



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor:

Adrian Ulises Mercado Martínez

Asignatura:

Fundamentos de programación

07

Grupo:

No de Práctica(s):

05

Integrante(s):

Arredondo Granados Gerardo
Casillas Herrera Leonardo Didier
Diaz Gonzalez Rivas Angel Iñaqui
Galván Castro Martha Selene

*No. de Equipo de cómputo
empleado:*

20

No. de Lista o Brigada:

04

Semestre:

2022-1

Fecha de entrega:

6 de noviembre del 2021

Observaciones:

CALIFICACIÓN: _____

Práctica 5

Pseudocódigo

Índice

Objetivo:	3
Introducción	3
Desarrollo	4
Ejercicio 1	4
Ejercicio 2	5
Ejercicio 3	6
Ejercicio 4	7
Ejercicio 5	8
Conclusiones	9
Conclusión del grupo	9
Conclusión individual	9
Galvan Castro Martha Selene:	9
Referencias	10

Objetivo:

El alumno elaborará pseudocódigos que representen soluciones algorítmicas empleando la sintaxis y semántica adecuadas.

Introducción

Pseudocódigo:son una herramienta que para organizar los pasos a realizar en un programa es un intermedio entre el lenguaje literario y el lenguaje de programación con esta herramienta que ayuda a identificar los elementos a usar en el programa.

Características de un pseudocodigo:

- Es independiente del lenguaje de programación.
- Es un método que ayuda a la solución de algoritmos.
- Facilitar la creación del código del programa.
- Describe todo tipo de instrucciones (primitivas, de proceso, de control, compuestas, de descripción).

Partes de un pseudocodigo:

- ❖ **Inicio:** Principio de pseudocódigo.
- ❖ **variables:** Declaración de las variables.
- ❖ **Constante:** Declaración de constantes
- ❖ **Escribir:** Muestras información en la pantalla .
- ❖ **Leer:** Lectura de las variables.
- ❖ **Proceso:** Operaciones a efectuar.
- ❖ **Mientras:** Iteración hasta que se cumpla la condición realizar esa acción.
- ❖ **Fin Mientras:** Fin de la iteración mientras. .
- ❖ **Si:** Iteración si se cumple una condición.
- ❖ **Si-No:** Si no se completa esta condición.
- ❖ **Fin Si:** Cierre de la iteración si.
- ❖ **Según:** Iteración si una variable cumple con distintas condiciones dependiendo de sus valor seguir una serie de acciones.
- ❖ **De Otro Modo:** Si no se completa esta condición.
- ❖ **Fin según:** Cierre de la iteración si.
- ❖ **Repetir:** Iteración de repetir una serie de pasos
- ❖ **Hasta Que:** Indica las veces a repetir la iteración Repetir.
- ❖ **Para:** Si la variable cumple en un conjunto cerrado de valores se realiza una acción/es.
- ❖ **Fin Para:** Cierre de la iteración para.
- ❖ **Imprimir:** Salida de datos.
- ❖ **Fin:** Cierre del pseudocódigo.

Desarrollo

Ejercicio 1

- Construye un algoritmo que reciba tres números reales, identifique cuál es el mayor. Considere que los números pueden ser iguales.

Restricción: que el número sea real y entero

Datos de entrada: 3 números reales

Datos de salida: El número mayor

Dominio: Todos los números enteros

1. Solicitar un número real y almacenarlo en una variable.
2. Solicitar otro número real y almacenarlo en otra variable.
3. Solicitar otro número real y almacenarlo en otra variable.
4. Hacer una condición if, si el primer número es mayor al segundo y tercero, imprimir que el número 1 es mayor.
5. Si no se cumple esto, abrir otra condición if en el else del primer if.
6. En este if se validará si el número 2 es mayor al número 3.
7. Si esto es verdad, se imprimirá el número 2.
8. Si no se cumple, llegamos al segundo else.
9. Se imprimirá el número 3

Pseudocódigo:

1. Inicio
2. a, b, c: Real
3. Escribir "ingresa el primer número"
4. Leer primer número
5. $a :=$ primer número
6. Escribir "ingresa el segundo número"
7. Leer segundo número
8. $b :=$ segundo número
9. Escribir "ingresa el tercer número"
10. Leer tercer número
11. $c :=$ tercer número
12. SI $a \geq b \ \& \ a \geq c$
13. Escribir "El mayor es el:", +primer número
14. DE LO CONTRARIO
15. SI $b \geq c \ \& \ b \geq a$
16. Escribir "El mayor es el:", +segundo número
17. DE LO CONTRARIO
18. Escribir "El mayor es el:", +tercer número
19. FIN SI
20. FIN SI
21. FIN

Ejercicio 2

- Elabore un algoritmo que permita imprimir de forma separada los cuatro dígitos que forman un número.

Sugerencia: Utilice la división entera y el operador de módulo.

Datos de entrada: Número entero de 4 dígitos

Datos de salida: 4 dígitos que formen un número

Dominio: Números enteros mayores a 999 y menores a 10 000

1. Solicitar un número de 4 dígitos y almacenarlo en una variable.
2. Si el número entero no tiene 4 cifras, vuelve al paso 1.
3. Separar el número en millares, centenas, decenas, unidades.
4. Imprimir los dígitos formados con separación en el orden millar, centena, decena y unidad.

Pseudocódigo:

1. **Inicio**
2. a: Entero
3. **Escribir** "ingresa el número"
4. **Leer numero**
5. **SI** numero > 999 & número < 10000
6. a := número
7. **Sí No**
8. Escribir "número no válido"
9. **Fin SI**
10. millar := a/1000
11. centena := a-millar*1000/100
12. decena := a- (millar*1000 + centena*100)/10
13. unidad := a-(millar*1000+centena*100+decena*10)
14. Escribir millar, "millares", centena, "centenas", decena, "decenas", unidad, "unidades"
15. **FIN**

Ejercicio 3

- **Elabore un algoritmo para determinar los divisores de un número.**

Restricción: Números enteros positivos y mayores a 0

Datos de entrada: Un número natural que no sea 0

Datos de salida: Divisores del número entero

Dominio: Todos los números naturales excepto el 0

1. Solicitar al usuario un número natural que no sea 0 y guardarlo como una variable..
 - 1.1 Si el número entero es menor a cero, vuelve al paso 1.
2. Dividir el número entre valores menores o iguales al mismo, tales que su residuo sea igual al número cero
3. Todos los números con los que el residuo NO fue igual a cero, descartarlos
4. Imprimir los números para los cuales el residuo de la división es igual a cero

Pseudocódigo:

1. Inicio
2. a, b: Entero
3. b := 1
4. Escribir "ingresa el número"
5. Leer numero
6. SI numero > 0
7. a := número
8. DE LO CONTRARIO
9. Escribir "El número ingresado no es válido"
10. FIN SI
11. Mientras a > b hacer
12. c := a mod b
13. Si c ← 0
14. Entonces
15. Divisores := a/b
16. FinSi
17. b ← b+1
18. Fin Mientras
19. Escribir "El número es divisible entre", Divisores
20. Fin

Ejercicio 4

- Se dice que un número N es primo si los únicos enteros positivos que lo dividen son exactamente 1 y N . Elabore el algoritmo que de solución al problema.

Restricción: Número de entrada debe de ser entero y positivo

Datos de entrada: Un número entero

Datos de salida: El número es primo o no lo es

Dominio:

Los números entero de (0, infinito)

1. Solicitar al usuario un número entero positivo
 - 1.1 Si el número entero es menor a cero o igual a este, volver al paso 1
2. Almacenar ese número en una variable de tipo entero
3. Empezar a dividir el número empezando con 1 número uno, siguiendo del número 2, siguiendo del número 3, y así sucesivamente hasta alcanzar el mismo número que ha sido ingresado
4. El conjunto de números que arroje como residuo 0, deberán ser el número 1, y el mismo número ingresado
5. Si es que se cumple el paso 4, mostrar en pantalla "El número es primo"
 - 5.1 De lo contrario, mostrar en pantalla "El número no es primo"

Pseudocódigo:

1. Inicio
2. a : entero
3. Contador = 0
4. Escribir "Ingrese un número"
5. Leer el número en la variable a
6. Para $i \leftarrow$ hasta a hacer
7. Si $a \% i = 0$ entonces
8. contador \leftarrow contador + 1
9. FinSi
10. Fin Para
- 11.
12. Si contador=2 entonces
13. Escribir a , "es un número primo"
14. SiNo
15. Escribir a , "no es un número primo"
16. FinSi
17. Fin Algoritmo

Ejercicio 5

- Elaborar un algoritmo que reciba un número entero y calcule los números perfectos que existen entre 1 y el número seleccionado. Un número se considera perfecto si la suma de todos sus divisores son igual a dicho número.

Restricción: Número entero y mayor a 1

Datos de entrada: Un número entero

Datos de salida: Los números perfectos entre el 1 y el número seleccionado

Dominio: Todos los números enteros empezando por el 1

1. Solicitar al usuario un número entero positivo
 - 1.1 Si el número entero es menor a cero, vuelve al paso 1.
2. Almacenar el número en una variable de tipo entero.
3. Dividir el número entre todos los números enteros positivos menores a él.
4. Sumar todos los divisores encontrados.
5. Comparar el resultado de la suma con el número dado, si el resultado fue igual al número dado entonces es un número perfecto, si el resultado fue diferente al número dado entonces no es un número perfecto.

Pseudocódigo:

1. Inicio
2. n,x,perfecto: Enteros
3. Escribir "Ingrese un número"
4. Leer el número en la variable n
5. x=2
6. Mientras x <= n Hacer
7. si n mod x == 0 Entonces
8. perfecto=perfecto + (n/x)
9. FinSi
10. x=x+1
11. Fin Mientras
- 12.
13. si perfecto==n Entonces
14. Escribir "El número ", n, " es perfecto"
15. SiNo
16. Escribir "El número ", n, " no es perfecto"
17. FinSi
18. Fin Algoritmo

Conclusiones

Conclusión del grupo

Concluimos que esta fue una práctica muy interesante, ya que con los algoritmos de los problemas al crear el pseudocódigo nos acercamos un poco más al lenguaje de programación por medio de la gran herramienta que son los pseudocódigos, además cumplimos el objetivo principal de la práctica el cual era elaborar los pseudocódigos en la solución de problemas, lo cual se pudo cumplir con los cinco ejercicios, logramos elaborar pseudocódigos funcionales y con la sintaxis y semántica correcta de problemas variados.

Conclusión individual

Arredondo Granados Gerardo: En lo personal la práctica me pareció interesante porque recolectamos el conocimiento necesario para poder pasar de lleno ahora a programar, todos los conocimientos de las prácticas anteriores se vieron utilizados en esta práctica, lo cual se me hace interesante el ver cómo todos los ejercicios de las prácticas anteriores nos pueden servir para los próximos.

Casillas Herrera Leonardo Didier: Se cumplió el objetivo que tenía esta práctica ya que pudimos elaborar los pseudocódigos a los diferentes problemas que se nos habían planteado con anterioridad y logramos realizar un pseudocódigo que nos va a poder servir para posteriormente resolver problemas en un lenguaje de programación.

Diaz Gonzalez Rivas Angel Iñaqui: En esta práctica pudimos emplear los operadores lógicos, asignar un tipo de dato a las variables y escribir las estructuras de control de flujo a partir de su sintaxis, para dar forma a un pseudocódigo. Los pseudocódigos permitieron dar una solución al problema solicitado, desde la codificación del algoritmo partiendo de la representación gráfica del algoritmo.

Galvan Castro Martha Selene: En esta práctica con los problemas se pudo practicar el uso de los distintos elementos de los pseudocódigos como son el uso correcto tanto de la estructura como de sus distintas iteraciones como son mientras, para, etc. además que nos dio una primera vista de que es el lenguaje de programación y nos ayudó para acabar de ordenar la información, y mejorar nuestros algoritmos previamente hemos y darnos una idea de cómo crear los programas para que cuenten con un eficiente funcionamiento. Por esto se puede decir que se cumplió con el objetivo de la práctica.

Referencias

- “Metodología de la programación.” Osvaldo Cairó, tercera edición, México D.F., Alfaomega 2005.
- “Metodología de la programación a través de pseudocódigo.” Miguel Ángel Rodríguez Almeida, primera edición, McGraw Hill