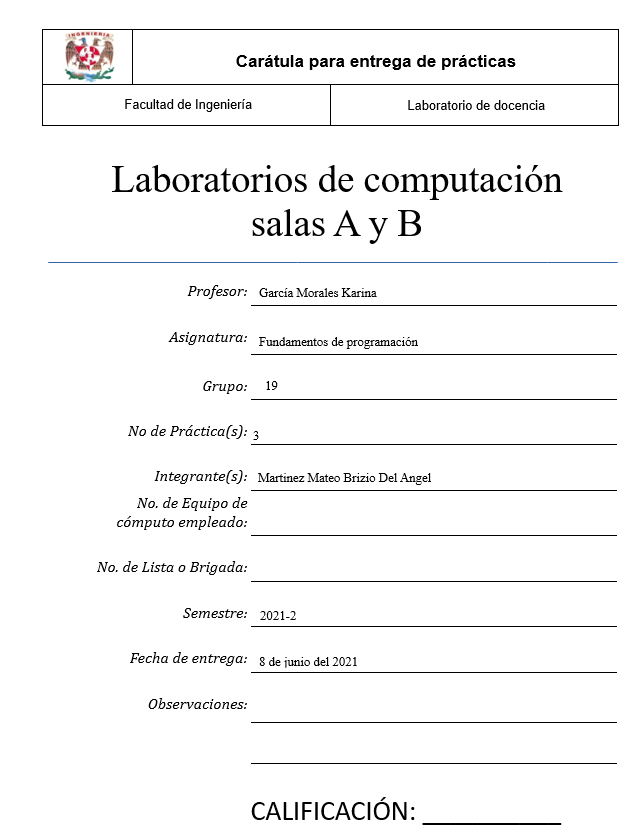
****

.

.

**Ejemplo 1**

**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 números reales.

**SOLUCIÓN:**

1.- Solicitar un número real diferente a cero.  
2.- Evaluar si el número es imaginario, si el numero resulta ser imaginario prosigue al punto 2.1, si el numero resulta no ser imaginario se pasa al punto 3.

2.1.-Imprimir un mensaje de que no se aceptan números imaginarios, se devuelve al usuario al punto 1.

3.-Evaluar si el numero es 0, si el numero es 0 se devuelve al usuario al punto 1. si el numero resulta no ser 0 se pasa al punto 4.

4.- Si el número no tiene una parte imaginaria o es 0, se pueden evaluar si el numero es mayor a 0, si el numero es mayor a 0 se pasa al paso 4.1. si el numero es menor a 0 se pasa al paso 4.2.

4.1. imprimir “el numero ingresado” es positivo, pasar al paso 5.

4.2. imprimir “el numero ingresado” es Negativo, pasar al paso 5.

5.- preguntar al usuario si desea ingresar otro número, si el usuario dice si el programa se reinicia, si el usuario dice no el programa finaliza.

|  |  |  |
| --- | --- | --- |
| **Iteración** | Número | Salida |
| 1 | 123 | 123 es positivo |
| 2 | -232 | -232 es negativo |
| 3 | 0 | - |
| 4 | i | - |

**Ejemplo 2**

**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 números reales.

**SOLUCIÓN:**

1.- Solicitar un primer número real.

2.- Solicitar un segundo número real diferente al primero.

3.-Evaluar si los números son imaginarios, si los números resultan ser imaginarios prosigue al punto 3.1, si los números resultan no ser imaginarios se pasa al punto 4.

3.1.-Imprimir un mensaje de que no se aceptan números imaginarios, se devuelve al usuario al punto 1.

4.- Evaluar si el segundo número es igual al primer numero si el numero es igual se regresa al usuario al punto 1, si no es así el usuario pasa al punto 5.

5.-Evaluar si el primer numero es mayor al segundo, de ser así pasa al punto 5.1, si no es así pasa al punto 5.2.

5.1.- Imprimir El “primer número” es mayor, pasa al punto 6.

5.2.- Imprimir El “segundo número” es mayor, pasa al punto 6.

6.- preguntar al usuario si desea ingresar otro número, si el usuario dice si el programa se reinicia, si el usuario dice no el programa finaliza.

|  |  |  |  |
| --- | --- | --- | --- |
| **Iteración** | Primer Número | Segundo Número | Salida |
| 1 | 523 | 21 | El 523 es mayor |
| 2 | -242 | -435 | -242 es mayor |
| 3 | 0 | 34234 | - |
| 4 | i | 432456 | - |

**Ejemplo 3**

**PROBLEMA:** Obtener la factorial de un número dado. La 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.

**Solución:**

1.- Solicitar un número real.

2.- Si el número es <0 regresar al paso 1, en caso contrario pasar al punto 3.

3.- Si el número es mayor igual a 0 se comienza un contador en 2 y una factorial en 1 y se procede al punto 4.

4.- Si el contador es menor a el numero proceder a el paso 4.1.

4.1.- Realizar la operación contador \*factorial y guardar el resultado en factorial y proceder a el paso 4.2.

4.2.- Incrementar el contador en +1 y repetir el paso 4.

5.- Si el contador es mayor al número se procede a mostrar el contenido de factorial y terminar el programa.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| iteración | Numero | Factorial | Contador | impresión |
| 1 | -9 | - | 2 | Error |
| 2 | 5 | 1 | 2 | - |
| 3 | 5 | 2 | 3 | - |
| 4 | 5 | 6 | 4 | - |
| 5 | 5 | 24 | 5 | - |
| 6 | 5 | 120 | 6 | 5!=120 |

**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.

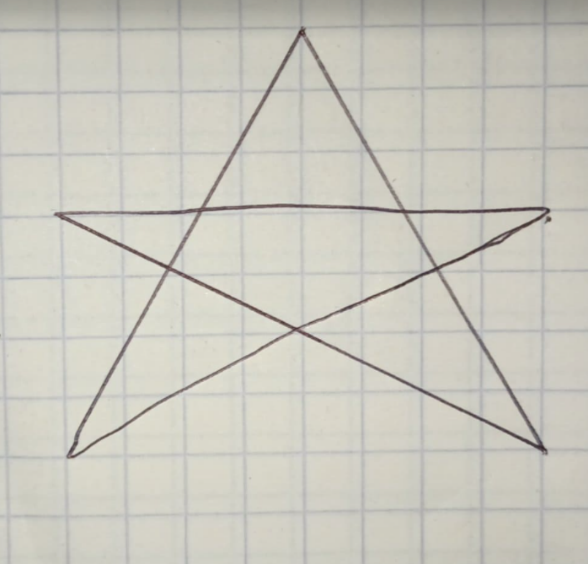
3. Ahora, dibuja una línea horizontal hacia la derecha. Debe cruzar la V invertida más

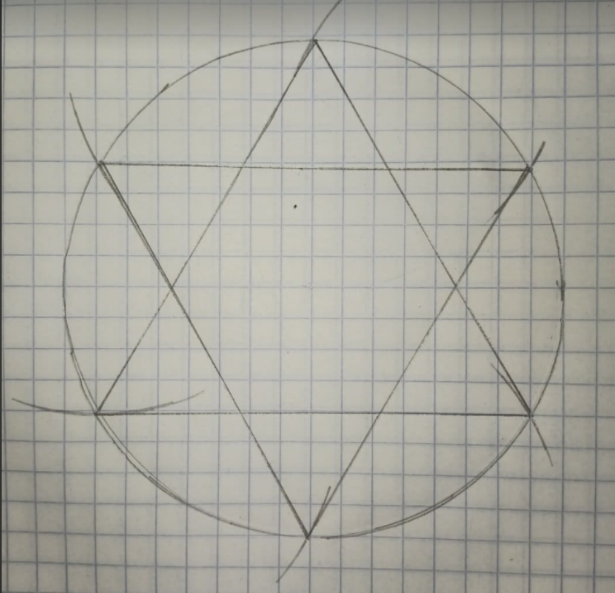
o menos a 2/3 de la altura total. Sigue sin levantar el lápiz.

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 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.

**Tarea:**

1. describe las salidas de los ejercicios finales y la comparación entre ambos algoritmos.

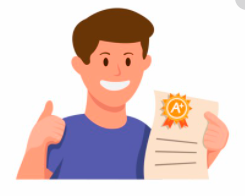
Las salidas de ambos ejercicios son estrellas de 5 y 6 puntas respectivamente, el primer algoritmo es mucho más sencillo que el segundo ya que el según pide compas y regla, parecen más instrucciones de un ejercicio de dibujo técnico.

2. ¿Qué se necesita para dar solución a un problema?

Una serie de pasos para resolverlo de llamándolo de otro paso un algoritmo hecho a la medida para la resolución de un problema dado

3. Describe las fases del ciclo de vida del desarrollo de software explicadas en clase e ilustra

* Definición de necesidades: toma de requisitos sobre qué tiene que hacer el sistema que hay que desarrollar.
* Análisis: revisión de los requisitos revisando si existen carencias o inconsistencias.
* Diseño: descripción de los diferentes componentes tecnológicos y cómo van a interactuar entre ellos
* Codificación: implementación de las diferentes funcionalidades por medio de lenguajes de programación.
* Pruebas: confirmación que los desarrollos no contienen errores que el diseño es completo y consistente y que los requisitos y objetivos se cumplen
* Validación: La validación consiste en en poner en practicas a los componentes ya que es posible que algunos componentes funcionen bien por separado pero en conjunto causen problemas



* Mantenimiento y evolución:

Esta es una de las fases más importantes del ciclo de vida de desarrollo del software. Puesto que el software ni se rompe ni se desgasta con el uso, su mantenimiento incluye tres puntos diferenciados:

Eliminar los defectos detectados durante su vida útil (mantenimiento correctivo).

Adaptarlo a nuevas necesidades (mantenimiento adaptativo).

Añadirle nuevas funcionalidades (mantenimiento perfectivo).

4. Analizar las siguientes problemáticas, desglosando y dando solución al problema identificando claramente su conjunto de entrada (datos de entrada), el conjunto de salida (datos de salida) y restricciones, si es que tiene.

* En una playa el estacionamiento cobra $ 2.5 por hora o fracción. Determinar cuánto debe pagar un cliente por el estacionamiento de su

vehículo, conociendo el tiempo de estacionamiento en horas y minutos.

**RESTRICCIONES:** Los números de entrada deben ser diferentes.

**DATOS DE ENTRADA:** Costo de la hora o fracción, Tiempo en minutos u horas.

**DATOS DE SALIDA:** Costo del estacionamiento.

**Solución:**

1.Solicitar el ticket de entrada

2.Revisar la hora de entrada

3.Calcular las horas dentro en el estacionamiento

4. si hay minutos hay que seguir al paso 5. Sí no hay al paso 6.

5.Cobrar los minutos como otra hora. Minutos>0

6.Multiplicar las horas y fracciones por $2.5

7. Obtener el costo del estacionamiento

* Dados dos números realizar suma, resta, producto y división (considerar el dividendo ≠ 0).

**RESTRICCIONES:** La división no puede efectuarse entre 0.

**DATOS DE ENTRADA:** Dos números

**DATOS DE SALIDA:** Resultado de la suma, Resultado de la resta, Resultado de la multiplicación, Resultado de la división

**Resultado:**

1.Ingresar el primer número (≠0 por la división)

2.Ingresar el segundo número

3.Realizar n1+n2, pasar al paso 7

4.Realizar n1-n2, pasar al paso 8

5.Realizar n1\*n2, pasar al paso 9

6.Realizar n1/n2, pasar al paso 10

7.Imprimir el resultado de la suma

8. Imprimir el resultado de la resta

9. Imprimir el resultado de la multiplicación

10. Imprimir el resultado de la división

* Dados tres números; si el primero es negativo, deberá realizar el producto de los tres, y en caso contrario realizará la suma.

**RESTRICCIONES:** El primer número ≠0

**DATOS DE ENTRADA:** Tres números

**DATOS DE SALIDA:** Resultado de la operación correspondiente

**Solución:**

1.Ingresar el primer número (≠0)

2.Ingresar el segundo número

3.Ingresar el tercer número

4.Identificar si es negativo o positivo el primer número. Si es negativo ir al paso siguiente, si es positivo ir a paso 7

5.Realizar n1\*n2\*n3

6.Resultado de la multiplicación de los tres números

7. Realizar n1+n2+n3

8.Resultado de la suma

GitHub

<https://github.com/BrizioMartinez/practica3_fdp>