



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorios de docencia

03: Solución de Problemas y Algoritmos.

Profesor: García Morales Karina.

Asignatura: Fundamentos de Programación.

Grupo: 1121

No de Práctica(s): 3

Integrante(s): Vega Angel Leonardo Yael

Semestre: Primero.

Fecha de entrega: 04/09/2018

Observaciones:

CALIFICACIÓN: _____

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.

Ciclo de vida de un software.



Figura 1: Ciclo de vida del software.

¿Cómo se obtiene un número factorial?

¿Qué es la función factorial?

La función factorial se representa con un signo de exclamación "!" detrás de un número. Esta exclamación quiere decir que hay que multiplicar todos los números enteros positivos que hay entre ese número y el 1.

Por ejemplo:

$$6! = 1 \times 2 \times 3 \times 4 \times 5 \times 6 = 720$$

A este número, **6!** le llamamos generalmente "**6 factorial**", aunque también es correcto decir "**factorial de 6**".

En tu calculadora podrás ver una tecla con "n!" o "x!". Esta tecla te servirá para calcular directamente el factorial del número que quieras.

Algunos ejemplos de factoriales

Vamos a ver algunos ejemplos más de factoriales:

$$\begin{aligned}4! &= 1 \times 2 \times 3 \times 4 = 24 \\10! &= 1 \times 2 \times \cdots \times 9 \times 10 = 3628800 \\100! &= 1 \times 2 \times 3 \times \cdots \times 98 \times 99 \times 100 \approx 9,33 \times 10^{157}\end{aligned}$$

Como ves, 100! es enorme...

Y, ¿qué hacemos con los números más pequeños? 1 factorial es, lógicamente, 1, ya que multiplicamos 1 x 1:

¿Qué es la teoría de la compatibilidad. ?

Es la parte de la computación que estudia los problemas de decisión que pueden ser resueltos con un algoritmo.

Características del algoritmo. Un algoritmo es la parte más importante y durable de las ciencias de la computación debido a que éste puede ser creado de manera independiente tanto del lenguaje como de las características físicas del equipo que lo va a ejecutar.(Pregunta de examen)

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

Ejercicio 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 número reales.



1. Inicio.
2. Solicitar el número real.
3. Leer número real
4. Si el número ingresado es cero regresar al punto 2 si no pasar al punto 5
5. Si el numero ingresa es mayor a 0 ir el numero es positivo e ir a paso 7 en caso de ser menor a 0 ir a paso 6
6. Si el numero ingresado es menor a 0 es negativo ir a paso 7
7. Fin.

Iteración.	x	Salida.
1	0	-
2	10	Es positivo.
3	-2	Es negativo.

Ejercicio 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 número reales.

- 1.Inicio.
2. Solicitar el primer número real A.
3. Leer A
- 4.Solicitar segundo número real B.
- 5.Leer B
- 6.Si $B=A$ regresar al punto 4 en caso contrario ir a paso 7.
7. Si $B \neq A$ ir a paso 8.
8. $A > B$, se puede afirmar que el primer numero es el mayor de los números ir a paso 10, en caso de lo contrario ir a paso 9.
- 9.Se puede afirmar que el segundo numero es el mayor de los números ir a paso 10.
- 10.Fin.

Interacción.	A	B	
1	20	5	El numero A es el mayor.
2	6	120	El numero B es mayor
3	20	20	-

Ejercicio 3.

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.

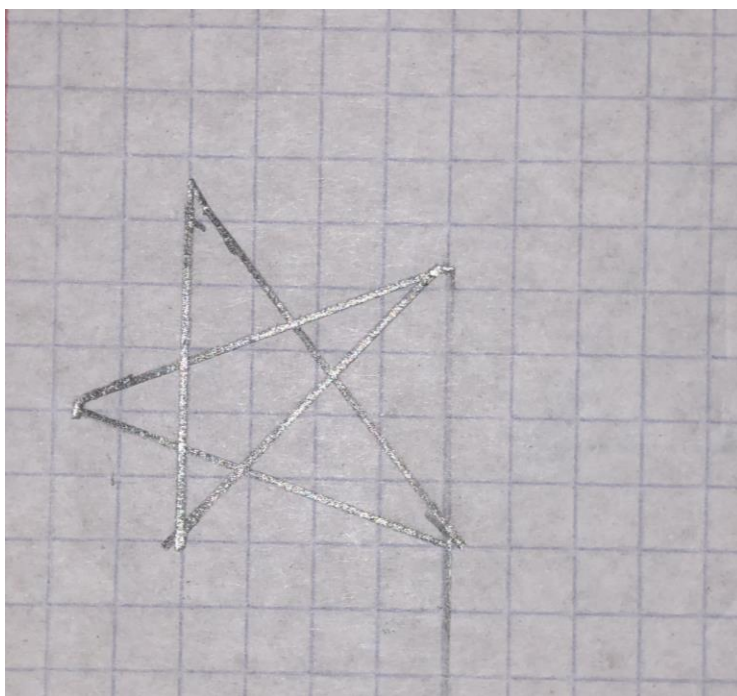
DATOS DE SALIDA: La impresión del factorial del número.

DOMINIO: Todos los números naturales positivos.

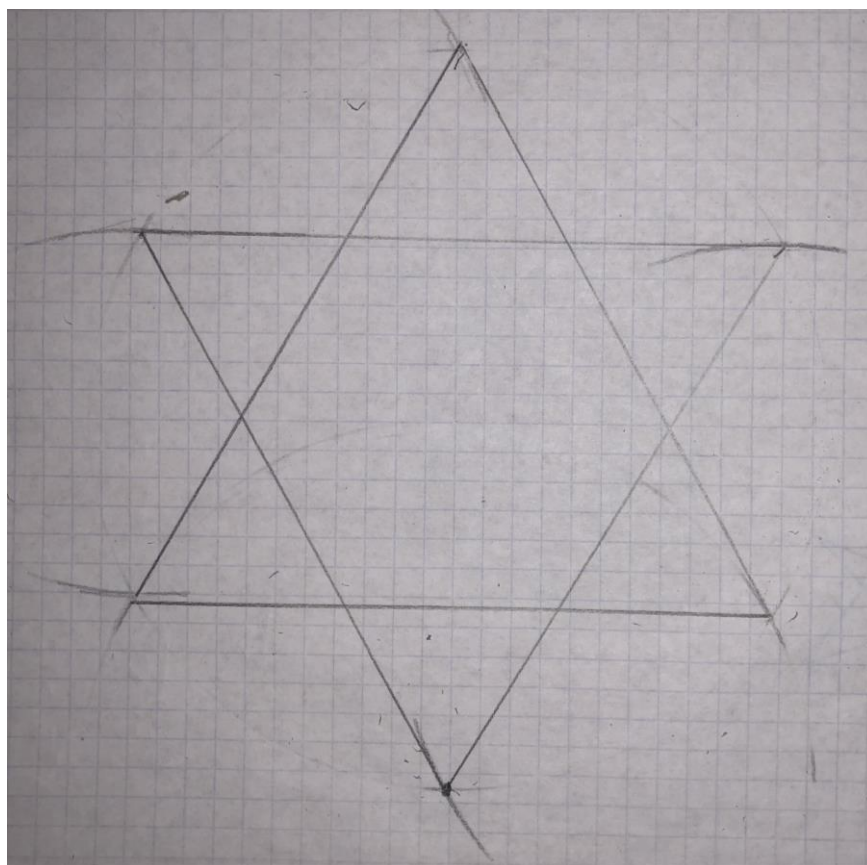
1. Inicio.
2. Solicitar un numero entero x
3. Leer x.
4. Si x es menor a cero regresar al paso 1.
5. Si x es mayor a cero se crea una variable entera contador que inicie en 2 y una variable entera factorial que inicie en uno.
6. Si la variable contador es menor o igual al número entero de entrada pasar a paso 7.
7. Multiplicar la variable contador con el valor factorial, el resultado se almacena en la variable factorial.
8. Se utiliza la siguiente formula $cont = cont + 1$ después regresar a paso 6.
9. Si la variable contador no es menor o igual al número entero se muestra el resultado almacenado en la variable factorial.
10. Fin.

Iteración	X	Factorial	Contador	Salida.
1	4	1	2	-
2		2	3	-
3		6	4	-
4		24	5	El factorial de 4 es 24

Ejercicio 4.



Ejercicio 5.



Tarea

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

Análisis.

- Datos de entrada:
Radio, Base y Altura.
- Salida:
 π
- Restricciones:
 $V > 0$, $r > 0$, h
- Metodología:
 $V = \pi r^2 h$

Algoritmo.

1. Inicio.
2. Escribir valores para r y h ,
3. Leer r , h
4. Si $3.1415 * r^2 * h > 0$ pasar a paso 5 en caso de lo contrario ir a paso 2.
5. $3.1415 * r^2 * h = C$ imprimir e ir a paso 6
6. Fin.

Iteración	R^2	h	Salida.(c)
1	4	6	301.5936
2	1	2	6.2832
3	12	0	-

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

$$D = \sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2}$$

Análisis.

- Datos de entrada:
Coordenadas P1 (a1, b1) y P2 (a2, b2)
- Salida:
Distancia = D
- Restricciones:
D>0
Se debe sustituir los valores en la fórmula por orden primero los datos de una coordenada y se les resta los de la segunda coordenada.
- Metodología:

$$D = \sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2}$$

Algoritmo.

1. Inicio
2. Leer valores de P1 (a1, b1)
3. Leer valores de P2 (a2, b2)
4. Sustituir a1, a2, b1 y b2 en la fórmula.
5. Realizar operación ir a paso 6.
6. Si D>0 ir paso 8 en caso de lo contrario ir a paso 7.
7. Imprimir "no existe distancia entre las coordenadas" e ir a paso 8.
8. Fin.

Iteración	(a1,b1)	(a2,b2)	Salida.(D)
1	(1,6)	(2,2)	4.12
2	(10,10)	(5,5)	7.071
3	(2,4)	(6,1)	5

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:
2 números enteros, los cuales se deben dividir.
- Salida:
Son divisibles, no existe, es infinito.
- Restricciones:
Los números deben de ser enteros.
- Metodología:
 $A/B=C$

Algoritmo.

1. Inicio.
2. Escribir valores para A y B
3. Leer A y B.
4. Dividir $A/B=C$ e ir paso 5.
5. Si C es entero imprimir "son divisibles", C no es entero "no existe" o si C es infinito "es infinito", pasar a paso 6.
6. Fin

Iteración	A	B	Salida.
1	4	2	Son divisibles
2	10	3	No existe
3	1	0	Es infinito.
4	24	12	Son divisibles.

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

Análisis.

- Datos de entrada:
Un número (n) y verificar si un número es par o impar.
- Salida:
Par o Impar
- Restricciones:
n>0
Los números deben de ser enteros.
- Metodología:
N/2

Algoritmo.

1. Inicio.
2. Escribir valores para n
3. Leer n
4. Dividir $n/2=x$ e ir paso 5.
5. Si X es entero imprimir "par", C no es entero "Impar", pasar a paso 6.
6. Fin

Iteración	n	/2	Salida.
1	4	2	Par
2	3	2	Impar
3	1	2	Impar
4	60	2	Par.

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

Análisis.

- Datos de entrada:
Números de 1 a 50
- Salida:
Es múltiplo de 3
- Restricciones:
cont \geq 0
cont \leq 50
cont: 0
Los números deben de ser múltiplos de tres y enteros
- Metodología:
 $n/3 = \text{Entero}$.

Algoritmo.

1. Inicio.
2. Escribir cont =0
3. Utilizar cont +1= cont
4. Leer cont
5. Dividir $\text{cont}/3 = x$ e ir paso 5.
6. Si x es entero imprimir valor de cont e ir a paso e ir paso 3, si x no es entero tomar el valor de cont regresar a paso 3
7. Fin

Iteración	Cont.	/3	Salida.
1	1	3	-
2	2	3	-
3	3	3	3
4	4		-

Conclusión.

Esta práctica fue muy útil debido a que así podemos observar el análisis que realiza una computadora para realizar operaciones complejas y como se crea un algoritmo en el cual se explica paso a paso lo que se debe de hacer una máquina para ejecutar una operación y llegar a la solución que se requiere, aquí se pudo observar como la máquina realiza las operaciones en cuestión de segundos analizando y procesando los datos que nosotros le proporcionamos.

Referencias

- Raghu Singh (1995). International Standard ISO/IEC 12207 Software Life Cycle Processes. Agosto 23 de 1996, de ISO/IEC. Consulta: Junio de 2015. Disponible en: <http://www.abelia.com/docs/12207cpt.pdf>
- Carlos Guadalupe (2013). Aseguramiento de la calidad del software (SQA). [Figura 1]. Consulta: Junio de 2015. Disponible en: <https://www.mindmeister.com/es/273953719/aseguramiento-de-la-calidad-del-software-sqa>
- Andrea S. (2014). Ingeniería de Software. [Figura 2]. Consulta: Junio de 2015. Disponible en: <http://ing-software-verano2014.blogspot.mx>
- Michael Littman. (2012). Intro to Algorithms: Social Network Analysis. Consulta: Junio de 2015, de Udacity. Disponible en: <https://www.udacity.com/course/viewer#!/ccs215/I-48747095/m-48691609>