Lenguaje C – Práctica Complementaria - Año 2020

Estructuras y Arreglos de Estructuras

1) Declarar un tipo de estructura de nombre **alumnos** para almacenar la siguiente información:
   - **nombre del alumno** **char[20]**
   - **nº de legajo** **char[5]**
   - **domicilio** **char[15]**
   - **turno** **char**

   Declarar una variable del tipo alumnos dentro del main y mostrar todos sus campos.

2) Si declaramos fuera del main el tipo de estructura: struct alumnos {int cod; char nom [10] } alum; La variable alum tendrá un alcance……..en el programa.

3) Declarar un tipo de estructura de nombre **alumnos** para almacenar la siguiente información:
   - **nombre del alumno** **char[20]**
   - **turno** **char**

   Dentro del main declarar dos variables del tipo alumnos y asignarle valores a sus respectivos campos. Determinar si ambos alumnos pertenecen al mismo turno o no.

4) Declarar un tipo de estructura de nombre alumnos para almacenar la siguiente información:
   - **nombre del alumno** **char[20]**
   - **nº de legajo** **char[5]**
   - **domicilio** **char[15]**
   - **turno** **char**

   Luego en el main declarar un arreglo para cada uno de los 300 alumnos de un establecimiento.

   Asignar en la primera posición del arreglo de estructura, los siguientes datos: “Juan Gomez”, “123245”, “San Juan 1056”, ‘M’. Finalmente mostrar los datos cargados en la posición 0 del arreglo.

5) De los alumnos de una escuela se registra su apellido, nombre y su altura.
Como máximos hay 400 alumnos.
Diseñar un programa que indique el nombre del alumno más alto y su altura (sólo uno es el más alto).

6) Se dispone de registros con los siguientes datos: código (5 caracteres), nombre (20), domicilio (20), saldo.
   Se solicita cargar dichos datos (500 registros como máximo) y luego satisfacer un esquema de consulta interactiva donde al ingresar un código se informan los datos restantes correspondientes.

7) Se ingresa el apellido, la nota y el legajo de los 8 mejores alumnos de un curso. Realizar un programa que permita modificar los datos ingresados hasta que el usuario lo determine. Para ello

se ingresa el legajo y la nota a modificar. Si no se encuentra el legajo se deberá mostrar un mensaje sobre lo ocurrido. Finalmente mostrar los datos ordenados por legajo.

8) Un profesor de matemática de un establecimiento educativo registra de cada alumno Nº de legajo, nombre y promedio. Según el promedio desea saber cuántos alumnos aprobaron (promedio mayor o igual a 6), cuantos rinden en diciembre (promedio menor a 6 y mayor o igual a 4) y cuantos rinden examen en marzo (promedio menor a 4). Además, desea conocer el Nº de legajo y nombre del alumno con mejor promedio. El profesor tiene 200 alumnos en sus clases.

9) Cargar en un arreglo de estructura del tipo **struct dias {int num; char dia[10]};** 5 días de la semana y su posición en la semana, ej: 1 Lunes, 3 Miércoles.
Luego ingresar por teclado un día de la semana cualquiera (entre 1 y 7 -validar-) y determinar si ese día se encuentra en el arreglo de estructura de tipo **struct dias**. Mostrar ambos campos en caso de encontrarse o "no existe" en el caso de no estar dentro del arreglo.

10) Cargar en un arreglo de estructura del tipo **struct dias {int num; char dia[10]};** los 7 días de la semana. Ej: 1 Lunes, 2 Martes, 3 Miércoles, …
Luego cargar un arreglo bidimensional con el nombre de los 7 días de la semana (matriz de 7 filas).
Mostrar por pantalla si los nombres de los días en el arreglo bidimensional se encuentran y coinciden exactamente con los nombres de los días en el arreglo de estructura del tipo **struct dias**.
Verificar que existan todos los días en ambas estructuras de datos, y coincidan exactamente los nombres, no importa en qué orden se encuentren en una u otra estructura de datos.
Informar lo ocurrido.

Práctica de Funciones (pasaje de parámetros por Valor)

1) Diseñar un programa en C para calcular el cubo de los números del 1 al 5 utilizando una función definida por el usuario para realizar el cálculo del cubo. El prototipo de la función es el siguiente:
int cubo (int base);

2) Ingresar dos números enteros y luego presentar el siguiente menú de opciones:
1- SUMAR
2- RESTAR
3- MULTIPLICAR
4- DIVIDIR
A esto el usuario debe responder con la opción correspondiente a la operación que desee hacer entre los números y el programa debe devolver el resultado. Utilizar funciones para las llamadas a cada punto del menú de opciones. El programa debe ser iterativo.

3) Escribir un programa que lea el radio de un círculo y que imprima el diámetro del mismo, su circunferencia y su área. Utilice el valor constante 3.14159 para "PI" y defina al mismo como constante simbólica. Utilizar funciones para cada uno de los diferentes cálculos a realizar.

4) Escribir un programa que ingrese 10 números enteros, los cargue en un arreglo unidimensional, y a través de dos funciones busque el mayor de ellos y calcule el promedio y muestre:

LOS NÚMEROS INGRESADOS SON: .....
EL MAYOR DE ELLOS ES: .....
SU PROMEDIO ES: .....

5) Escriba una función que tome un valor entero ingresado por teclado desde el main y mostrar un mensaje en la función si el número es primo o no.

Punteros

1) Decir cuál será la salida en cada caso:

```
#include<stdio.h>
   main() {
          int arr[10] = {23, 5, 98, 65, 3, 55, 73, 9, 21, 85}, *p;
         p = arr;
          printf ( " %d\n", arr[*(p + 7)]);
          printf ( " %d\n", *arr + 3);
          printf ( " %d\n", *p++);
          printf(" %d\n", *(arr + 1));
          printf(" %d\n", (*p)++);
         printf(" %d\n", *p);
         printf(" %d\n", *p++);
         printf(" %d\n", *p);
```

2) Dado el siguiente fragmento de programa:

```
char u, v = 'A';
char * pu, * pv = &v;
……….
*pv = v + 1;
 u = *pv + 1;
 pu = &u;
```

y suponiendo que el valor asignado a **u** se almacena en la dirección **F8C** y el valor asignado a **v** se almacena en la dirección **F8D**, decir:

a) ¿Qué valor es asignado a pv?
b) ¿Qué valor es representado por *pv?
c) ¿Qué valor es asignado a u?
d) ¿Qué valor es asignado a pu?
e) ¿Qué valor es representado por *pu?

3) Sabiendo que en código ASCII la letra 'e' es el valor decimal 101, escribir que muestra cada línea de printf.

```
#include <stdio.h>

int main() {
char c;
c='e';
printf("%c",c);
c++;
printf("%d",c);
printf("%c",c);
}
```

4) ¿Qué valor muestra el último printf?

```
#include <stdio.h>
main() {
int num ,* punt , tol;
num=8;
punt= &num;
tol=*punt;
printf("el total es: %d",tol);
}
```

5) Suponiendo que x es un puntero de tipo entero y num es una variable entera, si x = &num y a = *x, ¿cuánto vale a? (marque con una cruz la respuesta correcta).

a. a vale igual a num _____

b. a contiene la dirección de memoria de num _____

c. a contiene la dirección de memoria de x _____

Ahora, para comenzar a practicar funciones con pasaje por referencia utilizando punteros:

6) Crear una función de nombre micopy que me permita copiar una cadena en otra utilizando punteros. El prototipo de la función es el siguiente: void micopy (char *, char *);

Seguimos con más ejercicios de funciones con pasaje por referencia utilizando punteros:

7) Ingresar desde el main una cadena, y a través de una función mostrar cuantos dígitos tiene la cadena y también su longitud. La función recibe dos parámetros, un puntero a char para recibir la cadena, y un puntero a entero que devuelve al main la cantidad de dígitos que tiene la cadena. La función retorna un entero con la cantidad de caracteres que tiene la cadena (su longitud).

   El prototipo de la función es el siguiente: int verificaDigitos (char *, int *);

8) Ingresar una cadena y por medio de una función determinar si dicha cadena está formada sólo por caracteres alfanuméricos. La función devolverá el valor "SI" en caso afirmativo, o el valor "NO", en caso contrario.
   El prototipo de la función es el siguiente: char * verif (char *);

9) Desarrollar un programa en lenguaje C, que realice las siguientes acciones:

   a) Cargue un texto.
   b) Determine cuál es el porcentaje de vocales sobre los caracteres del texto.
   c) Determine cuál es el porcentaje de consonantes sobre los caracteres del texto.
   d) Muestre:

      El texto……. tiene……% de vocales y ……..% de consonantes.

Para resolver esta problemática deberá utilizar las siguientes funciones:

- Una función que cargue un texto (que incluya cualquier tipo de caracteres, incluso la nueva línea '\n') cuyo prototipo será:
  void carga (char*, int*)

- Una función que solamente determine si un carácter es una vocal (no cuente cantidad de vocales), su prototipo será:
  void esVocal (char, int*)

- Una función que solamente determine si un carácter es una consonante (no cuente cantidad de consonantes), con prototipo:
  int esConsonante (char)

10) Desarrollar un programa en lenguaje C, que realice las siguientes acciones:

a) Cargue un texto.
b) Cambie los dígitos por * (asteriscos).
c) Muestre:
   El texto

   ……

   cambiado se lee

   ……..

Para resolver esta problemática deberá utilizar las siguientes funciones:

- Una función que cargue un texto (que incluya cualquier tipo de caracteres, incluso el enter '\n') cuyo prototipo será:
  char * carga (int *)
  La función devuelve el texto y deja disponible al main como parámetro la cantidad de caracteres de dicho texto.

- Una función que cambie el texto, su prototipo será:
  char * cambia_texto (char *, int)
  La función devuelve el texto cambiado (su dirección de comienzo), recibiendo como parámetros la dirección de comienzo del texto original y su cantidad de caracteres.

11) Cargar en el main un arreglo bidimensional con el nombre de los 7 días de la semana (matriz de 7 filas). Luego dentro de la función **struct dias * cargaDiasSemana();** cargar en un arreglo de estructura del tipo **struct dias {int num; char dia[10]}** los 7 días de la semana. Ej: 1 Lunes, 2 Martes, 3 Miércoles, … . Mostrar en el main por pantalla si los nombres de los días en el arreglo bidimensional se encuentran y coinciden exactamente con los nombres de los días en el arreglo de estructura del tipo **struct dias**. Verificar que existan todos los días en ambas estructuras de datos, y coincidan exactamente los nombres, no importa en qué orden se encuentren en una u otra estructura de datos. Informar lo ocurrido.

12) Cargar en el main un arreglo de estructura del tipo **struct dias {int num; char dia[10]}** los 7 días de la semana. Ej: 1 Lunes, 2 Martes, 3 Miércoles, … . Luego dentro de la función **char * comparaDias**

(struct días *); cargar un arreglo bidimensional con el nombre de los 7 días de la semana (matriz de 7 filas) y comparar el arreglo de estructura enviado como parámetro con el arreglo bidimensional creado dentro. Mostrar en el main **"todos los días son correctos"** por pantalla si los nombres de los días en el arreglo bidimensional se encuentran y coinciden exactamente con los nombres de los días en el arreglo de estructura del tipo **struct dias**, caso contrario mostrar **"algún día es incorrecto"**.