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

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Rodriguez Espino Claudia |
| *Asignatura:* | Fundamentos de Programacion |
| *Grupo:* | 1102 |
| *No de Práctica(s):* | Practica numero 3 |
| *Integrante(s):* | Gutierrez Orozpe Luis Fernando |
|  |  |
|  |  |
| *Semestre:* | 1er Semestre. |
| *Fecha de entrega:* | 3 de Septiembre de 2017 |
| *Obervaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

OBJETIVO

Elaborar algoritmos correctos y eficientes en la solución de problemas siguiendo las etapas de Analisis y Diseño pertenecientes al Ciclo de vida del software.

ACTIVIDADES

Durante la practica pudimos conocer más acerca de los problemas informáticos los cuales se definen como el conjunto de instancias al cual corresponde un conjunto de soluciones. Para poder solucionar estos problemas se necesita recurrir a la Ingeniería de Software que se define como “La aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del Software”.

La ingeniería de Software provee métodos que indican como generar software. Estos métodos abarcan una amplia gama de tareas las cuales son:

* Planeación y estimación del proyecto.
* Análisis de requerimientos del sistema y software.
* Diseño de la estructura de datos, la arquitectura del programa y el procedimiento algorítmico.
* Codificación
* Pruebas y mantenimiento (validación y verificación).

Así mismo, dentro de la practica nos dimos a la tarea de conocer las diferentes etapas o procesos que se deben seguir para la solución de problemas.

ANALISIS

Dentro de esta etapa, es importante conocer lo que busca el cliente, es decir, la solución al problema, por lo tanto, debemos identificar el conjunto de datos de entrada (que son la serie de objetos o cosas con las que cuento) y el conjunto de salida (que son la serie de objetos o cosas que se deberán obtener a partir de los datos e entrada). A lo que une a los datos de entrada, con los datos de salida se le conoce como proceso, que es el que se encarga de manejar los datos de entrada para convertirlos en los datos de salida.

**ALGORITMO**

Una vez conociendo esta etapa del análisis de los datos y haber reconocido lo que el cliente nos ha solicitado, se procede a generar o diseñar el proceso de solución, es decir, se genera el algoritmo

Un algoritmo se define como el conjunto de estrategias o pasos a seguir con la finalidad de resolver un problema o llegar a una solución específica, por lo tanto el algoritmo es la parte más importante de las ciencias de la computación debido a que este puede ser creado de manera independiente tanto del lenguaje como de las características físicas del equipo que lo va a ejecutar.

Un buen algoritmo debe ser correcto (cumplir con el objetivo) y eficiente (realizarlo en el menor tiempo posible, además de ser entendible para cualquier persona.

**PRUEBA DE ESCRITORIO**

Para validar los datos que hemos introducido en nuestro algoritmo es necesario realizar una prueba de escritorio. Esta prueba es una matriz formada por los valores que adquiere cada una de las variables del programa en cada Iteración. La iteración es el número de veces que se ejecuta un código y permite ver los valores que van adquiriendo las variables en cada repetición.

A continuación se anexaran una serie de problemas en las que presentaremos su análisis de datos de entrada y de salida, así como de su proceso. El algoritmo que debe seguir para poder llegar a la solución deseada y la prueba de escritorio que valide que nuestras variables son correctas para lograr los datos de salida o solución que esperamos.

PROBLEMA 1

PROBLEMA: Realizar la formula general de manera que las raíces que nos de la ecuación sean tanto imaginarias como reales.

RESTRICCIONES: La variable “a” no debe ser cero

DATOS DE ENTRADA: Número real.

DATOS DE SALIDA: Definir si es una raíz real o imaginaia.

ALGORITMO

1. INICIO
2. Pedir el valor para “a” que sea diferente de cero.
3. Si “a”=0 mandar mensaje “No es posible porque crea una indeterminación” y regresar al paso 2; en caso contrario continuar al paso 4.
4. Pedir un valor para “b”
5. Pedir un valor para “c”
6. Realizar operación b2 – 4ac = d
7. Si d < 0, entonces realizar operación = z, hacer la operación **r = - b / 2a**, **n=z/2a** imprimir **x1= r + ni** y **x2= r - ni**, en caso contrario ir al paso 8.
8. X1= x2=
9. Mostrar x1 y x2
10. Fin

PRUEBA DE ESCRITORIO

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **a** | **b** | **c** | **d** | **z** | **r** | **n** | **X1** | **X2** |
| 0 | 4 | 5 |  |  |  |  | Indeterminacion | Indeterminacion |
| **a** | **b** | **c** | **d** | **z** | **r** | **n** | **X1** | **X2** |
| 3 | 5 | 2 | 49 |  |  |  | 1/3 | -2 |

En el ejemplo anterior de la prueba de escritorio, se cumplen los primeros pasos. Primero se cumple que si el valor de “a” es igual a 0, no se permite hacer ninguna operación y se manda el mensaje de que el resultado será una indeterminación, por lo tanto no se puede realizar y debe regresar al paso anterior. En el segundo ejemplo, una vez que se cumple la condición de que “a” no debe ser igual a 0, se empieza con las operaciones. De la misma forma, se cumple la condición de que si el valor de la variable “d” es mayor que 0, pasa directamente al paso número 8. Finalmente al realizar las operaciones del paso 8, se obtienen las raíces de la ecuación.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **a** | **b** | **c** | **d** | **z** | **r** | **n** | **X1** | **X1** |
| 2 | -1 | 3 | -23 |  | 1/4 |  |  |  |

En el ejemplo anterior, debido a que se cumple la condición de que la variable “d” es menor que 0 se realizan las operaciones de las variables z, r y n para que asi podamos obtener como resultado raíces imaginarias.

PROBLEMA 2

PROBLEMA: Obtener el factorial de un número.

CONDICION: Que el número ingresado sea mayor que 0 y menor que 11

DATOS DE ENTRADA: Número Real

DATOS DE SALIDA: Factorial del número real ingresado

ALGORITMO

1. INICIO
2. Contador = 0, Factorial 1
3. Pedir numero (m)
4. Si 0 < m < 11 ir al paso 5; en caso contrario regresar al paso 2
5. Mientras contador sea menor o igual que m:

Factorial = factorial \* contador

Contador = contador + 1

1. En caso contrario ir al paso 7
2. Fin

PRUEBA DE ESCRITORIO

|  |  |  |  |
| --- | --- | --- | --- |
| m | contador | factorial | salida |
| 0 |  |  | Regresar al paso 2, el número no debe ser 0 |

En el ejemplo anterior, debido a que el algoritmo nos pide que el numero ingresado sea mayor que cero, en la salida nos regresa al paso 2 para volver a ingresar el número.

|  |  |  |  |
| --- | --- | --- | --- |
| m | contador | factorial | salida |
| 1 | 0 | 1 | El factorial es 1 |

En el ejemplo anterior, como ya se cumple la condición de que el numero m sea mayor que cero, se realiza el proceso y como salida obtenemos que el factorial de 1 es igual a 1

|  |  |  |  |
| --- | --- | --- | --- |
| m | contador | factorial | salida |
| 5 | 1 | 1 |  |
|  | 2 | 2 |  |
|  | 3 | 6 |  |
|  | 4 | 24 |  |
|  | 5 | 120 | El factorial es 120 |

En el ejemplo anterior, se cumple la condición de que el numero debe ser mayor que 0 pero menor que 11 para que se lleve a cabo el proceso. Al realizarse el proceso obtenemos que el factorial de 5 es 120

PROBLEMA 3

PROBLEMA: RESOLVER LAS ECUACIONES DADAS

CONDICION: Si y < 2 resolver x = y2 + 4y – 25: Si y > 2 resolver x = 4y2 – 3y + 0: Si y = 2 no hay solución.

DATOS DE ENTRADA: Valor de “y”

DATOS DE SALIDA: Mostrar el valor de “x”

ALGORITMO

1. INICIO
2. Pedir el valor de “y”
3. Comparar y < 2 si se cumple ir al paso 4, sino ir al paso 3.1

3-1. Comparar si y > 2 si se cumple ir al paso 3-2, sino ir al paso 3.3

3-2. Resolver x = 4y2 – 3y + 0 e ir al paso 5

3-3. No hay solución para y =2

1. Resolver x = y2 + 4y – 25 e ir al paso 5
2. Mostrar “x”
3. Fin

PRUEBA DE ESCRITORIO

|  |  |  |
| --- | --- | --- |
| Y | x = y2 + 4y – 25 | x = 4y2 – 3y + 0 |
| 2 | No hay solucion | No hay solucion |

En el ejemplo anterior, debido a que en el algoritmo se especifica que si el valor de “y” es igual a 2 se debe escribir el mensaje de que no hay solución y se termina con el programa.

|  |  |  |  |
| --- | --- | --- | --- |
| y | x = y2 + 4y – 25 | x = 4y2 – 3y + 0 | x |
| 1 | -20 |  | -20 |
| -10 | 35 |  | 35 |
| 3 |  | 370 | 370 |
| 10 |  | 27 | 27 |

En el ejemplo anterior, se cumple con la condición de que el valor de “y” no debe ser igual a 2.

Por lo tanto, en el caso de arriba, como el valor de “y” es menor que 2, se realiza la ecuación correspondiente a esa condición y se obtienen los valores de salida de “x” (-20 y 35)

Por otro lado, como se cumple la condición de que “y” es mayor que 2, se realiza la otra ecuación correspondiente a esa condición y se obtienen los valores de salida de “x” (370 y 27).

PROBLEMA 4

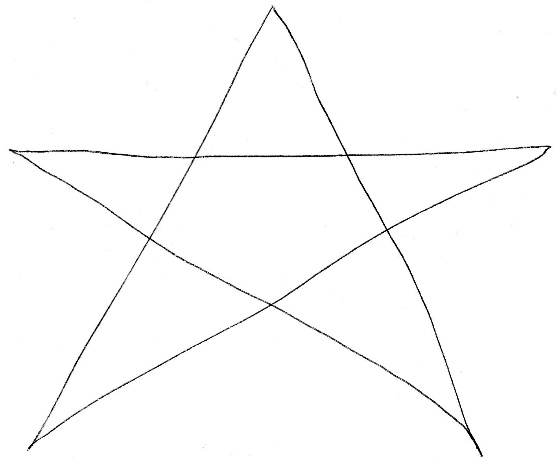
PROBLEMA: Seguir el algoritmo para obtener una figura

ENTRADA: Hoja tamaño carta en limpio, regla y lápiz.

SALIDA: Figura correcta

ALGORITMO

1. Dibuja un 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 de 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 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.



Para la actividad anterior se necesitó que siguiéramos los pasos del algoritmo para tener como resultado final o de salida, el dibujo que anexé.

CONCLUSION

Pienso que esta práctica es de gran importancia ya que gracias al conocimiento de los algoritmos y de las pruebas de escritorio es que podemos iniciar con las bases de programación, aunque a muchos alumnos les resulte aburrido ver esta parte de los algoritmos y pruebas de escritorio (incluyéndome) debido a que ya sabemos programar, creo que sin esta parte no podríamos realizar una solución o programa que sea optimo

Como lo vimos anteriormente, para poder resolver un problema, es necesario que entremos en ese análisis acerca de que es lo que el cliente me está pidiendo y que es lo que le voy a dar finalmente, por esta razón, aunque a muchos nos resulte aburrida esta parte, creo que es primordial para que a futuro se nos facilite la resolución de problemas y podamos entregar trabajos de calidad cuando ya estemos en la etapa laboral.