Universidad Simón Bolívar
Departamento de Computación
y Tecnología de la Información
Computación I
CI-2125/27 Abr-Jul 2018

Hacia la luz.—Los hombres se empujan hacia la luz, no para ver mejor, sino para brillar mejor. Se considera gustosamente como una luz aquel ante quien se brilla.
El viajero y su sombra, Federico Nietzsche.

Tercer Examen Parcial

NOMBRE CARNET NOTA

PARTE I Responda las siguientes preguntas con respuesta cortas en el espacio indicado:

1. Escriba una *definición* para cada uno de los siguientes *tipos* (**use sólo una línea**): (2 Pts.)

   a) **complex** es una estructura con dos campos doubles de nombres *re* e *im*.

   b) **date** es una estructura con tres miembros enteros de nombres dia, mes, agn.

   c) **id** es una union que puede almacenar un entero de nombre *ent* o una cadena de caracteres de a lo sumo 8 caracteres de nombre *str*.

   d) **status** es un conjunto de constantes enumeradas de nombres *done, stby, fail* con valores 0, 1, y 8 respectivamente.

2. Use la palabra reservada **typedef** para escribir una *declaración* para cada uno de las siguientes *tipos* (**use sólo una línea**): Sug.: use la primera en las siguientes. (2 Pts.)

   a) **Punto** es una estructura cuyos campos son dos punto-flotantes de nombres x, y.

   b) **Circ** es una estructura cuyos miembros son un Punto de nombre c y un punto-flotante de nombre r.

   c) **Segment** es un struct cuyos campos son los puntos p y q.

   d) **Poligonal** es un struct cuyos campos son un arreglo de nombre *vert* de a lo sumo 18 Puntos y un entero n que representa el número de puntos (vértices) que forman la linea poligonal.

3. Responda las siguientes preguntas en el espacio proporcionado: (6 Pts.)

   a) Escriba un ciclo **while** que cambie la letras minúsculas de la cadena s por mayúsculas.

   b) Defina y dé valor inicial a una variable de nombre *segm* del tipo Segmento definido en la pregunta 2*c*.

c) Escriba un ciclo **for** que copie en t la cadena s:

d) Si se invoca con dos cadenas de caracteres u y w, indique cuándo `foo(u,w)` retorna 1 y cuándo 0,

```
int foo(char s[], char t[]){
   int k;
   for(k = 0; s[k] != '\0' && t[k] != '\0'; k++) if(s[k] != t[k]) break;
   return (s[k] == t[k]);
}
```

PARTE II De cada uno de los siguientes enunciados marque el correcto. (5 Pts.)

1. En el lenguaje C,

   —una cadena de caracteres se implementa como un arreglo de caracteres que contiene un carácter nulo `\0` que marca el fin de la cadena.

   —se puede asignar directamente una cadena a otra.

   —en el tamaño de una cadena se cuenta el carácter nulo.

   —para darle el valor "foo" a la cadena s se puede usar la instrucción `s = "foo\0"`.

2. Si s y t son cadenas, la instrucción `strcat(s,t);`

   — copia al final de la cadena t la cadena s.

   — retorna en la cadena t el resultado de copiar al final de la s la t.

   — concatena la cadena t al final de la s—no altera a t y retorna s.

   — concatena la cadena t al principio de la s y retorna s.

3. En C los tipos definidos mediante la palabra reservada **struct**,

   — no pueden ser retornados por una función, pero sí pasados como parámetros.

   — para el propósito de pasaje de parámetros, retorno de valores, y asignación son exactamente igual a cualquier tipo de dato-básico.

   — si bien se pueden asignar directamente y pasar por parámetros, no se pueden retornar.

   — se pueden pasar como parámetros pero sólo por referencia.

4. En C se puede,

   — usar el operador de asignación (=) para producir una copia de un arreglo de caracteres.

   — se puede copiar un arreglo en otro usando el operador de asignación.

   — no se puede copiar un *struct* usando el operador de asignación.

   — se puede usar el operador de asignación para copiar un *struct*.

5. En C la palabra reservada **union** sirve para,

   — unir varias variables de distintos tipos con un mismo nombre.

   — almacenar en el mismo momento objetos de distinto tipo.

   — definir varias variables de distinto tipo.

   — definir un tipo de dato que permite almacenar, en la misma localidad de memoria, diferentes tipos de datos—uno a la vez.

6. En el lenguaje C se usa la palabra reservada **typedef**,
   - — para dar (declarar) un nuevo nombre a un tipo de datos.
   - — sólo en conjunto con la palabra reservada **struct** para definir nuevos tipos de datos.
   - — sólo se puede usar fuera de las funciones.
   - — sólo se puede usar dentro de una función.
7. En lenguaje C,
   - — si se define `struct pp {int a; int b;};` y se declara `struct pp m;`, es válido hacer `m = {8};`.
   - — se usa la palabra reservada **struct** para definir un tipo de datos que permita almacenar al mismo tiempo varias variables de distinto tipo bajo un mismo nombre.
   - — se usa la palabra reservada **enum** para definir varias *variables* numéricas.
   - — la definición `enum foo {done = 0, ok = 0, fail = 1}` es válida.
8. Para comparar y copiar dos cadenas de caracteres en C se usan respectivamente,
   - — **strcat** y **strcpy**.
   - — **strcmp** y **strcpy**.
   - — **strcmp** y **strtok**.
   - — **strcpy** y **strcmp**.
9. El siguientes código permite descomponer el *string* s en sus tokens usando los separadores `" ,."`,
   Nota: `usar(tok)` significa hacer algo con *tok*.
   - — `do{ tok = strtok(s, " ,."); usar(tok);} while(tok != NULL);`
   - — `char *tok = strtok(s, " ,."); while(tok != NULL){usar(tok); tok = strtok(NULL, " ,.")}`.
   - — `char *tok = strtok(s, " ,."); while(tok != NULL){usar(tok); tok = strtok(s, " ,.")}`.
   - — `char * tok; while((tok = strtok(s, " ,.")) != NULL) usar(tok);`.
10. La función `FILE *fopen(char name[], char mode[])`,
    - — si se invoca como `fopen(name, "r")` retorna NULL sólo si name no existe.
    - — si se invoca como `fopen(name, "r+")` name no tiene que existir.
    - — si se invoca como `fopen(name, "a")` abre el archivo name para escribir al final del mismo y si no existe lo crea para escribir al principio.
    - — si se invoca como `fopen(name, "w")` el archivo name debe existir.

PARTE III Se desea leer el carné (*car*), el nombre (*nom*) y las notas de los tres exámenes parciales $p1, p2, p3$ de cada uno de los estudiantes que se encuentran en un archivo de nombre "notas.txt" en un arreglo de estructuras de tipo **Est**—que se definirá más adelante— y luego usando la función cuya *declaración* es `int definitiva(Est e)` asignarle valor al campo *def* de cada estructura. Finalmente se desea crear un nuevo archivo de nombre "actas.txt" que contenga el carné, el nombre y la nota definitiva de cada estudiante.

Si el archivo de "notas.txt" luce como se muestra,

```
1410888 Rodriguez 30.5 30 35
1511668 Castro 20 16.5 30
1511806 Perez 12 13.5 20
1511666 Andrade 20 22 31.5
```

su archivo de salida "actas.txt" debería ser de la forma:

```
1410888   Rodriguez          5
1511668   Castro             3
1511806   Perez              2
1511666   Andrade            4
```

Por favor coloque aquí la definición de su estructura.

Concretamente se le pide que haga las siguientes tareas en el espacio indicado:

1. Usando las palabras reservadas **typedef** y **struct** escriba una declaración del tipo **Est** que de tener los campos *car*, *nom*, *p1*, *p2*, *p3 y def* donde los dos primeros son cadenas de a lo sumo 9 y 17 caracteres respectivamente, las siguientes tres son punto-flotantes y el útimo es entero.
   Responder en el espacio indicado de la página anterior. (4 Pts.)

2. Escriba la *definición* de una función cuya **declaración** es `void leeArchivo(Est e[], int *n)` que debe leer los datos del archivo "notas.txt" en el arreglo de estructuras e, actualizar la variable $*n$ y asignarle valor al campo *def* usando la función **definitiva**. Nota: no debe hacer la función definitiva(), sólo usarla. (7 Pts.)

3. Escriba la *definición* de una función cuya *declaración* es `void escribeArchivo(Est e[], int n)` que escribe en el archivo "actas.txt" el carné, el nombre y la nota definitiva de cada uno de los n estudiantes en e. (4 Pts.)