Laboratorios de computación salas A y B

Profesor: García Morales Karina

Asignatura: Fundamentos de Programación

Grupo: 1121

No de Práctica(s):3

Integrante(s): Loeza Encarnación Jafet Tonatiuh

No. de Equipo de cómputo empleado:

Semestre: 1

Fecha de entrega: 04 de septiembre de 2018

Observaciones:

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

Guía práctica de estudio 03: 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.

**DESARROLLO:**

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.

DEFINICIÓN DE NECESIDADES

MANTENIMIENTO Y EVOLUCIÓN

ANÁLISIS

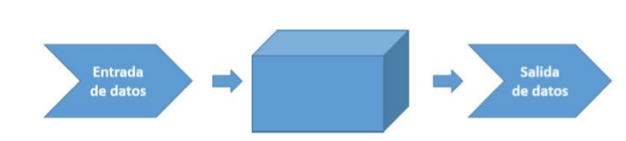
O

**SOLUCIÓN DE PROBLEMAS**

Dentro del ciclo de vida del software, en el análisis se busca comprender la necesidad, es decir, entender el problema.

El conjunto de entrada está compuesto por todos aquellos datos que pueden alimentar al sistema.

El conjunto de salida está compuesto por todos los datos que el sistema regresará como resultado del proceso. Estos datos se obtienen a partir de los datos de entrada.



La etapa de análisis es crucial para la creación de un software de calidad, ya que si no se entiende qué es lo que se desea realizar, no se puede generar una solución. Sin embargo, es común caer en ambigüedades debido al mal entendimiento de los requerimientos iniciales.

Un número factorial se obtiene multiplicando el número inicial por ese mismo número menos uno y todo eso se vuelve factorial y así hasta llegar al número 1.

Ejemplo: 5!= 5(5-1)! y así hasta llegar al 1. 5!=5x4x3x2x1= 120

Las principales características con las que debe cumplir un algoritmo son:

* Preciso: Debe indicar el orden de realización de paso y no puede tener ambigüedad
* Definido: Si se sigue dos veces o más se obtiene el mismo resultado.
* Finito: Tiene fin, es decir tiene un número determinado de pasos.
* Correcto: Cumplir con el objetivo.
* Debe tener al menos una salida y esta debe de ser perceptible
* Debe ser sencillo y legible
* Eficiente: Realizarlo en el menor tiempo posible
* Eficaz: Que produzca el efecto esperado

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:

1. Inicio
2. Leer numero
3. Si numero =0 regresar a paso , de lo contrario pasar a paso 4
4. Si el número es mayor a 0 pasar al paso 5, de lo contrario pasar a paso 6
5. Imprimir “positivo” pasar a paso 7
6. Imprimir “negativo”
7. Fin

|  |  |  |
| --- | --- | --- |
| Iteración | X | Salida |
| 1 | 0 | - |
| 2 | 19 | +19 |
| 3 | -17 | -17 |

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. Solicitar un numero real
3. Solicitar el segundo numero real
4. si los numeros son iguales, regresar al paso 3, de lo contrario pasar al paso 5
5. leer A
6. Leer B
7. si A!=B pasar al paso 8
8. si A>B pasar al punto 10
9. si A<B pasar al punto 11
10. Imprimir “el primer numero es el mayor”
11. Imprimir “el segundo numero es el mayor”
12. Fin

|  |  |  |  |
| --- | --- | --- | --- |
| Interacción | A | B | Salida |
| 1 | 15 | 15 | - |
| 2 | 5 | 1 | 5 es mayor |
| 3 | 17 | 19 | 19 es mayor |

PROBLEMA: Obtener el factorial de un número dado. El factorial de un número está dado por el producto de ese número por cada uno de los números anteriores hasta llegar a 1. El factorial de 0 (0!) es 1.

RESTRICCIONES: El número de entrada debe ser entero y no puede ser negativo.

DATOS DE ENTRADA: Número entero.

DATOS DE SALIDA: La impresión del factorial del número. DOMINIO: Todos los números naturales positivos.

1. Inicio
2. arrojar valor n natural
3. n(n-1)(n-2)...(2)(1)
4. imprimir n
5. fin

|  |  |  |  |
| --- | --- | --- | --- |
| Iteración | N | Factorial | Salida |
| 1 | 4 | 1\*2\*3\*4 | 24 |
| 2 | 6 | 1\*2\*3\*4\*5\*6 | 720 |
| 3 | 3 | 1\*2\*3 | 6 |

Ejercicio 1

PROBLEMA: Seguir el algoritmo para obtener una figura

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

Algoritmo

1. Dibuja una V invertida. Empieza desde el lado izquierdo, sube, y baja hacia el lado derecho, no levantes el lápiz.
2. Ahora dibuja una línea en ángulo ascendente hacia la izquierda. Debe cruzar la primera línea más o menos a 1/3 de la altura. Todavía no levantes el lápiz del papel.
3. Ahora, dibuja una línea horizontal hacia la derecha. Debe cruzar la V invertida más o menos a 2/3 de la altura total. Sigue sin levantar el lápiz.
4. Dibuja una línea en un ángulo descendente hasta el punto de inicio. Las líneas deben unirse.
5. Ahora ya puedes levantar el lápiz del papel. Has terminado la estrella de 5 puntas.

Ejercicio 2

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 dibuja un círculo perfecto alrededor de la punta del compás.

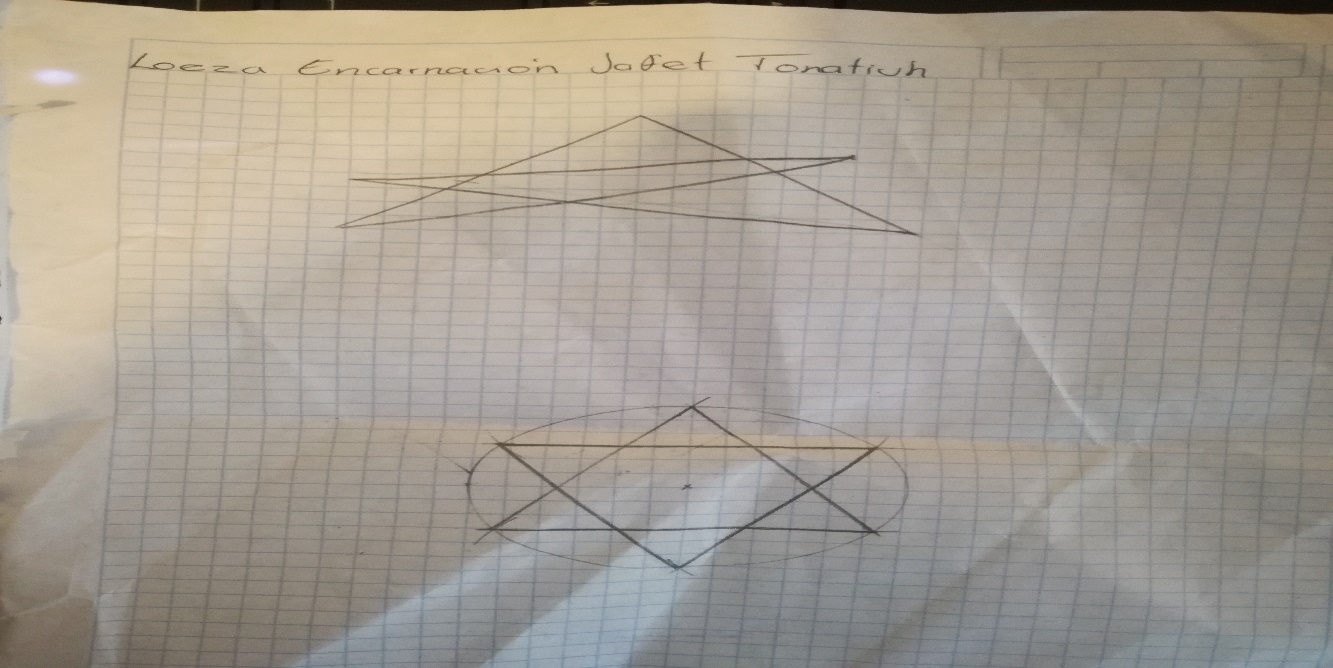
3. Marca un punto en la parte superior del círculo con el lápiz. Ahora, coloca la punta del compás en la marca. No cambies el radio del compás con que hiciste el círculo.

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, saltándose la marca de la parte más baja. Complementa el triángulo con una línea hacia la marca superior. Así completarás el triángulo.



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

ANÁLISIS:

* Datos de entrada:π, ,h
* Restricciones: r=0
* Metodología:

ALGORITMO:

1. Inicio
2. Leer r
3. Si r=0 regresar a paso 2
4. Leer h
5. Multiplicar πr^2(h)=V
6. Imprimir V
7. Fin

|  |  |  |  |
| --- | --- | --- | --- |
| Iteración | r | h | Salida |
| 1 | 4 | 6 | 96π |
| 2 | 6 | 15 | 540π |
| 3 | 0 | 7 | - |

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

ANÁLISIS:

Datos de entrada: P1(a1, b1) P2(a2, b2)

Datos de salida: D (Distancia entre puntos)

ALGORITMO:

1. Inicio
2. Se dará el P1 y P2
3. Definir a1, b1, a2, b2. D: REAL
4. Escribir a1, b1, a2, b2
5. Leer a1, b1, a2, b2
6. D= sqrt[]
7. Imprime D
8. Fin

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Iteración | A1 | A2 | B1 | B2 | Salida |
| 1 | 2 | 11 | 6 | 15 | 12.7279 |
| 2 | 7 | 8 | 10 | 15 | 5.0990 |
| 3 | 12 | 15 | 30 | 34 | 5 |

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

ANÁLISIS:

Datos de entrada: X Y (Dos números)

Restricciones: Números enteros

Datos de salida: Saber si es divisible

ALGORITMO:

1. Inicio
2. Definir x, y como enteros
3. Leer x, y
4. Si x MOD y=0 Entonces ir a paso 5 si no a paso 6
5. Imprimir “x es divisible entre y”
6. Imprimir “No es divisible”
7. Fin

|  |  |  |  |
| --- | --- | --- | --- |
| Iteración | X | Y | Salida |
| 1 | 10 | 5 | x es divisible entre y |
| 2 | 14 | 3 | No es divisible |
| 3 | 69 | 3 | x es divisible entre y |

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

ANÁLISIS:

* Entrada: a
* Salida: par, impar
* Restricción: 0

ALGORITMO:

1. Inicio
2. Leer a
3. a/2
4. Si a/2= número entero imprimir “par” de lo contrario imprimir “impar”
5. Fin

|  |  |  |
| --- | --- | --- |
| Iteración | n | Salida |
| 1 | 5 | “impar” |
| 2 | 6 | “par” |
| 3 | 40 | “par” |

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

ANÁLISIS:

* Entrada: 1<a<50
* Salida: múltiplo de 3
* Restricción: ninguna

ALGORITMO:

1. Inicio
2. Leer a
3. Si 1<a<50 dividir /3, de lo contrario regresas a paso 2
4. Si a/3 es entero imprimir “múltiplo de 3” , de lo contrario regresar a paso dos
5. Imprimir todas las a/3 enteras
6. Fin

|  |  |  |
| --- | --- | --- |
| Iteración | n | Salida |
| 1 | 33 | “Múltiplo de 3” |
| 2 | 69 | “Múltiplo de 3” |