***Normas Generales***

Numere las hojas entregadas. **Lea detenidamente cada pregunta y consulte las dudas de interpretación que pudiesen surgir.** Complete en la primera hoja la cantidad total de hojas entregadas.

Realice este parcial en lápiz, o con tinta color azul o negro. ***Por favor NO utilice rojo ni verde***.

Identificar cada hoja con: Nombre, Apellido, Legajo.

***Por favor entregar esta hoja y las restantes del tema junto al examen.***

### Parte Teórica

1)

* 1. ¿Qué es una variable?. Explique tipos, tamaños en bytes.
  2. Explique comando sizeof
  3. ¿Qué es un puntero?
  4. ¿Qué es un vector?
  5. &VEC[0] es lo mismo que decir: \_\_\_\_\_\_

2) Dado la siguiente declaración de un array:

**int v[5] = {32,12,15,89,6};**

y asumiendo que el sistema operativo nos asigna la dirección **0x1cbf0000** para el comienzo del mismo:

a) ¿Qué dirección de memoria contendrá el valor de **v[2]** ?

b) ¿Qué dirección de memoria resultará de resolver **v+3** ?

c) Indique cómo accedería al valor **89** dentro del vector, por cualquier método que conozca.

d) Podemos recorrer el vector utilizando post-incremento **(v++)** ? Justifique su respuesta.

e) Recibiríamos algún error durante la compilación o el linkeo si tratáramos de acceder al contenido de **v+10**? Justifique su respuesta.

3) ¿Que diferencia existe entre un vector de char y un string?

4) ¿Que tamaño tiene una variable puntero a un int, una variable puntero a un char y una variable puntero a un long? Incluya una explicación de la razón de los tamaños e implicaciones.

5)

a) ¿Que es una estructura?

b) Describa un ejemplo de uso, y el tamaño en memoria que ocupa una variable instanciada de esa estructura de ejemplo que ha creado.

c) Describa como accede a las partes de la estructura

d) Cree un puntero a la estructura y de un ejemplo de acceso a un campo de la misma utilizando ese puntero.

### Parte Práctica

Escriba en sendos archivos fuente las siguientes funciones:

**1) int mi\_contador (char \*str, char c);**

Donde **char \*str** es un puntero a char que contiene la dirección de inicio de una cadena de caracteres, y **char c** es un carácter elegido por el usuario.

Retorna: la cantidad de veces que se encuentra **c** en **str** *int*.

**2) int mi\_concat (char \*str1, char \*str2, char \*str3);**

Donde **char \*str1** y **char \*str2** son punteros a char que contiene la dirección de inicio de una cadena de caracteres, y **char \*str3** es un puntero a char.

La función debe concatenar **\*str1** y **\*str2** dejando la cadena resultante en un string apuntado por **\*str3**

Retorna: 1 si se pudo hacer la operación, y 0 en caso contrario en un *int*.

**3)**Desarrolle un programa que recibe **por línea de comandos** dos vectores de un espacio bidimensional y calcule el producto escalar y lo imprima en stdout. Ejemplo:

**$ *producto* 1.2;2 3.01;4** *(donde A\*B será Ax\*Bx+Ay\*By)*

La validación de ambos vectores será realizada por una función llamada validar que se encuentra desarrollada en el archivo validar.c y su prototipo esta en validar.h. El prototipo de esta función es

**int validar (char \*p)**

Donde **p** es el puntero al vector a validar. La función devuelve 0 si el vector es correcto, -1 en caso de error.

El programa desarrollado estará en el archivo main.c y deberá indicar la forma de compilar y linkear el mismo.

**4)** Implemente una función que elimine de un string el carácter pasado como parámetro. Tenga en cuenta

que al eliminar caracteres deberá ajustar la posición del '\0'

El prototipo de la función es el siguiente:

**int eliminarCaracter (char \*dataPtr, char c);**

Donde **dataPtr** es el puntero al string a modificar y **c** el carácter a eliminar.

Retorna: La cantidad de caracteres eliminados.