

Trabajo Práctico N°1 “Sentencias de asignación y sentencias selectivas”

Universidad Tecnológica Nacional FRBA

Materia: Algoritmos y Estructura de Datos

Curso: K1031

Alumno: Axel Leonel Guiñazú

Legajo: 175.568-7

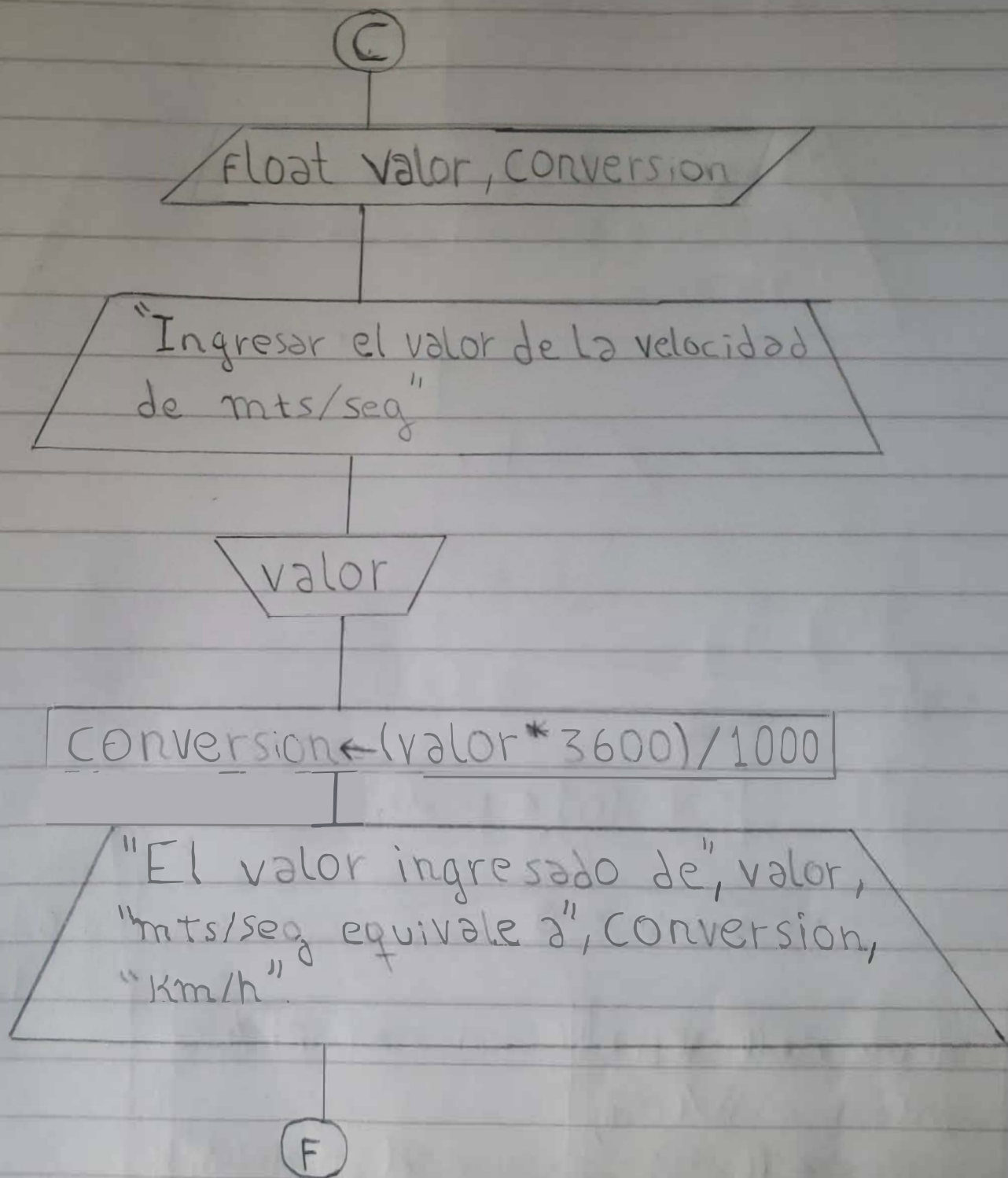
Correo: aguinazu@frba.utn.edu.ar

Usuario GitHub: AxelLeonelGuinazu

Repositorio:

<https://github.com/AxelLeonelGuinazu/AlgoritmosK10312021>

3.a-



3.b_

(C)

```
float a,b,c, calculo1, calculox1;
float calculox2, calculoI;
```

"Ingrese los valores (en formato entero o decimal) de las siguientes variables"

"a,b,c"

a,b,c

$a \neq 0$

$\text{calculo1} \leftarrow \text{pow}(b,2) - (4*a*c)$

"El valor de 'a' debe ser distinto de 0. La función es lineal"

$\text{calculo1} \geq 0$

$\text{calculox1} \leftarrow ((-1)*b) + \sqrt{\text{calculo1}} / (2*a);$
 $\text{calculox2} \leftarrow ((-1)*b) - \sqrt{\text{calculo1}} / (2*a);$

$\text{calculox1} \leftarrow ((-1)*b) / (2*a);$
 $\text{calculoI} \leftarrow \sqrt{\text{calculo1} * (-1)} / (2*a);$

$\text{calculox1} == \text{calculox2}$

"Un resultado real encontrado, calculox1;

"Dos resultados reales encontrados, calculox1, " " calculox2;

"No tiene raíces reales. Resultado imaginario encontrado, calculox1, "+, calculoI, "i", " ", calculox1, "-", calculoI, "i";

(F)

Informe de análisis del problema 3.A

Consigna: Realizar un programa que lea una velocidad en mts/seg y la transforme a kmts/hr. El programa debe solicitar por pantalla un valor numérico, luego el usuario debe ingresar un valor, finalmente el programa debe mostrar por pantalla el resultado de la conversión.

Informe:

Comienzo declarando 2 variables, una almacenará el valor ingresado por el usuario, y la otra almacenará la conversión solicitada.

Se solicita al usuario la velocidad en mts/seg, luego a la variable “conversión” le vamos a asignar el resultado del producto entre el valor ingresado y 3600 segundos correspondientes a 1 hora, dividido 1000 metros correspondientes a 1 kilómetro:

$$\text{conversión} = (\text{valor} * 3600) / 1000.$$

Si el resultado es negativo, implica una velocidad de frenado o rozamiento.

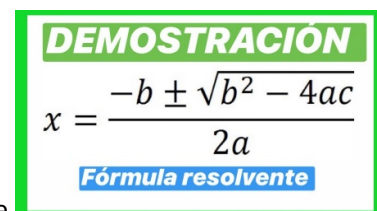
Finalmente se muestra el resultado de la conversión al usuario.

Informe de análisis del problema 3.B

Consigna: Confeccionar un programa que calcule la solución de una ecuación

cuadrática: $y = ax^2 + bx + c$

Informe:


$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Fórmula resolvente

El problema fue resuelto en base a la formula de la resolvente

Se declaran variables en la parte superior del programa a medida de que vaya apareciendo su necesidad.

Se solicita al usuario que ingrese los valores correspondientes a las variables a, b, c de su función cuadrática. Asumimos que los datos se ingresan en formato entero o decimal.

Verificamos que la variable ‘a’ sea distinta de 0 ya que en caso de ser igual a 0 no realizaremos ningún calculo, implicando una función lineal.

Si la variable a es distinta de 0, procedemos a calcular el valor resultante dentro de la raíz cuadrada. Consultamos si dicho valor (asignado a la variable ‘calculo1’) es mayor o igual a 0,.

- Si la condición no se cumple: Implicaría una raíz negativa, por lo tanto se procede a calcular el resultado imaginario de la cuadrática.

Para calcularlo separamos de la siguiente forma para asignarlo a variables

$$\frac{-b}{2.a} \pm \frac{\sqrt{b - 4.a.c}}{2.a}$$

distintas: como el valor dentro de la raíz ya lo tenemos calculado y sabemos que dará negativo, simplemente lo multiplicamos por (-1) representando el '-1' la 'i' de imaginario que luego le

$$\frac{-b}{2.a} \pm \frac{\sqrt{\text{calculo1}*(-1)}}{2.a}$$

mostraremos al usuario. Para separarlos lo hice de la siguiente manera:

$$\text{calculox1} = \frac{-b}{2.a}$$

$$\text{calculol} = \frac{\sqrt{\text{calculo1}*(-1)}}{2.a}$$

Luego simplemente le mostramos al usuario: "Resultado imaginario encontrado: " calculox1 "+" calculol "i" o " calculox1 "-" calculol "i".

Como se puede apreciar se muestra el mismo resultado dos veces pero diferenciándose por el operando de suma y resta, además se le agrega 'i' simbolizando el resultado imaginario.

- Si la condición ($\text{calculo1} \geq 0$) se cumple: calculamos las dos raíces reales aplicando la resolvente. En un caso con el operador de suma y en el otro con el operador de resta. Las variables utilizadas para almacenar los valores son: calculox1 y calculox2. Preguntamos si el resultado de ambas variables es el mismo para saber si existe 1 o 2 raíces. Luego dependiendo la veracidad de la expresión, se le informa el resultado al usuario.

4.

Declaración de variables:

- **JavaScript al igual que C++.**
 - Es *case-sensitive* (distingue entre mayúsculas y minúsculas).
 - Primero se indica el tipo de dato y luego el identificador o nombre de la variable.
 - No es necesario realizar la asignación al momento de crear la variable.
 - Declarar una variable fuera de cualquier función, se denomina **variable global**, y declarar una variable dentro de alguna función se denomina **variable local** por solo estar disponible dentro de la función.
- **JavaScript a diferencia de C++.**
 - No es necesario un punto y coma después de una declaración si está escrita en su propia línea. (Si se desea más de una declaración en una línea, entonces se debe separar con punto y coma).
 - Tiene 3 tipos de declaraciones de variables: **var, let, const**.
 - El identificador debe comenzar con una letra, o un guión bajo, o un signo de dólar, los siguientes caracteres también pueden ser dígitos(0-9).
 - Puede utilizar la mayoría de las letras ISO 8859-1 o *Unicode* en los identificadores.
 - Se puede asignar un valor a una variable no declarada (Ej: x=24). Esta asignación crea una **variable global no declarada**. También genera una advertencia estricta de JavaScript como recordatorio.

Sentencia *if*:

- **JavaScript al igual que C++.**
 - Es una estructura de control utilizada para tomar decisiones de forma condicional. Sirve para realizar unas u otras operaciones en función de una expresión. Ej: if(expresión) { }
 - La expresión a evaluar se coloca siempre entre paréntesis y esta compuesta por variables que se combinan entre si mediante operadores condicionales.
 - Primero se evalúa la expresión, si da resultado positivo se realizan las acciones relacionadas con el caso positivo.
 - Las llaves { } engloban las acciones que se quieren realizar en caso de que se cumplan o no las expresiones.
 - En el caso de que sólo haya una instrucción como acciones a realizar, no es necesario colocar las llaves, pero sí el punto y coma al final de la instrucción.
 - Para señalar las instrucciones a ejecutar en caso negativo de la condición de la expresión, se utiliza **else**. Ej: if(a=b){ "entra aquí"} else { "entra aquí"}
 - La sentencia **if ternario** se comporta igual que en C++. Ej: (condición)? "caso positivo" : "caso negativo";
- **JavaScript a diferencia de C++.**

En mi investigación no encontré diferencias entre ambos lenguajes respecto a la sentencia *if*.

