

FINAL INFORMATICA I		<i>Duración</i>	<i>Fecha</i>	<i>Hojas</i>
<i>Nombre y Apellido</i>	<i>Nº Legajo</i>	<i>Calificación</i>		<i>Docente Evaluador</i>
		<i>número</i>	<i>letras</i>	<i>Nombre</i>

*Numere las hojas entregadas, y complete el casillero **Hojas** con la cantidad sin incluir las del tema.
Lea detenidamente el enunciado, su correcta interpretación forma parte de esta evaluación.
En la cantidad de hojas, no cuente las que corresponden al enunciado.*

Parte Teórica (30 minutos)

- 1) Para el caso de llamar a una función: Explique la diferencia entre pasarle un argumento por referencia a pasarlo por valor. ¿En qué caso utilizaría cada uno?
De un ejemplo de un prototipo de función que reciba como argumento un puntero a entero por referencia. ¿En qué caso es necesario?
- 2) Hablando de manejo avanzado de datos. ¿Cuál es la diferencia entre una estructura y una unión?
¿Qué es un campo de bits y para qué se utiliza?
- 3) ¿Qué significa que una función es recursiva? ¿En qué casos la puedo utilizar y qué condiciones debe cumplir? ¿Cuáles son las diferencias destacables con una función convencional?

FINAL INFORMATICA I		<i>Duración</i>	<i>Fecha</i>	<i>Hojas</i>
		120 minutos	11/02/2025	
<i>Nombre y Apellido</i>	<i>Nº Legajo</i>	<i>Calificación</i>		<i>Docente Evaluador</i>
		<i>número</i>	<i>letras</i>	<i>Nombre</i>

Numere las hojas entregadas, y complete el casillero **Hojas** con la cantidad sin incluir las del tema.
Lea detenidamente el enunciado, su correcta interpretación forma parte de esta evaluación.
En la cantidad de hojas, no cuente las que corresponden al enunciado.

Parte Práctica (90 minutos)

- 1) Para un nuevo desarrollo se requiere una función que reciba un vector de números enteros. La cantidad de valores se desconoce pero se sabe que finaliza cuando se encuentra un elemento de valor -1. Los números de dicho vector corresponden a las posiciones donde hay alojados UNOS en una variable del tipo unsigned int. No están necesariamente en orden ascendente o descendente. El retorno de la función es el valor del unsigned int correspondiente.

Se pide:

Realizar la función que cumpla con los requerimientos mencionados.

El prototipo de la función será:

*unsigned int ennumero (int * posiciones);*

Donde:

posiciones es el vector con las posiciones donde están los UNOS.

unsigned int es el valor resultante.

- 2) Con el objeto de poder probar la función del punto anterior, se pide:
Realizar un programa que reciba por argumento de main una serie de valores enteros (mínima cantidad 0, máxima cantidad 32) y los agregue en un vector que contenga dichos valores con el agregado del valor -1 al final.
- 3) **Se pide** realizar una función capaz de convertir el nombre de una variable en formato camello a formato serpiente. Es decir que el nombre posee letras mayúsculas en medio que deberán ser reemplazadas por un guión bajo '_' y la misma letra pero en minúscula.

No contiene espacios.

Ej. **nombreVariable** pasaría a ser **nombre_variable**

El string que contenga a la nueva variable deberá ser del tamaño exacto.

La primera letra de la variable en formato camello siempre deberá ser minúscula sino no realizará la conversión.

No puede usar las funciones cuyos prototipos se encuentren en el archivo de encabezado **string.h**

Prototipo de la función:

*char * camello2serpiente (char * nombre);*

Donde:

nombre es el vector que contiene el string a convertir.

*char ** es la posición de memoria donde se encuentra el string convertido.

FINAL INFORMATICA I	<i>Duración</i>	<i>Fecha</i>	<i>Hojas</i>
	120 minutos	11/02/2025	

Si desea probar el código de la función puede utilizar la siguiente función main:

```
int main (void)
{
    char palabra[6][20] = {"contador","Contador","contAux","contAux2","contDePasadas","");
    char * resu = NULL;
    int i;

    for (i = 0; i < 6; i++)
    {
        resu = camello2serpiente (palabra[i]);
        if (resu)
        {
            printf("El nombre %s se convirtio a %s\n", palabra[i], resu);
        }
        else
        {
            printf("El nombre %s no se pudo convertir\n", palabra[i]);
        }
    }

    return 0;
}
```

Condiciones generales:

Deberá entregar un archivo fuente main1.c con la función main del punto 2, el archivo fuente funciones1.c con la función pedida en el punto 1 y todas aquellas que considere oportuno agregar, y el archivo fuente funciones1.h correspondiente al funciones1.c

Por otro lado entregará el archivo fuente funciones2.c con la función pedida en el punto 3 y todas aquellas que considere oportuno agregar, y el archivo fuente funciones2.h correspondiente al funciones2.c

Tenga presente y respete las buenas prácticas de programación.

Considere todos los errores posibles que se pueden producir e informelos de manera apropiada según donde se produzca.

Fin del examen

Condición de aprobación:

Práctica: Debe sumar 6 puntos y resolver bien tres de los seis puntos en negrita.

Lógica del programa: 2 pts.

Manejo de funciones: 1 pto.

Manejo de errores: 1 pto.

Manejo de argumentos de main: 1 pts

Manejo de memoria dinámica: 1 pto.

Operadores a nivel de bits: 1 pto.

Estructuras de decisión/iteración: 1 pts.

Manejo de strings: 1 pto.

Aplicación de buenas prácticas de programación: 1 pto.