



## Segunda Práctica Calificada Secciones D, E

CC112

30/04/2024 Tiempo: 1:45 horas

Ciclo: 2024-I

### Normas:

1. No compartir respuestas/consultas con sus compañeros a través de chats, redes sociales u otros medios digitales.
2. No se permiten apuntes de clase.
3. Las soluciones serán enviadas a la plataforma y/o a la cuenta de correo del profesor.
4. Todo acto anti-ético será amonestado y registrado en el historial del estudiante.

Apellidos: \_\_\_\_\_ Nombres: \_\_\_\_\_

Código: \_\_\_\_\_ Sección: \_\_\_\_

1. [5.0 ptos.] La **sucesión de Padovan**,  $P(n)$  es una secuencia de números enteros positivos que está definida por la siguiente ecuación:

$$P(n) = \begin{cases} 1 & n = 0 \\ 1 & n = 1 \\ 1 & n = 2 \\ P(n-2) + P(n-3) & n \geq 3 \end{cases}$$

Escriba un programa usando punteros que exhiba la sucesión de Padovan para cualquier valor de  $n$  mayor que 3.

Un ejemplo de los 10 primeros términos de la sucesión de Padovan:  $P(10)=1,1,1,2,2,3,4,5,7,9$

2. [5.0 ptos.] En un juego de video los jugadores se encuentran en celdas dentro de un tablero de 4 filas y 5 columnas, la posición inicial (fila, columna) se ingresa por teclado y además se pide una cadena que contiene una frase a elección del usuario. La ruta que tomará el jugador está formada por direcciones: norte, sur, este, oeste que se toman de la cadena ingresada según el orden de los caracteres 'n', 's', 'e', 'o' respectivamente. Por ejemplo, si la cadena es "adelante, **no retrocedamos**" entonces las direcciones son: **este, norte, este, norte, oeste, este, oeste, este, oeste, sur**. El desplazamiento se realiza de manera tal que si estamos en la última columna avanzar al este nos regresa a la primera columna y si estamos en la última fila avanzar al sur nos regresa a la primera fila, es decir el desplazamiento no está limitado por las dimensiones del tablero. Las celdas contienen en forma aleatoria un valor cero o uno, en caso el jugador cae en una celda de valor uno, aumenta el puntaje ganado por el jugador y la celda pierde su valor.

Utilizando **doble puntero** escriba un programa que pida el ingreso de la fila y columna inicial, el mensaje

requerido por el juego y muestre: el tablero inicial, la ruta seguida por el jugador y el puntaje total ganado, como en el siguiente ejemplo:

```
Ingrese el mensaje: el joven de pecas usa el asno
Fila inicial: 3
Columna inicial: 3
Tablero Inicial:

0 0 1 1 0
1 1 1 1 1
0 0 0 0 0
0 0 1 0 1

Ruta: (3,3) (3,4) (3,3) (3,4) (2,4) (2,0) (2,1) (3,1) (0,1) (0,2) (1,2) (0,2) (0,1)

Total ganado :3

-----
Process exited after 11.93 seconds with return value 0
Presione una tecla para continuar . . .
```

3. [5.0 ptos.] En el mundo de la criptografía, la seguridad de la información es fundamental. Una técnica comúnmente utilizada para ocultar palabras y hacer que no sean reconocibles es el uso de "**alfagramas**". Un alfagrama es una palabra en la cual las letras originales cambian de posición en orden alfabético. Por ejemplo, el alfagrama de la palabra "**alfagrama**" es "**aaaafglmr**". Utilizando funciones, escriba un programa en C++ que permita al usuario verificar si una palabra dada es un alfagrama de otra palabra. Los usuarios deben ingresar dos palabras, y el programa debe determinar si la segunda palabra es un alfagrama de la primera.

**Ejemplo de entradas:**

Palabra 1: *alfagrama*  
Palabra 2: *aaaafglmr*

Palabra 1: *hola*  
Palabra 2: *aloh*

**Ejemplo de salidas:**

La palabra *aaaafglmr* es un alfagrama de *alfagrama*.  
La palabra *aloh* no es un alfagrama de *hola*.

4. [5.0 ptos.] Implementar un programa en C++ el cual pide por teclado un conjunto de 5 nombres y los almacena en un arreglo, dado que los nombres de personas comienzan con letras mayúsculas se debe transformar a mayúscula las primeras letras de cada uno de los nombres ingresados. Luego se muestra la lista de nombres ingresados y se pide un nombre adicional, el programa debe determinar si dicho nombre se encuentra en la lista inicial. Transformar la primera letra del nombre a buscar antes de realizar la búsqueda lineal simple.

Se recomienda usar el método ***toupper*** para transformar a mayúsculas las primeras letras iniciales y la clase ***string*** y sus funciones de manejo de cadenas como por ejemplo ***compare*** (comparar dos cadenas).

**Ejemplo 1:**

*Ingresar alumno 1: pedro*

*Ingresar alumno 2: juan*  
*Ingresar alumno 3: miguel*  
*Ingresar alumno 4: Rocio*  
*Ingresar alumno 5: Patty*

*La Lista de alumnos ingresados es:*

*Pedro*  
*Juan*  
*Miguel*  
*Rocio*  
*Patty*

*Ingrese un nombre a buscar: patty*

*El nombre se encuentra en la lista*

**Ejemplo 2:**

*Ingresar alumno 1: Miguel*  
*Ingresar alumno 2: Jose*  
*Ingresar alumno 3: erik*  
*Ingresar alumno 4: leo*  
*Ingresar alumno 5: maribel*

*La Lista de alumnos ingresados es:*

*Miguel*  
*Jose*  
*Erik*  
*Leo*  
*Maribel*

*Ingrese un nombre a buscar: Tito*

*El nombre no se encuentra en la lista*