|  |
| --- |
| Laboratorio de Computación  Salas A y B |
|  |

|  |  |
| --- | --- |
| Profesor: | Karina García Morales |
| Asignatura: | Fundamentos de Programación |
| Grupo: | 1121 |
| No de Práctica(s): | 3 |
| Integrante(s): | Diego Ramírez Martínez |
| No. de Equipo de cómputo empleado: | 36 |
| Semestre: | 2019-1 |
| Fecha de entrega: | 3 de septiembre de 2018 |
| Observaciones: |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**PRÁCTICA #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 e identificando los módulos de entrada, de procesamiento y de salida.

**DESARROLLO**

**TEORÍA 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. La teoría de la computabilidad se interesa por 4 preguntas:

1. ¿Qué problemas puede resolver una máquina de Turing?
2. ¿Qué otros formalismos equivalen a las máquinas de Turing?
3. ¿Qué problemas requieren máquinas más poderosas?
4. ¿Qué problemas requieren máquinas menos poderosas?

Con respecto a las máquinas de Turing no todos los problemas pueden ser resueltos por estas. A estos problemas se les llama como ***problemas indecidibles***, son aquellos que no pueden ser resueltos con un algoritmo aún si se dispone de espacio y tiempo ilimitado. Actualmente se conocen muchos problemas indecidibles, algunos son:

* ***El Entscheidungsproblem:*** (problema de decisión en alemán), dada una frase del cálculo de predicados de primer orden, decidir si ella es un teorema. Church y Turing demostraron independientemente que este problema es indecidible.
* ***El Problema de la parada:*** dado un programa y su entrada, decidir si ese programa terminará para esa entrada o si correrá indefinidamente. Turing demostró que se trata de un problema indecidible.
* Un número computable es un número real que puede ser aproximado por un algoritmo con un nivel de exactitud arbitrario. Turing demostró que casi todos los números no son computables. Por ejemplo, la Constante de Chaitin no es computable, aunque sí que está definida.

**CARACTERÍSTICAS DE UN ALGORTIMO COMPUTACIONAL**

Un algoritmo es una secuencia de pasos lógicos necesarios para llevar a cabo una tarea específica, como la solución de un problema. Los algoritmos son independientes tanto del lenguaje de programación en que se expresan como de la computadora que los ejecuta. 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. Si se sigue un algoritmo, se debe terminar en algún momento; o sea debe de tener un número finito de pasos.

La definición de un algoritmo debe describir tres partes: **Entrada, Proceso y Salida**, por ejemplo, para elaborar un platillo.

**Entrada:** ingredientes y utensilios empleados.

**Proceso:** elaboración de la receta de cocina.

**Salida:** platillo listo para servir.

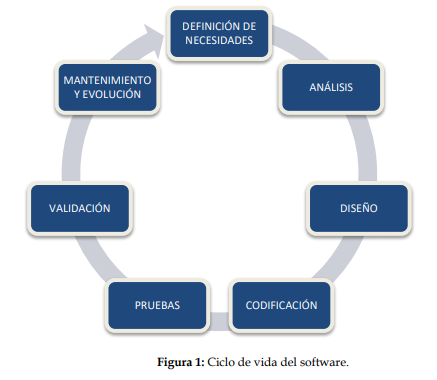
Un algoritmo debe tener varias propiedades importantes, como, por ejemplo:

* Los pasos de un algoritmo deben ser simples y exentos de ambigüedades.
* Deben seguir un orden cuidadosamente prescrito.
* Deben ser efectivos y deben de resolver el problema en un número finito de pasos.

**CICLO DE VIDA DEL SOFTWARE**

La ISO (International Organization for Standarization) en su norma 12207 define al ciclo de vida de un software como:

*“Un marco de referencia que contiene las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando desde la definición hasta la finalización de su uso.”*



**PROBLEMA:** Determinar al número dado es positivo o negativo

**RESTRICCIONES:** El número no puede ser cero.

**DATOS DE ENTRADA:** Número real.

**DATOS DE SALIDA:** La validación si el número es positivo.

**DOMINIO:** Todos los números reales.

**SOLUCIÓN:**

1. Inicio.
2. Introducir cualquier número real y proceder al paso 3.
3. Leer el número y proceder al paso 4.
4. Si el número obtenido es cero, retroceder al paso 2, en caso contrario, proceder al paso 5.
5. Si el número obtenido es mayor que cero (0<n) escribir “POSITIVO” y proceder al paso 6, en caso contrario si el número obtenido es menor que cero (0>n) entonces escribir “NEGATIVO” y proceder al paso 6.
6. FIN.

|  |  |  |
| --- | --- | --- |
| ITERACIÓN | VARIABLE | DATOS DE SALIDA |
| *1* | ***10*** | ***POSITIVO*** |
| *2* | ***-34*** | ***NEGATIVO*** |
| *3* | ***0*** | ***-*** |

**ANÁLISIS**

PROBLEMA: Seguir el algoritmo para obtener una figura

ENTRADA: Hoja tamaño carta en limpio, regla y lápiz.

SALIDA: Figura correcta.

**ALGORITMO**

1. Empieza dibujando un círculo con un compás. Coloca un lápiz en el compás.

Coloca la punta del compás en el centro de una hoja de papel.

2. Ahora gira el compás, mientras mantienes la punta apoyada en el papel. El lápiz

dibujará un círculo perfecto alrededor de la punta del compás.

3. Marca un punto en la parte superior del círculo con el lápiz. Ahora, coloca la punta

del compás en la marca. No cambies el radio del compás con que hiciste el círculo.

4. Gira el compás para hacer una marca en el propio círculo hacia la izquierda. Haz

una marca también en el lado derecho.

5. Ahora, coloca la punta del compás en uno de los puntos. Recuerda no cambiar el

radio del compás. Haz otra marca en el círculo.

6. Continúa moviendo la punta del compás a las otras marcas, y continúa hasta que

tengas 6 marcas a la misma distancia unas de otras. Ahora, ya puedes dejar tu

compás a un lado.

7. Usa una regla para crear un triángulo que empiece en la marca superior del círculo.

Coloca el lápiz en la marca superior. Ahora dibuja una línea hasta la segunda

marca por la izquierda. Dibuja otra línea, ahora hacia la derecha, saltándote la

marca de la parte más baja. Complementa el triángulo con una línea hacia la marca

superior. Así completarás el triángulo.

8. Crea un segundo triángulo empezando en la marca en la base del círculo. Coloca el

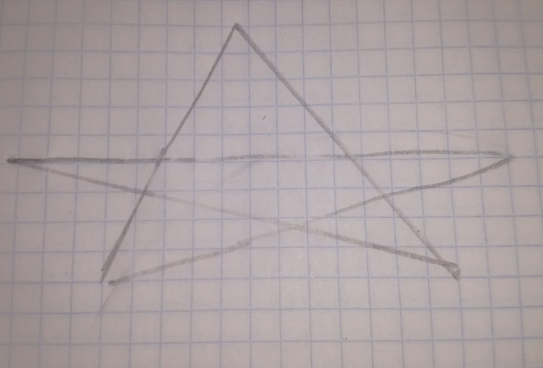
lápiz en la marca inferior. Ahora conéctala con la segunda marca hacia la

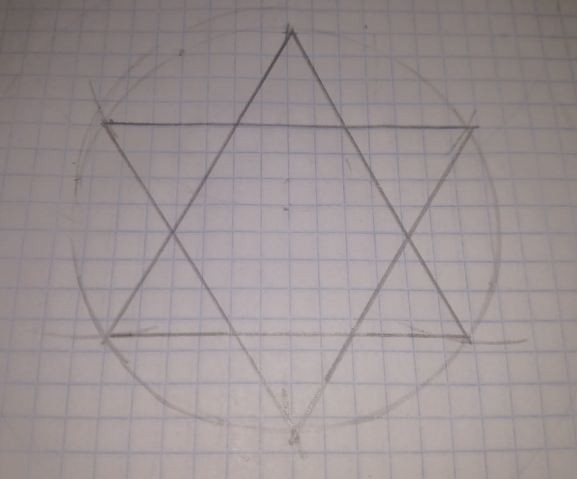
izquierda. Dibuja una línea recta hacia la derecha, saltándote el punto superior.

Completa el segundo triángulo dibujando una línea hasta la marca en la parte

inferior.

9. Borra el círculo. Has terminado de dibujar tu estrella de 6 puntos.





**EJERCICIOS DE TAREA**

1. Calcular el volumen de un cilindro a partir del radio de la base y la altura. (Hacer uso de la fórmula V r h2 = π).

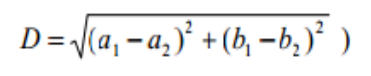
**ANÁLISIS**

* DATOS DE ENTRADA: radio = 3.1416.
* DATOS DE SALIDA: Volumen (V).
* RESTRICCIONES: el radio debe ser mayor o igual a cero (r>=0).
* METODOLOGÍA: V r h2 = π

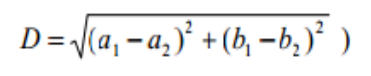
**ALGORITMO**

1. Inicio.
2. Escribir V e ir al paso 2.
3. Leer los valores de (r), (h) y (π) e ir al paso 4.
4. Escribir la fórmula V = (π)(r)2(h) e ir al paso 5.
5. Resolver ecuación e ir al paso 6.
6. Si el radio es menor a cero (r<0) regresar al paso 2 o ir al paso 8.
7. Imprimir el resultado de la ecuación e ir al paso 6.
8. Fin.

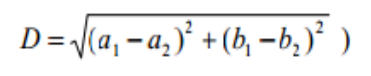
2. Calcular la distancia entre dos puntos. (Sea P1 (a1, b1) y P2 (a2, b2), hacer uso de:



**ANÁLISIS**

* DATOS DE ENTRADA: a1, b1 y a2, b2.
* DATOS DE SALIDA: Distancia entre puntos (D = a1, b1 y b1, b2).
* RESTRICCIONES: (D=0)
* METODOLOGÍA: 

**ALGORITMO**

1. Inicio.
2. Escribir D e ir al paso 3.
3. Leer los valores de (a1, b1 y a2, b2) e ir al paso 4.
4. Escribir la fórmula e ir al paso 5.
5. Resolver la ecuación e ir al paso 6.
6. Si el valor de D es igual a cero (D = 0), regresar al paso 2 o continuar al paso 8.
7. Imprimir el resultado de la ecuación.
8. Fin.

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**

* DATOS DE ENTRADA: números enteros (Z).
* DATOS DE SALIDA: Verificar si son divisibles o no existen o son infinitos.
* RESTRICCIONES: (Z<0).

**ALGORITMO**

1. Inicio.
2. Leer los valores e ir al paso 3.
3. Verificar si son divisibles, el resultado no existe o es infinito e ir al paso 4.
4. Si el resultado es negativo, regresar al paso 2 o continuar al paso 8.
5. Imprimir el resultado.
6. Fin.

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

**ANÁLISIS**

* DATOS DE ENTRADA: números enteros (Z).
* DATOS DE SALIDA: si son números pares o impares.
* RESTRICCIONES: (n<0).

**ALGORITMO**

1. Inicio.
2. Escribir números enteros.
3. Leer los valores e ir al paso 3.
4. Verificar si es número par o impar e ir al paso 4.
5. Si el resultado es negativo, regresar al paso 2 o continuar al paso 8.
6. Imprimir el resultado.
7. Fin.

5. Leer del número 1 al 50 e indicar cuales números son múltiplos de 3.

**ANÁLISIS**

* DATOS DE ENTRADA: números enteros (Z).
* DATOS DE SALIDA: Si son múltiplos de 3.
* RESTRICCIONES: (n<0).

**ALGORITMO**

1. Inicio.
2. Leer del número 1 al 50 e ir al paso 3.
3. (n es 1) y (suma es 0) e ir al paso 4.
4. Verificar si son múltiplos de 3 e ir al paso 5.
5. Imprimir el resultado e ir al paso 6.
6. Fin.

**CONCLUSIONES**

Fue muy interesante realizar esta práctica ya que aprendía realizar algoritmos correctamente y estos son muy importantes.

**BIBLIOGRAFÍA**

http://lcp02.fi-b.unam.mx/