



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: GARCIA MORALES KARINA

Asignatura: FUNDAMENTOS DE PROGRAMACIÓN

Grupo: 1121

No de Práctica(s): PRÁCTICA 3

Integrante(s): CADENA MARTÍNEZ CARLOS DAVID

*No. de Equipo de
cómputo empleado*

Semestre: 2019-1

Fecha de entrega: 5/Sep/2018

Obervaciones:

CALIFICACIÓN: _____

PRACTICA 3: SOLUCIÓN DE PROBLEMAS Y ALGORITMOS

Objetivo: Elaborar algoritmos correctos y eficientes en la solución de problemas siguiendo las etapas de Análisis y Diseño pertenecientes al Ciclo de vida del software.

Actividades:

A partir del enunciado de un problema, identificar el conjunto de entrada y el conjunto de salida.

Elaborar un algoritmo que resuelva un problema determinado (dado por el profesor), identificando los módulos de entrada, de procesamiento y de salida.

Introducción

Un problema informático se puede definir como el conjunto de instancias al cual corresponde un conjunto de soluciones, junto con una relación que asocia para cada instancia del problema un subconjunto de soluciones (posiblemente vacío). Para poder solucionar un problema nos apoyamos en la Ingeniería de Software que de acuerdo a la IEEE se define como "La aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software". Por lo que el uso y establecimiento de principios de ingeniería sólidos, son básicos para obtener un software que sea económicamente fiable y funcione eficientemente. La Ingeniería de Software provee métodos que indican cómo generar software. Estos métodos abarcan una amplia gama de tareas.

CICLO DE VIDA DEL SOFTWARE:



CARACTERÍSTICAS DE LOS ALGORITMOS

Las características fundamentales que debe cumplir todo algoritmo son:

- Un algoritmo debe ser preciso e indicar el orden de realización de cada paso.
- Un algoritmo debe estar definido. Si se sigue un algoritmo dos veces, se debe obtener el mismo resultado cada vez.
- Un algoritmo debe ser finito. el algoritmo se debe terminar en algún momento; o sea, debe tener un número finito de pasos.
- Un algoritmo debe ser legibles: El texto que lo describe debe ser claro, tal que permita entenderlo y leerlo fácilmente.



TEORIA DE LA COMPUTABILIDAD:

La Teoría de la computabilidad es la parte de la computación que estudia los problemas de decisión que pueden ser resueltos con un algoritmo o equivalentemente con la llamada máquina de Turing.

El propósito inicial de la teoría de la computabilidad es hacer precisa la noción intuitiva de función calculable; esto es, una función cuyos valores pueden ser calculados de forma automática o efectiva mediante un algoritmo. Así podemos obtener una comprensión más clara de esta idea intuitiva; y solo de esta forma podemos explorar matemáticamente el concepto de computabilidad y los conceptos relacionadas con ella, tales como decibilidad, etc... Surge así una teoría que producirá resultados positivos y negativos (estamos pensando en resultados de no computabilidad o de indecidibilidad).



DESARROLLO DE LA PRÁCTICA

PROBLEMA: Determinar si un número dado es positivo o negativo.

RESTRICCIONES: El número no puede ser cero.

DATOS DE ENTRADA: Número real.

DATOS DE SALIDA: La validación de si el número es positivo

DOMINIO: Todos los número reales.

Solución del primer problema
con su correspondiente algoritmo y
prueba de escritorio.

SOLUCION:

1. Inicio
2. Leer un numero real
3. Si el numero es 0 ve al final
4. Si el numero es diferente de 0 se valida dos opciones:
 - 4.1 Mayor a 0 es positivo, entonces imprime pantalla
 - 4.2 Menor que 0 es negativo entonces imprime pantalla
5. Imprime pantalla
6. Final

ITERACIÓN	X	SALIDA
1	10	El número 10 es positivo
2	0	Termina mi programa
3	-200	El -200 es negativo

PROBLEMA: Obtener el mayor de dos números dados. RESTRICCIONES:

Los números de entrada deben ser diferentes.

DATOS DE ENTRADA: Número real.

DATOS DE SALIDA: La impresión del número más grande. DOMINIO:

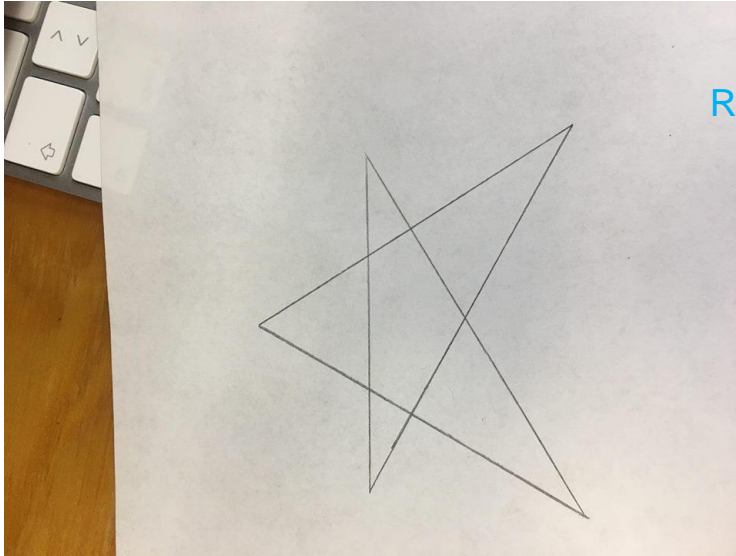
Todos los número reales.

SOLUCIÓN:

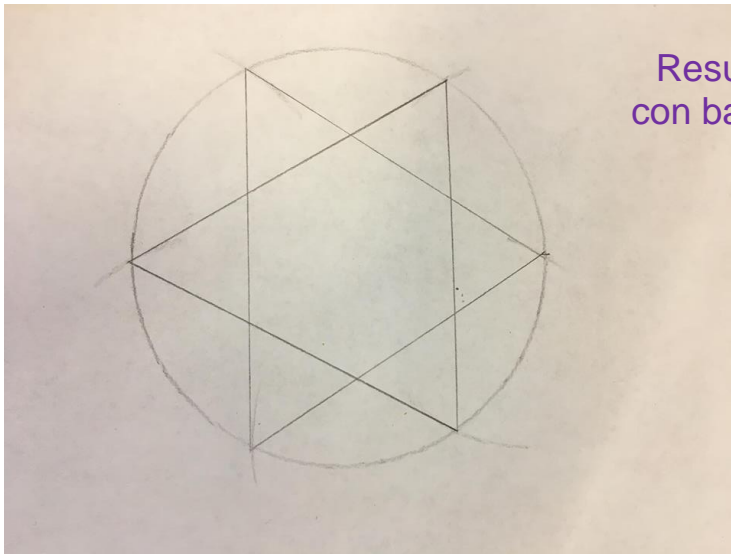
- 1.Inicio
- 2.Ingresar un número real x
- 3.Ingresar otro número real y
- 4.Si al ingresar el segundo número real, resultan ser iguales, regresa al punto 3
- 5.Si resultan ser diferentes entonces las condiciones son:
 - 5.1 $X, Y \dots X > Y$
 - 5.2 $X, Y \dots Y > X$
6. Fin

Solución al segundo
problema con su correspondiente
algoritmo y prueba de escritorio.

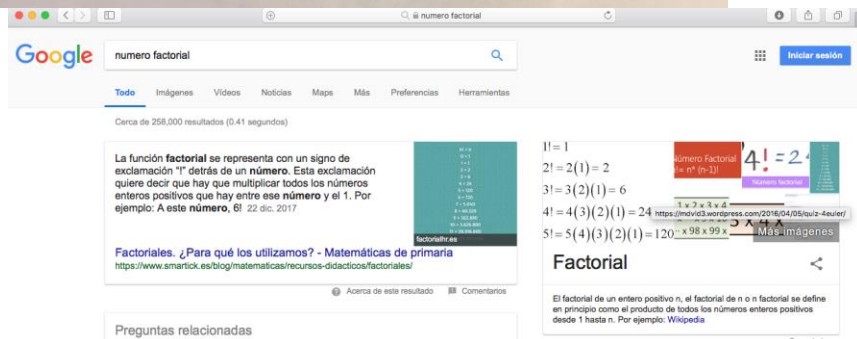
ITERACIÓN	X	Y	SALIDA
1	8	5	$8 > 6$
2	9	9	-
3	3	3	$11 \ 3 < 11$



Resultado final, tras seguir el algoritmo del ejercicio número 1. Se formó una estrella con 5 puntas.



Resultado del ejercicio 2 resuelto en clase con base al algoritmo, se formó una estrella con 6 puntas.



Búsqueda de lo qué es un número factorial.

EJERCICIOS DE TAREA:

1. Calcular el volumen de un cilindro a partir del radio de la base y la altura. (Hacer uso de la formula $V = \pi (r^2) (h)$)

Realización del algoritmo para este primer problema.

Análisis:

Entrada: Radio, π , h

Salida: V

Restricciones: $R > 0$, $h > 0$

DISEÑO:

1. Inicio
2. Introducir R y h
3. Leer R y h
4. Si $R > 0$ y $h > 0$ entonces pasa al punto 6
5. Si $R < 0$ y $h < 0$ no se puede calcular e introduce otros valores, regresa al punto 2 o ve a Fin.
6. Fin

+

Iteración	R	h	Volumen
1	3	4	$V = \pi(9)(4)$
2	5	5	$V = \pi(25)(5)$
3	-3	-5	V= No se puede calcular

2. Calcular la distancia entre dos puntos. Sea $P1(x_1, y_1)$ y $P2(x_2, y_2)$, hacer uso de:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Análisis:

Entrada: P1 y P2

Salida: Distancia entre ellos

Restricciones: Sean números reales

Diseño:

1. Inicio
2. Introducir P1 y P2
3. Leer P1 y P2
4. Calcular su distancia, tomando en cuenta el valor positivo de la raíz
5. Fin

Resolución al segundo problema con su respectivo análisis y pruebas de escritorio.

+

Iteración	P1	P2	$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
1	(-4, 6)	(5, -5)	$\sqrt{202}$
2	(2, 2)	(3, 4)	$\sqrt{5}$
2	(-3, 4)	(2, -1)	$5\sqrt{2}$

3. Leer 2 números y verificar si son divisibles, o el resultado no existe, o es infinito. (Considere que los números deben ser enteros).

Análisis:

Entrada: 2 números enteros

Salida: Verificar si son divisibles, no existe su resultado o es infinito.

Restricciones: No hay, dado que en la salida pide que su resultado no exista, tal es el caso de denominador 0.

Diseño:

1. Inicio
2. Escribir 2 números enteros
3. Leer los números dados
4. Hacer la división de estos números
5. Imprimir el resultado y solo en cada caso realiza lo siguiente:
 - 5.1 "Divisible" y mostrar resultado
 - 5.2 Cuando el divisor sea 0 mostrar "No existe"
6. Fin

Resolución del tercer problema con su respectivo algoritmo y prueba de escritorio.

Iteración	Primer número	Segundo número	Resultado
1	-10	-5	"Divisible" 2
2	15	0	"No existe"

4. Leer un número y verificar si es par o impar.

Análisis:

Entrada: Un número

Salida: Decir si es par o impar

Restricciones: Que el número pertenezca a los enteros

Resolución del cuarto problema con su algoritmo su prueba de escritorio.

Diseño:

1. Inicio
2. Dar un número entero
3. Leer el número dado
4. Dividir el número entre 2 y en cada caso hacer lo siguiente:
 - 4.1 Si al dividir entre 2 su residuo es 0 entonces "par"
 - 4.2 Si al dividir entre 2 su residuo != 0 entonces "impar"
5. Fin

Iteración	Número	Dividir entre 2	Salida
1	10	10/2= 5 residuo 0	"Par"
2	15	15/2=7.5 residuo !=0	"Impar"

CONCLUSIONES:

- Realmente me resultó una práctica un poco compleja dado que al principio me costó organizar los algoritmos para cada problema.
- También no pude resolver el último ejercicio, por más que traté en encontrar una forma para poder resolverlo.
- Por último puedo decir que con esta práctica aprendí y reforcé la construcción de un algoritmo.

BIBLIOGRAFÍA:

- <http://decsai.ugr.es/~castro/MCII/node2.html>
- https://es.wikipedia.org/wiki/Teor%C3%ADa_de_la_computabilidad
- <https://aulatec.wordpress.com/caracteristicas-y-propiedades-de-los-algoritmos/>