|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Ing. Claudia Rodríguez Espino |
| *Asignatura:* | Fundamentos de Programación |
| *Grupo:* | 1102 |
| *No de Práctica(s):* | 04 |
| *Integrante(s):* | Alejandro Meneses Mercado |
|  |  |
|  |  |
| *Semestre:* | 2018-1 |
| *Fecha de entrega:* | 08 de septiembre del 2017 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

Práctica 04: Diagramas de flujo

Objetivo:

Elaborar diagramas de flujo que representen soluciones algorítmicas vistas como una serie de acciones que comprendan un proceso.

Desarrollo:

Primero vimos lo que es un diagrama de flujo, el cual es la representación gráfica de un algoritmo, también vimos lo importantes que son porque gracias a estos podemos codificar un programa en algún lenguaje de programación.

Luego vimos las formas del diagrama de fujo, es decir símbolos que nos permiten encontrar la solución a un problema de manera gráfica.

El primer simbolo es un óvalo el cual quiere decir inicio y fin, después vimos las líneas verticales u horizontales y estas indican la direccion el flujo del programa, aparte estas líneas siempre deben ir conectadas a un símbolo.

Vimos que el diagrama de flujo debe estar construido de arriba hacia abajo y de izquierda a derecha; que a cada símbolo solo le puede llegar una línea de dirección de flujo.

Otro símbolo que vimos es el de un rectangulo que significa proceso y es donde se expresan todas las operaciones, etc.

También vimos el símbolo de decisión el cual es un rombo y donde se valida una condición y toma un camino u otro.

Vimos el símbolo de escritura el cual imprime los resultados, otro símbolo es el conector, que es un círculo y nos indica que hay una conexión dentro de la misma página.

Por último vimos el símbolo de decisión multiple que es un hexagono y que almacena un selctor que determina la rama por la que sigue el flujo.

Después vimos las estructuras de control de flujo que permiten la ejecución condicional y la repetición de un conjunto de instrucciones,

Son 3: secuencial, condicional y repetitivas o iterativas

Vimos que las estructuras de control secuenciales son las sentencias o declaraciones que se realizan una a continuación de otra en el orden en el que están escritas.

También vimos que las estructuras de control condicionales permiten evaluar una condición lógica, es decir, una condición que puede ser verdadera o falsa y dependiendo del resultado se realiza uno u otro flujo de instrucciones.

A continuación vimos que dentro de las estructuras de control condicionales se encuentran derivaciones, la mas simple de ellas se conoce como SI (IF), donde se evalúa la condición lógica y si se cumple se ejecutan las instrucciones, si no se cumple la condición se continúa con el flujo normal.

La siguente estructura de control condicional que vimos fue la de: SI-DE LO CONTRARIO (IF-ELSE), donde se evalúa la expresión lógica y si se cumple se ejecutan las instrucciones, si no se cumple la condición se ejcutan otras instrucciones y al final se continúa con el flujo normal.

La última estructura de control condicional fue la de SELECCIONAR CASO, donde se valida el valor de la variable que esta en el hexágono y comprueba si es igual al valor que está definido en cada caso, que son las líneas que emanan del hexágono y si la variable no tiene el valor de algún caso se va a la instrucción por defecto.

Déspues vimos las estructuras de control de flujo iterativas o repetitivas, las cuales permiten ejecutar una serie de instrucciones mientras se cumpla la expresión lógica.

Vimos que existen dos tipos de estructuras de control de flujo iterativas o repetitivas: MIENTRAS y HACER- MIENTRAS.

Después vimos que la estructura MIENTRAS primero valida la condición y si ésta es verdadera procede a ejecutar el bloque de instrucciones de la estructura, de lo contrario rompe el ciclo y continúa el flujo normal del programa.

A continuación vimos que la estructura HACER-MIENTRAS primero ejecuta las instrucciones descritas en la estructura y al final valida la expresión lógica. Si la condición se cumple vuelve a ejecutar las instrucciones de la estructura, de lo contrario rompe el ciclo y sigue el flujo del algoritmo.

Por último vimos lo que cuando la solución de un problema es muy compleja se suele ocupar el diseño descendente. Este diseño implica la división de un problema en varios subprocesos más sencillos que juntos forman la solución completa y a estos subprocesos se les llaman módulos o funciones.

Vimos que las funciones estan constituidas por un identificador de función, de cero a n parámetros de entrada y un valor de retorno, el nombre con el que se llama a la función es: nomFun

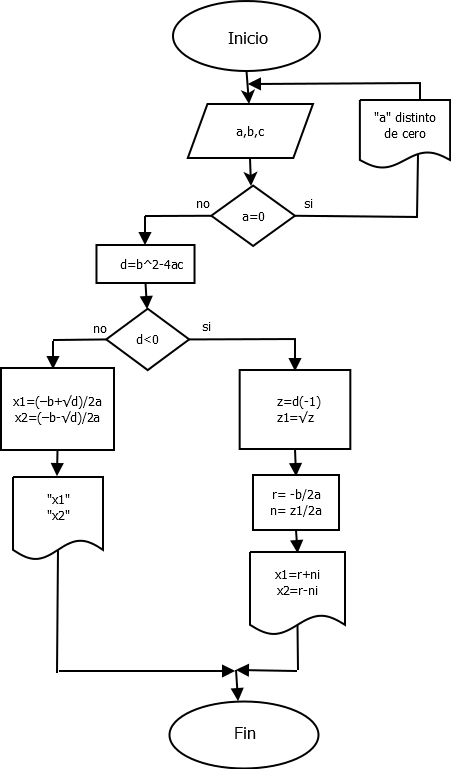
Déspues como actividad teníamos que hacer 3 algoritmos con sus respectivos diagramas de flujo, los cuales se encuantran a continuación:

Ejercicio 1

PROBLEMA: Resolver la fórmula general

Algoritmo:

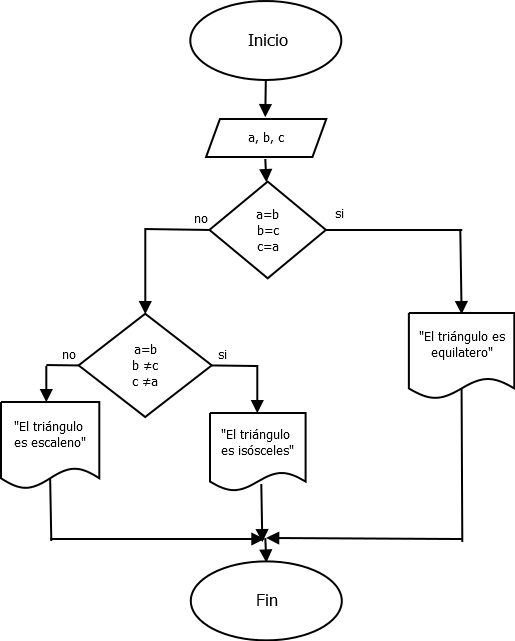
1. Inicio
2. Pedir un valor para “a”. Mostrar “No debe ser cero”.
3. Si a=0 mandar mensaje “No se puede porque crea una indeterminacion” y regresar al paso 2; en caso continúa al paso 4.
4. Pedir un valor para “b” y “c”.
5. Realizar la operación
6. Si d<0, entonces realizar la operación , hacer la operación r= , n= , imprimir x1=r+ni y x2=r-ni, en caso contrario ir al paso 7
7. Realizar la operación x1= , x2=, imprimir x1 y x2
8. Fin.



Ejercicio 2

PROBLEMA: Formación de triangulos

Algoritmo:

1. Inicio
2. Pedir un valor para “a”, “b” y “c”
3. Si a=b , b=c y c=a ir al paso 3.1, en caso contrario ir al paso 4
   1. Imprimir “El triángulo es equilátero”
4. Si a=b, b ≠c y c ≠a ir al paso 4.1, en caso contrario imprimir “El triangulo es escaleno”
   1. Imprimir “El triángulo es isósceles”
5. Fin

Ejercicio 3

PROBLEMA: Suma de 2 números

Algoritmo:

1.Inicio

2.Pedir un valor para “a”, “b” y “c”

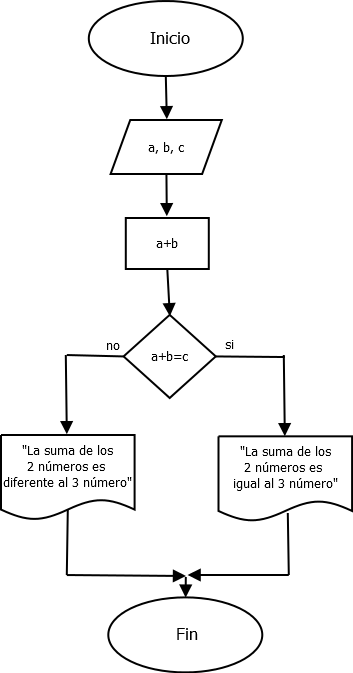
3. Realizar la operación a+b

4. Si a+b=c ir al paso 4.1, en caso contrario ir al paso 5

4.1 Imprimir “La suma de los 2 números es igual al 3 número”

5. Imprimir “La suma de los 2 números es diferente al 3 número”

6.Fin



Conclusiones:

En esta práctica retomamos el concepto de algoritmo pero ahora de una manera gráfica para así poder crear un diagrama de flujo, está práctica nos servira mucho ya que gracias a los diagramas de flujo podremos codificar nuestro programa en un lenguaje de programación mas facil.