UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Programación II
Práctica Calificada 1
Pregrado
2022-I
Profesor: Jonathan Silva
Lab 2.05

Indicaciones específicas:

- Esta evaluación contiene 10 páginas (incluyendo esta página) con 3 preguntas. El total de puntos son 20.
- El tiempo límite para la evaluación es 100 minutos.
- Cada pregunta deberá ser respondida en un solo archivo con el número de la pregunta.
  - `p1.cpp`
  - `p2.cpp`
  - `p3.cpp`
- Deberás subir estos archivos directamente a www.gradescope.com, uno en cada ejercicio.
- En los archivos .cpp evitar incluir tildes en la impresión de textos y comentarios.

Competencias:

- Para los alumnos de la carrera de Ciencia de la Computación

  Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (Evaluar)

  Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución.(Usar)

  Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (Usar)

- Para los alumnos de las carreras de Ingeniería

  Capacidad de aplicar conocimientos de matemáticas (nivel 3)

  Capacidad de aplicar conocimientos de ingeniería(nivel 2)

  Capacidad para diseñar un sistema, un componente o un proceso para satisfacer las necesidades deseadas dentro de restricciones realistas (nivel 2)

Calificación:

Tabla de puntos (sólo para uso del professor)

Question	Points	Score
1	7	
2	6	
3	7	
Total:	20	

1. (7 points) **Tiempo digital**
Tema a evaluar: Estructuras de control
La empresa**"Tiempo digital"** quiere realizar una campaña de marketing realizando un desafío de programación en sus redes sociales para sus consumidores. El objetivo del desafío es crear el logo de su marca. Usted desea participar usando una solución en el lenguaje C++ para generar el logo que se muestra a continuación:

Listing 1: Ejemplo del logo

```
-----
 \ /
  \
 / \
-----
```

Para resolver el ejercicio debe utilizar **estructuras de control selectivas e iterativas**, donde el input y output se obtiene de la siguiente forma:
Input

- Se recibe del usuario el tamaño del alto de la figura con un entero entre 4 y 50.
- La figura tiene el mismo ancho y alto.

Output

- Se imprime la primera y última línea utilizando el caracter hyphen o guion ("").
- Se imprime la línea que inicia en la parte superior izquierda hasta la inferior derecha utilizando el caracter backslash ("\").
- Se imprime la línea que inicia en la parte superior derecho hasta la inferior izquierda utilizando el caracter slash ("/").

IMPORTANTE: En este ejercicio no se permite el uso de bibliotecas. Para generar el caracter backslash debe utilizar el caracter dos veces ("\\").

A continuación se muestra algunos ejemplos de la ejecución correcta del código:

Listing 2: Ejemplo de resultado 1

```
Ingrese el alto:7

-------
 \   /
  \ /
   \
  / \
 /   \
-------
```

Listing 3: Ejemplo de resultado 2

```
Ingrese el alto:1
Ingrese el alto:2
Ingrese el alto:12

------------
 \          /
  \        /
   \      /
    \    /
     \  /
      \/
      /\
     /  \
    /    \
   /      \
  /        \
 /          \
------------
```

Listing 4: Ejemplo de resultado 3

```
Ingrese el alto:200
Ingrese el alto:20

--------------------
 \                  /
  \                /
   \              /
    \            /
     \          /
      \        /
       \      /
        \    /
         \  /
          \/
          /\
         /  \
        /    \
       /      \
      /        \
     /          \
    /            \
   /              \
  /                \
 /                  \
--------------------
```

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Ejecución	El diseño del algoritmo es ordenado y claro, siguiendo buenas prácticas en programación. La ejecución es correcta (`3pts`)	El diseño del algoritmo es ordenado y claro. La ejecución es correcta (`2pts`)	La ejecución es correcta (`1pts`).	La ejecución no es correcta (`0.5pts`)
Sintaxis	No existen errores sintácticos o de compilación (`2pts`)	Existen algunos errores sintácticos de menor relevancia, que no afectan el resultado (`1.5pts`).	Existen errores sintácticos en la forma de ejecución, que no afectan el resultado (`1pts`).	El código tiene errores de sintáxis que afectan el resultado (`0.5pts`).
Optimizacion	El código es óptimo y eficiente. De buen performance e interacción con el usuario (`2pts`)	El codigo es de buen performance durante la ejecución (`1.5pts`).	El codigo no está optimizado pero la ejecución no es deficiente(`1pts`).	El codigo no está optimizado y la ejecución es deficiente (`0pts`).

2. (6 points) **Suma de números Capicúas**
Tema a evaluar: Funciones (nivel básico) y recursividad
Crear un programa que realice la suma de **números Capicúas**. El programa debe solicitar un número repetidamente hasta que el usuario ingrese el valor cero (0). Luego debe realizar la suma de los número capicúas y finalmente debe mostrar los números que son capicuas, la suma de los mismos, y si la suma es capicua o no. En caso el usuario ingrese un número que no es capicua, el programa debe reportarlo.
Debe implementar al menos 2 funciones realizando el pase por valor de las variables:

   1. Invertir el número de forma recursiva.
   2. Imprimir los números que son capicuas, la suma de los mismos, y si la suma es capicua o no.

   Además, para invertir el número debe implementar una función recursiva, de la siguiente forma:

   - Sea B la cantidad de digitos del número A

$$INVERTIR(A,B) = \begin{cases} A, & \text{si } A < 10 \\ (A\%10)*10^{B-1} + \text{INVERTIR}(\frac{A}{10}\text{,B-1}), & \text{si } A \geq 10 \end{cases} \quad (1)$$

   IMPORTANTE: En este ejercicio solo se permite el uso de las bibliotecas String y cmath.
   A continuación se muestra algunos ejemplos de la ejecución correcta del código:

Listing 5: Ejemplo de resultado 1

```
----
NUMEROS CAPICUAS
----
INGRESE NUMERO:1111
----
NUMEROS CAPICUAS
----
INGRESE NUMERO:2
 ----
NUMEROS CAPICUAS
----
INGRESE NUMERO:1313
El numero 1313 no es capicua.
----
NUMEROS CAPICUAS
----
INGRESE NUMERO:0
La suma de los numeros capicuas 1111,2, es 1113, y el numero
    no es capicua.
```

Listing 6: Ejemplo de resultado 2

```
----
NUMEROS CAPICUAS
----
INGRESE NUMERO:1221
 ----
NUMEROS CAPICUAS
----
INGRESE NUMERO:2332
----
NUMEROS CAPICUAS
----
INGRESE NUMERO:0
 La suma de los numeros capicuas 1221,2332, es 3553, y si es
     capicua.
```

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Ejecución	El diseño del algoritmo es ordenado y claro, siguiendo buenas prácticas en programación. La ejecución es correcta (`2pts`)	El diseño del algoritmo es ordenado y claro. La ejecución es correcta (`1.5pts`)	La ejecución es correcta (`1pts`).	La ejecución no es correcta (`0.5pts`)
Sintaxis	No existen errores sintácticos o de compilación (`2pts`)	Existen algunos errores sintácticos de menor relevancia, que no afectan el resultado (`1.5pts`).	Existen errores sintácticos en la forma de ejecución, que no afectan el resultado (`1pts`).	El código tiene errores de sintáxis que afectan el resultado (`0.5pts`).
Optimizacion	El código es óptimo y eficiente. De buen performance e interacción con el usuario (`2pts`)	El codigo es de buen performance durante la ejecución (`1.5pts`).	El codigo no está optimizado pero la ejecución no es deficiente(`1pts`).	El codigo no está optimizado y la ejecución es deficiente (`0pts`).

3. (7 points) **Plan de contingencia en Unidad de Hospitalización**
Tema a evaluar: Funciones (nivel intermedio) y punteros
Usted se encuentra trabajando en el área de Tecnologías de Información de un Hospital y han experimentado un ataque de ransomware que ha comprometido a los computadores y servidores con Windows.
Debido a que no se tiene certeza cuando se van a reestablecer los sistemas, se decide activar el **plan de contingencia**. Las enfermeras en la **Unidad de Hospitalización** se están quejando por cálculos incorrectos que es realizado de forma manual para el goteo de las soluciones que se tienen que aplicar a los pacientes. Por tal motivo su equipo decide implementar un pequeño programa en Linux para ayudar a los usuarios. Se le solicita crear un programa que realice el cálculo con la siguiente fórmula:

- Para el caso de gotas:

$$GOTEO\_DE\_SOLUCIONES(gotas/min) = \frac{volumen\_solucion(mL) * 20}{(horas * 60)} \quad (2)$$

- Para el caso de microgotas:

$$GOTEO\_DE\_SOLUCIONES(gotas/min) = \frac{volumen\_solucion(mL) * 60}{(horas * 60)} \quad (3)$$

Además, le solicita que preserve la seguridad de los datos de los pacientes creando una opción para eliminar los datos.
El programa debe iniciar con un menú para que el usuario realice las siguientes opciones:

1. Registrar datos de paciente: Solicitar los siguientes datos:
   - ID de paciente: String.
   - Sexo: Char.
   - Edad: Integer.
   - Volumen de solución: Float.
   - Horas: Integer.
2. Realizar cálculo del goteo de soluciones: El programa debe realizar el cálculo y mostrarlo en pantalla, en caso no existan datos debe reportar que no puede realizar el cálculo.
3. Eliminar datos de paciente: Para eliminar los datos debe colocar los valores númericos en 0 y los valores de texto en vacío.
4. Finalizar programa.

Para realizar el programa se le solicita realizar las opciones 1, 2 y 3 en funciones que reciban punteros de los datos del paciente.
IMPORTANTE: En este ejercicio solo se permite el uso de la biblioteca String.

A continuación se muestra algunos ejemplos de la ejecución correcta del código:

Listing 7: Ejemplo de resultado 1

```
----
MENU
----
1. REGISTRAR DATOS DE PACIENTE
2. MOSTRAR CALCULO DE GOTEO
3. ELIMINAR DATOS DE PACIENTE
4. FINALIZAR PROGRAMA

Ingrese opcion (1-4):1
1. REGISTRAR ID DE PACIENTE:P001078
2. REGISTRAR SEXO DEL PACIENTE (Hombre: H o h / Mujer: M o m
   ):h
3. REGISTRAR EDAD:45
4. REGISTRAR VOLUMEN DE SUERO (mL):1000
5. REGISTRAR HORAS A ADMINISTRAR:8
----
MENU
----
1. REGISTRAR DATOS DE PACIENTE
2. MOSTRAR CALCULO DE GOTEO
3. ELIMINAR DATOS DE PACIENTE
4. FINALIZAR PROGRAMA

Ingrese opcion (1-4):2
EL CALCULO DE GOTEO PARA EL PACIENTE P001078, DE EDAD 45, Y
   DE SEXO HOMBRE ES: 41.6667(gotas/min) o 125(microgotas/
   min).
----
MENU
----
1. REGISTRAR DATOS DE PACIENTE
2. MOSTRAR CALCULO DE GOTEO
3. ELIMINAR DATOS DE PACIENTE
4. FINALIZAR PROGRAMA

Ingrese opcion (1-4):3
LOS DATOS DEL PACIENTE FUERON ELIMINADOS.
----
MENU
----
1. REGISTRAR DATOS DE PACIENTE
2. MOSTRAR CALCULO DE GOTEO
3. ELIMINAR DATOS DE PACIENTE
4. FINALIZAR PROGRAMA

Ingrese opcion (1-4):4
```

Listing 8: Ejemplo de resultado 2

```
----
MENU
----
1. REGISTRAR DATOS DE PACIENTE
2. MOSTRAR CALCULO DE GOTEO
3. ELIMINAR DATOS DE PACIENTE
4. FINALIZAR PROGRAMA

Ingrese opcion (1-4):2
PRIMERO DEBE REGISTRAR LOS DATOS DEL PACIENTE.
```

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Ejecución	El diseño del algoritmo es ordenado y claro, siguiendo buenas prácticas en programación. La ejecución es correcta (`3pts`)	El diseño del algoritmo es ordenado y claro. La ejecución es correcta (`2pts`)	La ejecución es correcta (`1pts`).	La ejecución no es correcta (`0.5pts`)
Sintaxis	No existen errores sintácticos o de compilación (`2pts`)	Existen algunos errores sintácticos de menor relevancia, que no afectan el resultado (`1.5pts`).	Existen errores sintácticos en la forma de ejecución, que no afectan el resultado (`1pts`).	El código tiene errores de sintáxis que afectan el resultado (`0.5pts`).
Optimizacion	El código es óptimo y eficiente. De buen performance e interacción con el usuario (`2pts`)	El codigo es de buen performance durante la ejecución (`1.5pts`).	El codigo no está optimizado pero la ejecución no es deficiente(`1pts`).	El codigo no está optimizado y la ejecución es deficiente (`0pts`).