);`
5. Utilice la función `memcpy` para copiar 5 bytes de una cadena origen (`char origen[40]`) en otra cadena destino (`char destino[20]`) a partir de la segunda posición de la cadena origen. ¿cómo se hace?
 - a. `memcpy(destino, origen + 1, 5);`
 - b. `memcpy(destino, origen[1], 5);`
 - c. `memcpy(destino, &origen[1], 5);`

- d. `memcpy(&destino[0], &origen[0]+1, sizeof(char)*5);`
6. Utilice la función `memcpy` para copiar 5 bytes de una cadena origen (`char origen[40]`) en otra cadena destino (`char destino[20]`) a partir de la tercera posición de la cadena destino. ¿cómo se hace?
- `memcpy(destino[2], origen, 5);`
 - `memcpy(*destino+2, origen, 5);`
 - `memcpy(&destino[2], origen, 5);`
 - `memcpy(&destino[2], &origen[0], 5);`
7. Si el prototipo de una función es: `f(char *cad);` ¿qué llamadas a la función son correctas?
- `char *cadena="Hola"; f(cadena);`
 - `char *cadena="Hola"; f(&cadena);`
 - `char *cadena="Hola"; f(&cadena[0]);`
 - `char *cadena="Hola"; f(cadena[]);`
 - `char *cadena="Hola"; f(cadena[0]);`
 - `char cadena[512]= "Hola"; f(cadena[]);`
 - `char cadena[512]= "Hola"; f(cadena);`
 - `char cadena[512]= "Hola"; f(*cadena);`
8. Se quiere declarar un array de cadenas de caracteres. ¿Qué declaraciones son correctas?
- `char * array[];`
 - `char *array[5];` //Declaración de un array de punteros a cadenas
 - `char array[][];`
 - `char **array;` //Declaración de un puntero a puntero de cadenas
 - `char array[5][512];`
9. Se quiere asignar el contenido "Hola" a una cadena previamente declarada (`char cadena[40]`) en otra parte del código. ¿Cuáles son formas correctas?
- `cadena = "Hola";`
 - `cadena = "Hola\0";`
 - `&cadena[0] = 'Hola';`
 - `&cadena = 'Hola';`
 - `cadena[0] = 'Hola';`
 - `cadena[0] = 'H'; cadena[1]='o'; cadena[2] = 'l'; cadena[3] = 'a'; cadena[4] = '\0';`
 - `cadena[0] = 'H'; cadena[1]='o'; cadena[2] = 'l'; cadena[3] = 'a'; cadena[39] = '\0';`
10. ¿Cómo se escribe contenido en la segunda cadena de un array de cadenas?
- `strcpy(array[1], "Hola");`
 - `memcpy(array[1], "Hola", 4);` //No garantiza la copia de una cadena
 - `memcpy(array[1], "Hola", 5);`
 - `strcpy(array[1][0], "Hola");`
 - `strcpy(&array[1], "Hola");`
 - `strcpy(*(array+1), "Hola");`
11. ¿Son imprimibles estos arrays como cadenas?
- `char array[4] = {'H', 'o', 'l', 'a'}; printf("%s\n", array);`
 - `char array[512] = {'H', 'o', 'l', 'a'}; printf("%s\n", array);`
 - `char array[5] = {'H', 'o', 'l', 'a', '\0'}; printf("%s\n", array);`
 - `char array[5] = {'H', 'o', 'l', 'a'}; array[4] = '\0'; printf("%s\n", array);`
12. Dada la estructura `struct A{ int a,b;};` Indique qué es correcto.
- `A x; x.a=5; x.b=7;`

- b. `A *x; x->a=5; x->b=7;`
 - c.** `A *x = new A; x->a=5; x->b=7;`
 - d. `A *x = new A; x.a=5; x.b=7;`
13. Con la definición de estructura anterior y una función de prototipo `f(struct A* s);`
¿Qué llamadas a la función son correctas?
- a. `A x; x.a=0; x.b=0; f(x);`
 - b. `A x; x.a=0; x.b=0; f(*x);`
 - c.** `A x; x.a=0; x.b=0; f(&x);`
 - d. `A x; f(x);`
14. Asigne el puntero `struct A * p` de forma correcta:
- a. `A x; p = x;`
 - b. `A x; p = *x;`
 - c.** `A x; p = &x;`
15. Se quiere copiar el contenido desde una estructura `A` a un array de caracteres (`char cadena[40]`). ¿Cómo se hace?
- a. `A x; memcpy(cadena, x, 40);`
 - b. `A x; memcpy(&cadena, &x, 40);`
 - c. `A x; memcpy(cadena, x, sizeof(A));`
 - d. `A x; memcpy(cadena, *x, 40);`
 - e. `A x; memcpy(cadena, x, sizeof(cadena));`
 - f.** `A x; memcpy(cadena, &x, sizeof(A)); //Sólo si sizeof(A) es <=40`