

Clase #07 de 27

Examen #0 & Trabajo #2

May 28, Miércoles

Agenda para esta clase

- Examen #0
 - Evaluación
 - Resolución
- Presentación Trabajo #2

Examen #0

Examen #0

1. Evaluación individual
2. Resolución cruzada
3. Resolución entre todos.

Trabajo #2

Funciones y Comparación de Valores en Punto Flotante —
Celsius y Fahrenheit

7.1. Objetivos

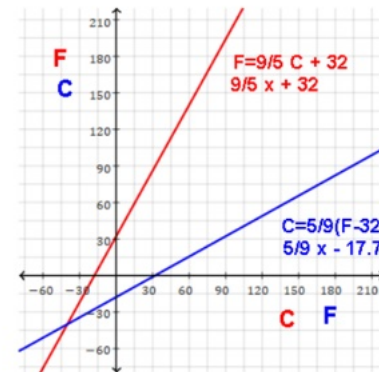
- Demostrar el manejo de funciones y valores punto flotante.

7.2. Temas

- Funciones.
- Tipo `double`.
- División entera y flotante.
- Pruebas con `assert`.
- Argumentos con valor por defecto.
- Introducción a iteraciones con `for`.

7.3. Problema

Se necesita una tabla que presente las temperaturas Celsius convertidas en Fahrenheit, y otra en el sentido opuesto. Use la siguiente imagen, [tomada de este artículo¹](#), como referencia gráfica de las relaciones:



7.3.1. Desafíos con respecto al formato de la tabla

- La columna izquierda muestra temperaturas con valores enteros (i.e, sin décimas de grados).
- La derecha debe tener precisión de una décima.
- Las dos columnas deben estar alineadas a derecha.

7.4. Restricciones

- Se deben construir dos funciones, una para cada conversión que se debe realizar, ambas son $\mathbb{R} \rightarrow \mathbb{R}$
- Las pruebas deben realizarse con `assert`.

7.4. Restricciones

- Se deben construir dos funciones, una para cada conversión que se debe realizar, ambas son $\mathbb{R} \rightarrow \mathbb{R}$
- Las pruebas deben realizarse con `assert`.

7.5. Análisis

Hay dos sub-problemas que se requieren solucionar antes de poder implementar y probar las funciones.

1. Para implementar la función *Celsius* el problema del valor de la fracción $\frac{5}{9}$ versus la división entera de la expresión `5/9` en C++; y para la *Fahrenheit*, $\frac{9}{5}$ versus `9/5`.
2. Para poder probar la implementación, el problema de la presentación no precisa de los tipos flotantes.

Una solución al primer problema es realizar división entre flotantes.

Para el segundo problema, debemos incorporar la comparación con *tolerancia*, para eso debemos diseñar una función `bool` que reciba dos flotantes a comparar y un flotante que represente la tolerancia y retorna verdadero si los flotantes están cerca según la tolerancia. Se propone la función *AreNear*, y también las soluciones presentadas en [\[DAWSON2012\]](#) y [\[ERICSON2008\]](#).

7.6. Tareas

1. Función *AreNear*:

- a. Escribir el léxico, es decir, la definición matemática de la función.
- b. Escribir las pruebas en `main`.
- c. Escribir el prototipo antes de `main`.
- d. Escribir la definición después de `main`.
- e. Compilar, ejecutar, y evaluar la salida resultante.

2. Función *Celsius* que calcula Celsius a partir de Fahrenheit: mismos pasos.

3. Función *Fahrenheit* que calcula Fahrenheit a partir de Celsius: mismos pasos.

4. (Opcional) Escribir en `main` las iteraciones que imprimen las tablas, también pueden ser dos funciones `void`.

5. (Opcional) Desarrollar funciones que grafiquen ambas funciones lineales.

Términos de la clase #07

Definir cada término con la bibliografía

- Introudcción a sentencia for

Tareas para la próxima clase

1. Empezar trabajo “Funciones y Comparación de Valores en Punto Flotante — Celsius y Fahrenheit”



¿Consultas?

Fin de la clase