



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Alejandro Esteban Pimentel Alarcón

Asignatura: Fundamentos de Programación

Grupo: 135

No de Práctica(s): 3

Integrante(s): Torres Alcántara Alan Eliezer

*No. de Equipo de
cómputo empleado:* Tailandia 47

No. de Lista o Brigada: 9032

Semestre: 2020 - 1

Fecha de entrega: 26/08/2019

Observaciones: Identificaste bien los casos en los que las precondiciones no se cumplen, pero debes hacer la actividad de comprobación completa.

Te recuerdo que todo buen reporte escrito cuenta con introducción y conclusiones, o secciones equivalentes

CALIFICACIÓN: 9

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.

Algoritmo

Un algoritmo es un conjunto ordenado e inequívoco de pasos ejecutables que definen un proceso finito.

Ciclo de vida del software



Actividad:

Explicar las precondiciones y el conjunto de salidas de los algoritmos para:

- Pescar
- Lavarse las manos
- Cambiar una llanta
- Convertir un número binario a decimal

PESCAR

Precondiciones

1. Estar en un río, lago o en el mar abierto en donde podamos encontrar peces vivos
2. Tener una caña o red para pescar
3. Una persona que conozca la manera de pescar y que pueda hacerlo
4. Llevar carnada, cuerda, gancho de repuesto y un contenedor para los peces que se vayan a pescar

Salidas

1. Ningún pez pescado
2. Un pez pescado o más de un pez pescado
3. Se cayó la persona al agua por pescar

Los errores no se toman en cuenta como las salidas deseadas de un algoritmo

LAVARSE LAS MANOS

Precondiciones

1. Que la persona tenga manos
2. Estar en un lugar en donde se pueda enjuagar las manos
3. El agua para enjuagarse debe de estar limpia
4. Usar un jabón (Opcional)

Salidas

1. No se pudo lavar las manos porque no se cumplió uno de los prerequisites
2. Se lavo la persona las manos solo con agua
3. Se lavo la persona las manos con jabón

CAMBIAR UNA LLANTA (En automóvil)

Precondiciones

1. Que la persona tenga la capacidad para cambiar una llanta
2. Conocer el uso de herramientas básicas de mecánica, por ejemplo, cómo usar un gato hidráulico y una llave.
3. Tener las herramientas básicas de mecánica para autos
4. Conocer la forma en la que se debe cambiar una llanta
5. El auto debe estar estacionado
6. Debe de haber un auto al que se le pueda cambiar la llanta

Salidas

1. Una llanta cambiada
2. De dos a más llantas cambiadas
3. No se cambió ninguna llanta por no cumplirse los prerequisites mencionados anteriormente

CONVERTIR UN NUMERO BINARIO A DECIMAL

Precondiciones

1. El número por convertir debe estar en base dos
2. El número debe ser positivo en binario
3. Conocer la conversión de binario a decimal
4. Saber los valores de las potencias en base dos
5. Conocer las equivalencias de base dos al sistema decimal

Salidas

1. Un número decimal entero
2. Un número decimal con fracción
3. Ningún número por no haber cumplido los prerequisites

Actividad:

Desarrollar los algoritmos para:

- Determinar si un número es positivo o negativo
- Obtener el mayor de dos números diferentes
- Obtener el factorial de un número

DETERMINAR SI UN NÚMERO ES POSITIVO O NEGATIVO

Algoritmo

1. Tenemos el número entero “a”
2. Comparamos el número “a” con el número cero
3. Si el número “a” es mayor o igual que cero, podemos determinar que es un número positivo
4. Si el número “a” es menor que cero, podemos determinar que el número es negativo

OBTENER EL MAYOR DE DOS NÚMEROS DIFERENTES

Algoritmo

1. Tenemos solo los números “a” y “b” que pertenecen a los números enteros
2. Comparamos los números “a” y “b” entre sí
3. Si los dos números son iguales, tenemos que regresar al inicio y cambiar el número “a” o “b”
4. Si el número “a” es mayor que “b”, obtenemos que el número mayor es “a”
5. Si el número “a” es menor que “b”, obtenemos que el número mayor es “b”

OBTENER EL FACTORIAL DE UN NÚMERO

1. Tenemos el número “a” positivo
2. Si “a” es igual a cero, su factorial es uno
3. Multiplicaremos el número “a” desde el número 1
4. Si “a” es igual a uno, su factorial será uno

5. Para sacar la factorial de los números debemos multiplicar siguiendo la siguiente regla donde: $a! = 1 * (1+1) * (1+2) \dots * (a-1) * (a)$
Es decir, desde uno multiplicaremos el número por el siguiente número, por el número siguiente hasta llegar a multiplicar por el valor de “a”
6. Luego obtendremos el factorial de “a” al hacer la multiplicación correspondiente

Verificar sus algoritmos anteriores, al "ejecutarlos" paso a paso con los siguientes valores:

- 54 , -9, -14, 8, 0
- (4,5), (-9,16), (127,8+4i) , (7,m)
- 5 , 9 , 0 , -3

- ❖ En el caso de determinar si un número es positivo o negativo con los valores 54, -9, -14, 8 y 0, para todos los casos aplica ya que el algoritmo acepta a todos los números enteros.
- ❖ En el caso de obtener el mayor de dos números diferentes para los valores (4,5), (-9,16), (127,8+4i) y (7,m), solo se aplica para las parejas de los números (4,5) y (-9,16), en los demás casos no se puede desarrollar la comparación ya que el algoritmo no contiene variables con valores de números asignados para “i” como es en el caso de (8 + 4i), y para “m” como es el caso para (7,m).
- ❖ En el caso de obtener la factorial de un número con los valores 5, 9, 0, -3, tenemos que el número “-3” no aplica en el algoritmo ya que este solo se utiliza para números positivos, en el caso de “0” si aplica por la regla extra que colocamos en el algoritmo que dice: Si “a” es igual a cero, su factorial es uno.

Actividad:

Desarrollar algoritmos propios de un procesador (asignando registros genéricos) para:

- Cambiar el signo de un número binario
- Hacer una suma larga binaria

CAMBIAR EL SIGNO DE UN NÚMERO BINARIO

1. Disco duro: Manda registros “x” y “z” a la RAM que solicito Control
2. Control: De la RAM extrae los registros negativos “x” y “z” y los manda a los registros del procesador vacíos en lista de espera
3. Control: Manda “x” y “z” a ALU para cambiar el signo de bit negativo 1 a el signo de bit positivo 0
4. ALU: Recibe “x” y “z” y aplica el comando NOT después del primer número uno del lado derecho yendo en el orden de derecha a izquierda
5. ALU: Manda registros “x” y “z” a Control
6. Control: Recibe “x” y “z” y los incorpora a los lugares principales del procesador en espera para mandar a la RAM
7. RAM: Recibe los registros “x” y “z” y los incorpora en el disco duro
8. Disco duro: Recibe “x” y “z” de la RAM y los almacena

HACER UNA SUMA LARGA BINARIA

1. Disco duro: Manda registros “a”, “b”, “c”, “d”, “e” y “f” a la RAM que solicito Control
2. Control: De la RAM extrae los registros “a”, “b”, “c”, “d”, “e” y “f” y los manda a los registros del procesador vacíos en lista de espera
3. Control: Manda “a”, “b”, “c”, “d”, “e” y “f” a ALU para cambiar los números negativos a positivos si es que fuera el caso, si no saltar al paso 6
4. ALU: Recibe de “a”, “b”, “c”, “d”, “e” y “f” solo los números que se encuentren negativos, es decir con el bit de signo 1 y aplica el comando NOT a cada registro después del primer número uno del lado derecho yendo en el orden de derecha a izquierda
5. ALU: Manda los registros ya cambiados de signo a Control
6. Control: Manda “a”, “b”, “c”, “d”, “e” y “f” a ALU para hacer una suma binaria
7. ALU: Recibe “a”, “b”, “c”, “d”, “e” y “f” y aplica el comando XOR a los registros de dos en dos hasta llegar a un solo resultado en el orden de derecha a izquierda
8. ALU: Manda solo un registro llamado “z”, que es la suma de los registros “a”, “b”, “c”, “d”, “e” y “f” a Control
9. Control: Recibe “z” y lo incorpora a un lugar principal del procesador en lista de espera para mandar a la RAM
10. RAM: Recibe el registro “z” y lo incorpora en el disco duro
11. Disco duro: Recibe “z” de la RAM y lo almacena