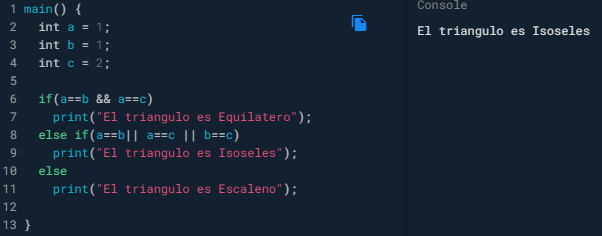
**1. Responda las preguntas a final del capítulo: (1-8, 11-18)**

**17.1. Myers [MYE79] usa el siguiente programa como auto-comprobación de su capacidad para especificar una prueba adecuada: un programa lee tres valores enteros. Los tres valores se interpretan como representación de la longitud de los tres lados de un triángulo. El programa imprime un mensaje indicando si el triángulo es escaleno, isósceles o equilátero. Desarrolle un conjunto de casos de prueba que considere que probará de forma adecuada este programa.**

|  |  |  |
| --- | --- | --- |
| Casos | Resultado obtenido | Resultado esperado |
| No ingresar valores para ninguna de los tres lados. | Imprime: “Debe de digitalizar la longitud de los tres lados” | Muestra el mensaje indicado anteriormente. |
| Solo leer el primer lado del triángulo. | Imprime: “Debe indicar el valor de los lados 2 y 3” | Solicita asignación de valores de los lados faltantes |
| Solo leer el segundo lado del triángulo. | Imprime: “Debe indicar el valor de los lados 1 y 3” | Solicita asignación de valores de los lados faltantes |
| Solo leer el tercer lado del triángulo. | Imprime: “Debe indicar el valor de los lados 1 y 2” | Solicita asignación de valores de los lados faltantes |
| Solo leer dos lados del triángulo. | Imprime: “Falta un lado del triángulo”. | Solicita asignación del valor del lado faltante |
| Digitar el mismo valor para los tres lados. | Imprime: “Es un triángulo equilátero” | Muestra el mensaje indicado anteriormente. |
| Digitar dos valores iguales y uno diferente. | Retornar mensaje: “Es un triángulo isósceles” | Muestra el mensaje indicado anteriormente. |
| Digitar tres valores diferentes. | Imprime: “Es un triángulo escaleno”. | Muestra el mensaje indicado anteriormente. |

**17.2. Diseñe e implemente el programa especificado en el Problema 17.1 (con tratamiento de errores cuando sea necesario). Obtenga un grafo de flujo para el programa y aplique la prueba del camino básico para desarrollar casos de prueba que garanticen la comprobación de todas las sentencias del programa. Ejecute los casos y muestre sus resultados.**

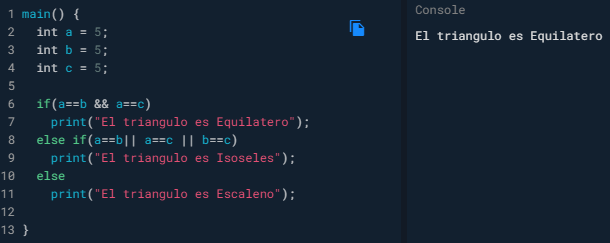


**17.3. ¿Se le ocurren algunos objetivos de prueba adicionales que no se hayan mencionado en la Sección 17.1.1?**

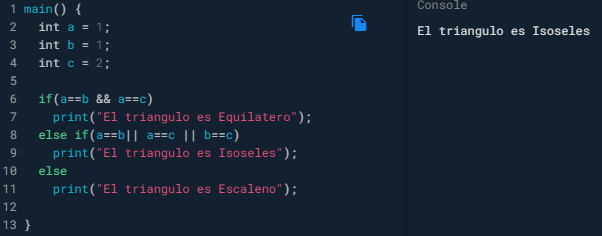
Se debe validar que no se ingrese valores no numéricos como letras, caracteres especiales, los valores sean mayores a 0, etc.

**17.4 Aplique la técnica de prueba del camino básico a cualquiera de los programas que haya implementado en los Problemas 17.4 a 17.11.**

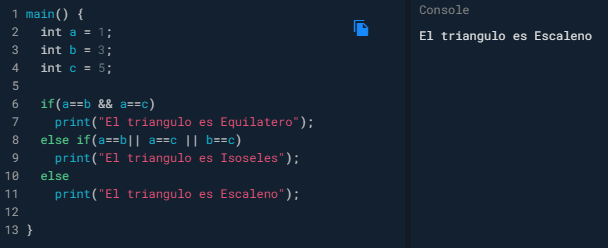
**Primer camino:**



**Segundo camino:**

****

**Tercer camino:**

****

**17.7. Use el enfoque de prueba de condiciones descrito en la Sección 17.5.1 para diseñar un conjunto de casos de prueba para el programa creado en el Problema 17.2.**

**Método que utiliza las condiciones para saber el tipo de triángulo que es, con los valores de cada parámetro.**

Se tomaran como referencia diversos parámetros para poder dar respuesta acerca de qué tipo de triangulo se están ingresando los lados, se utilizaran variables booleanas y se mostrara el resultado del triángulo cuya condición se cumpla.

Un ejemplo de implementación sería la siguiente tabla:

|  |  |  |
| --- | --- | --- |
| *Condición lógica* | *Condición a probar* | *Parámetros de test* |
| *Lado1==lado2* | True | Lado1=1,lado2=1 |
| *Lado1==lado2* | False | Lado1=2,lado2=1 |
| *Lado1==lado3* | True | Lado1=1,lado3=1 |
| *Lado1==lado3* | False | Lado1=2,lado3=1 |
| *Lado1==lado2 and lado1== lado3* | True | Lado1=1,lado2=1,lado3=1 |
| *Lado1==lado2 and lado1==lado3* | False | Lado1=1,lado2=1,lado3=2 |

**17.8. Mediante el enfoque de prueba de flujo de datos descrito en la Sección 17.5.2, crea una lista de cadenas de definición-uso para el programa creado en el Problema 17.2.**

Podríamos determinar la misma por medio del siguiente cuadro de explicación de código y diagramación de flujo:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Línea*** | **Definición** | **P-Use** | **C-Use** |
| *string Identificación\_Triangulo(int lado1,int lado2, int lado3)* | Lado1, lado2, lado3 |  |  |
| *if (lado1 <= 0 && lado2 <= 0 && lado3 <= 0)* | return "Debe digitar la longitud de los tres lados" | Lado1, lado2, lado3 |  |
| *if (lado1 > 0 && lado2 <= 0 && lado3 <= 0)* | return "Debe indicar el valor de los lados 2 y 3" | Lado1, lado2, lado3 |  |
| *if (lado2 > 0 && lado1 <= 0 && lado3 <= 0)* | return "Debe indicar el valor de los lados 1 y 3" | Lado1, lado2, lado3 |  |
| *if (lado3 > 0 && lado1 <= 0 && lado2 <= 0)* | return "Debe indicar el valor de los lados 1 y 2" | Lado1, lado2, lado3 |  |
| *if ((lado1 > 0 && lado2 > 0 && lado3 <= 0) ||*  *(lado1 > 0 && lado3 > 0 && lado2 <= 0) ||*  *(lado2 > 0 && lado3 > 0 && lado1 <= 0))* | return "Falta un lado del triángulo" | Lado1, lado2, lado3 |  |
| *if (lado1 == lado2 && lado1 == lado3)* | return "Es un triángulo equilátero" | Lado1, lado2, lado3 |  |
| *Else if ((lado1 == lado2 && lado1 != lado3) ||*  *(lado1 == lado3 && lado1 != lado2) ||*  *(lado2 == lado3 && lado2 != lado1))* | return "Es un triángulo isósceles" | Lado1, lado2, lado3 |  |
| *else if (lado1 != lado2 && lado1 != 3)* | return "Es un triangulo escaleno" | Lado1, lado2, lado3 |  |

**17.12. ¿Podría una prueba exhaustiva (incluso si fuera posible para pequeños programas) garantizar que un programa esté al 100 por 100 correcto?**

No, debido a que puede que no se contemplara algún parámetro en la prueba y el cliente pueda notar que existe una brecha en el funcionamiento, los requerimientos del cliente pueden cambiar ya sea porque no se dio a entender o no se interpretó con exactitud el requerimiento del cliente o simplemente porque lo procesos evolucionan con el tiempo a medida que la empresa crece.

**17.13. Usando el método de la partición equivalente, obtenga un conjunto de casos de prueba para el sistema *HogarSeguro* descrito anteriormente en el libro.**

Al utilizar el método de la partición equivalente, los casos de prueba para el sistema de Hogar Seguro son:

* Clave de un mínimo de 8 caracteres, por ejemplo: 2183, 8213, 2318, 1823.
* Entre las condiciones están:
  + Clave mínima de 8 caracteres.
  + Sin una secuencia de números idénticos.
  + Alternación de caracteres y símbolos especiales.
  + Utilización de mayúsculas y minúsculas.

**17.15. Investigue un poco y escriba un breve documento sobre el mecanismo de generación de tablas ortogonales para la prueba de datos.**

Debemos enfocarnos en la utilidad de los criterios de diseños los cuales indican la comprobación y la optimización de la entidad de prueba que han de ser utilizadas en el sistema o las tablas que se encuentran disponibles para la identificar el número de pruebas que se realizaran.

El personal encargado de las pruebas debe tener una idea clara de cómo debe funcionar y de la importancia de que este funcione correctamente,

**17.17. Investigue en un sistema Cliente/Servidor que le sea familiar. Desarrolle un conjunto de escenarios de usuario y genere un perfil operacional para el sistema.**

Un sistema cliente servidor facilita la distribución de las tareas entre los encargados de proveer los recursos necesarios (servidores) para manejar la carga de las peticiones o solicitudes generadas por los clientes, además de manejarlas brindan una respuesta.

**17.18. Pruebe un manual de usuario (o una facilidad de ayuda) de una aplicación que utilice frecuentemente. Encuentre al menos un error en la documentación.**

En la guía de Microsoft Teams se puede apreciar que especifica como programar reuniones esto se debe a que el manual no fue completado o actualizado.