



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

UNIVERSIDAD TECNOLÓGICA NACIONAL

Facultad Regional Buenos Aires

Algoritmos y estructura de datos

–2021–

DOCENTE: ING. PABLO DAMIÁN MÉNDEZ

TRABAJO PRÁCTICO N° 1

« *SEGOVIA 43013513* »

Localización	[CAMPUS / MEDRANO]	Curso: K1025												
Integrantes del equipo														
Legajo	Apellido y Nombre													
175.422-1	SEGOVIA, MARTIN													
Correo FRBA	marsegovia@frba.utn.edu.ar													
Usuario de GitHub y link	marsegovia			https://github.com/marsegovia/RepositorioPrivado1#tp1-ayed										
Entrega / Revisión	1				2				3					
Fecha de entrega	5/5													
Fecha de calificación														
Calificación	A	B	C	D	A	B	C	D	A	B	C	D		
Firma del Docente														
OBSERVACIONES:														

INFORME TRABAJO PRACTICO N° 1

Para el ejercicio a) que pide ingresar al usuario una velocidad, se planteo lo siguiente: Primero que nada, como se menciono anteriormente, solicitamos al usuario que ingrese un valor para indicar los kilómetros por hora. Una vez que se han ingresado los valores requeridos se muestran por pantalla en formato: "Km/h". De esta manera el usuario puede verificar que los valores ingresados son correctos.

Finalizado este proceso se le es mostrado por pantalla el resultado de la conversión al usuario, pasando los kilómetros a metros (para esto solo recurrí a multiplicar el valor ingresado por mil, ya que cada 1 km hay 1000m) y también convirtiendo las horas a segundos (se multiplico el valor ingresado por el equivalente de una hora en segundos, es decir, $1H=60min=3600s$, en una hora hay 3600 segundos) e inmediatamente se dividió los metros por los segundos; quedando expresado por pantalla en "m/s".

Para el ejercicio b) el cual se requiere calcular la intersección entre dos rectas, se me ocurrió el siguiente planteo:

Se le pide al usuario que ingrese cuatro valores, estos valores son m_1 , b_1 , m_2 , b_2 . Cada valor hace referencia a las pendientes y ordenadas de las rectas.

Una vez ingresado todos los valores se muestra por pantalla como quedarían ambas rectas con los valores propuestos por el usuario, quedando así las rectas de forma:

$F(x) = m_1x + b_1$ y $G(x) = m_2x + b_2$. Inmediatamente se muestran los distintos resultados que podrían ocurrir, dependiendo de los valores que se hayan ingresado. Todo esto utilizando sentencias IF.

- 1) El programa mostraría por pantalla: "No hay intersección, son paralelas" Si las pendientes son iguales (esto indica que las rectas son paralelas) y si al mismo tiempo los valores de las ordenadas son diferentes entre sí.
- 2) El programa mostraría por pantalla: "Son idénticas" es decir que si las pendientes y las ordenadas de ambas rectas son iguales, esto indicaría que las rectas son exactamente la misma función.
- 3) Por último el programa mostraría, para el caso de que las pendientes de ambas rectas son distintas entre sí, los valores de x y de y para calcular la intersección.
Para calcular el valor de x se utilizó la ecuación de igualación de rectas: $-(b_2 - b_1) / (m_2 - m_1)$.

Una vez obtenido el valor de x , como se haría en una hoja común, reemplazamos x en una de las ecuaciones. Para poder realizar esta parte del ejercicio utilice la misma

ecuación para obtener x pero esta vez multiplicada por la pendiente y luego sumarle la ordenada: $m1 * (-(b1-b2)/(m2-m1)) + b1$.

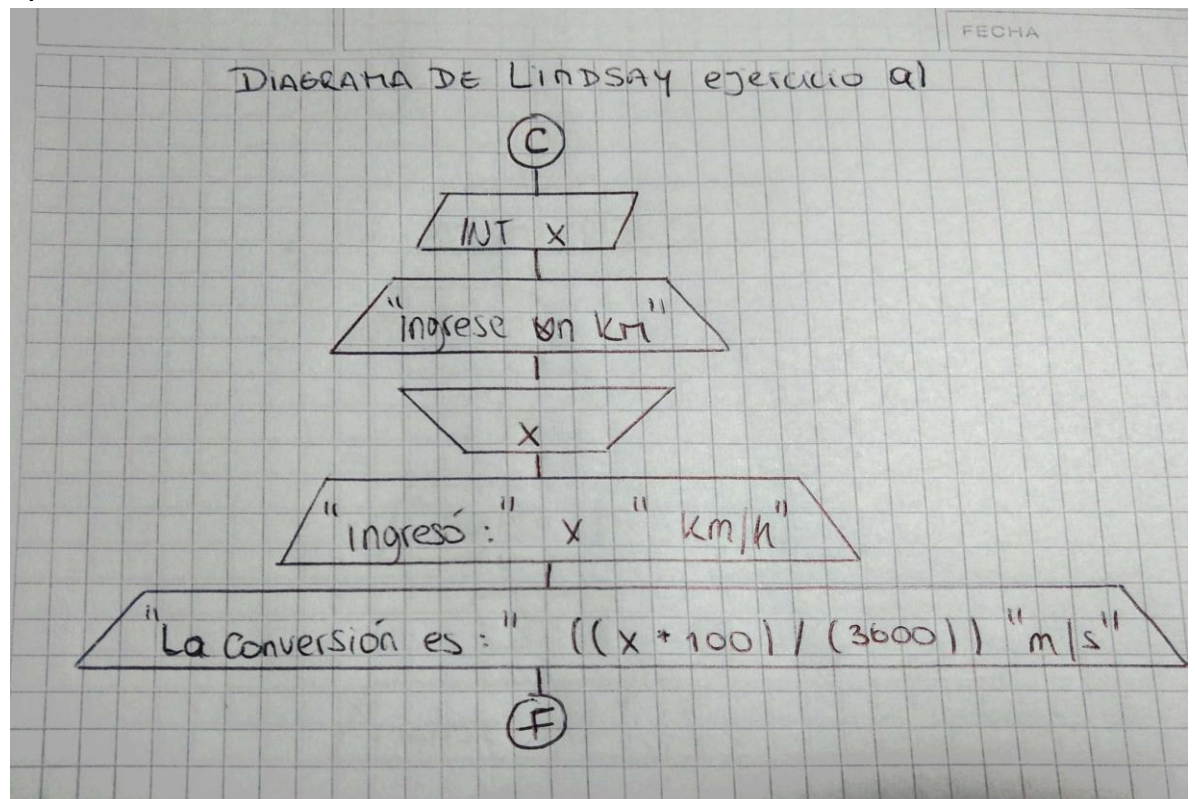
Al finalizar este proceso, inmediatamente se muestra el punto de intersección, de este modo quedaría expresado en formato de coordenada (x;y).

En este paso se utilizaron las mismas ecuaciones enunciadas anteriormente.

Esto sucede así ya que no se encontró una manera de obtener x (en código), este es un valor incógnita, por lo tanto es la forma más sencilla aunque más larga de obtener la representación del punto de coordenada x e y.

DIAGRAMAS DE LINDSAY

A)



B)

DIAGRAMA ejercicio b)

C

int m1, b1, m2, b2

Ingrese m1, b1, m2, b2

m1, b1, m2, b2

"f(x) = " m1 "x + " b1

"g(x) = " m2 "x + " b2

 $((m1 == m2) \ \&\& \ (b1 \neq b2))$

no hay intersección; son paralelas

 $(m1 \neq m2)$ "El valor de x = " $-(b2 - b1) / (m2 - m1)$ "El valor de y = " $m1 * (-(b2 - b1) / (m2 - m1)) + b1$ "La intersección (x,y) = (" $-(b2 - b1) / (m2 - m1)$ "; " $m1 * (-(b2 - b1) / (m2 - m1)) + b1$ ") " $((m1 == m2) \ \&\& \ (b1 == b2))$

"Son idénticas"

F

4) JAVA SCRIPT

La sentencia IF:

IF es una estructura de control utilizada para tomar decisiones. Es un condicional que sirve para realizar unas u otras operaciones en función de una expresión.

Funciona de la siguiente manera, primero se evalúa una expresión, si da resultado positivo se realizan las acciones relacionadas con el caso positivo.

```
if (expresión) {  
    //acciones a realizar en caso positivo  
    //...  
}
```

Se pueden indicar acciones a realizar en caso de que la evaluación de la sentencia devuelva resultados negativos.

```
if (expresión) {  
    //acciones a realizar en caso positivo  
    //...  
} else {  
    //acciones a realizar en caso negativo  
    //...  
}
```

DECLARACIONES DE SENTENCIAS:

JAVA Y C++:

if...else

Ejecuta una sentencia si una condición especificada es true. Si la condición es false, otra sentencia puede ser ejecutada.

Switch

Evalua una expresión, igualando el valor de la expresión a una clausula case y ejecuta las sentencias asociadas con dicho case

Do...while

Crea un bucle que ejecuta una instrucción especificada hasta que la condición de prueba se evalúa como falsa. La condición se evalúa después de ejecutar la instrucción, lo que da como resultado que la instrucción especificada se ejecute al menos una vez.

for

Crea un ciclo que consiste en 3 opciones diferentes, encerradas entre paréntesis y separadas por punto y coma, seguidas por un instrucción ejecutada en ciclo.

while

Crea un bucle que ejecuta la instrucción especificada siempre que la condición de prueba se evalúe como verdadera. La condición se evalúa antes de ejecutar la instrucción.

SIMILITUDES:

- Las llaves engloban las acciones que queremos realizar en caso de que se cumplan o no las expresiones.
- La expresión a evaluar se coloca siempre entre paréntesis y está compuesta por variables que se combinan entre sí mediante operadores condicionales.
- Las expresiones condicionales se pueden combinar con las expresiones lógicas para crear expresiones más complejas.
- Para hacer estructuras condicionales más complejas podemos anidar sentencias IF, es decir, colocar estructuras IF dentro de otras estructuras IF. Podemos evaluar y realizar una acción u otra según dos posibilidades.

DIFERENCIAS:

C ++	Java
<pre>if (exp_entera) { Instrucciones; }</pre>	<pre>if (exp_booleana) { instrucciones; }</pre>
<pre>if (exp_entera) { instrucciones; } else { instrucciones; }</pre>	<pre>if (exp_booleana) { instrucciones; } else { instrucciones; }</pre>

- En C, un resultado 0 en la exp_entera equivale a False
- En C, un resultado distinto de 0 equivale a True