

Por favor siga las siguientes indicaciones:

- Escriba con lápiz.
- Escriba su nombre y número de documento en todas las hojas que entregue.
- Numere las hojas e indique el total de hojas en la primera de ellas.
- Escriba las hojas de un solo lado.
- Comience cada ejercicio en una hoja nueva.
- El total máximo de puntos del examen es 100.
- El examen contiene un total de: 2 páginas.
- La prueba es individual y sin material.
- Solo se contestan dudas acerca de la letra de los ejercicios.
- Duración 3 horas.

Ejercicio 1 (15 puntos)

1. ¿Qué sucede si pasamos un parámetro a una función por referencia y lo modificamos dentro de ella? ¿Cómo lo hacemos?
2. Describa (no implemente) el algoritmo de ordenación por selección (*Selection Sort*).

Ejercicio 2 (30 puntos)

Una palabra es una secuencia ordenada de caracteres que puede ser vacía y que nunca puede tener más de 50 caracteres. Por tanto la estructura que modela la palabra podría ser la siguiente:

```
#define largo 50

struct palabra{
    char elems[largo];
    int pos;
};
```

Se utilizará el campo pos para indicar el último carácter en la palabra. Si la palabra es vacía pos= -1.

Ejemplo: Si palabra es “Hola” entonces elems = ['H','o','l','a'] y pos=3

Se pide implementar las siguientes funciones.

1. bool esPrefijo (palabra p1, palabra p2);
Determina si la palabra p1 es prefijo de la palabra p2.
p1 será prefijo de p2 si y solo si p1 es un segmento inicial de p2.
La palabra vacía es prefijo de todas las palabras.

Ejemplo: “Buen” es prefijo de “Bueno” pero no de “Bulto”.

2. int lengthSinCar (palabra pal, char car);
Calcula la longitud de una cadena de texto, sin considerar al carácter car.
Si la cadena es vacía devuelve 0.

Ejemplo ilustrativo: lengthSinCar (“Cocina”, “c”) = 4
lengthSinCar (“Cocina”, “a”) = 5

Ejercicio 3 (15 puntos)

Implemente recursivamente la función *ultimaAparicion*, a la cual se le pasa por parámetro una palabra, el largo de la misma, y un carácter, y devuelve la posición en la palabra donde aparece por última vez el carácter pasado por parámetro. Si el carácter no aparece en la palabra, la función debe devolver -1.

Usar el siguiente cabezal para la función: ***int ultimaAparicion(char str[], int largostr, char car)***

Ejemplos: *ultimaAparicion*("Mama", 4, 'a')= 3
ultimaAparicion("Mama", 4, 'm')= 2
ultimaAparicion("", 0, 'i')= -1
ultimaAparicion("estudiar", 8, 'z')= -1

Ejercicio 4 (40 puntos)

La cadena de supermercados "COMPRES BARATOS", va a sortear 15 automóviles entre sus clientes (1 por día) y una vez finalizado cada sorteo, desea registrar los resultados del mismo. Para cada sorteo necesita registrar la fecha en que se realizó, y los siguientes datos del ganador: cédula, apellido y nombre.

Escribe un programa en C/C++ que despliegue y maneje el siguiente menú:

- 1- Alta de resultado de sorteo
- 2- Listar resultados de sorteos
- 3- Buscar resultado de sorteo
- 0- Salir

En la **opción 1** se le pedirá al usuario la fecha del sorteo y los datos del ganador, y se le dará de alta en el sistema, siempre y cuando no exista ya un sorteo registrado para esa fecha.

En la **opción 2** se deben desplegar en la pantalla todos los datos de los sorteos realizados (fecha y datos del ganador).

En la **opción 3** dada la fecha de un sorteo, se buscará en los sorteos realizados, y si ya existe un sorteo llevado a cabo en esa fecha, deberá mostrar los datos del ganador, en caso contrario se deberá mostrar "Sorteo no registrado".

En la **opción 0** el programa debe terminar.

Notas:

- La cédula se representará con un arreglo de caracteres de largo 8.
- El apellido y el nombre se representarán como arreglos de caracteres de largo 20.
- La fecha del sorteo se representará con una estructura con 3 campos de tipo entero (un campo para el día, otro para el mes, y otro para el año).
- Se pueden definir funciones y procedimientos auxiliares.