EJERCICIO PRUEBAS DE CAJA BLANCA (Camino Básico)

1.- Dado el siguiente código:

Static void visualizarMedia (float x, float y){

❶

float resultado =0;

If ( (x<0) ❷ || (y<0) ❸ )

System.out.println (“X e Y deben ser positivos”); ❹

else {

resultado =(x+y)/2;

❺

System.out.println(“ la media es: “+ resultado);

}

} ❻

**Se pide:**

**1.- Realizar el grafo de flujo.**

❶

❷

❸ ❹

❺

❻

**2.- Realizar los caminos independientes y los casos de prueba según la siguiente tabla:**

|  |  |  |
| --- | --- | --- |
| Camino | Caso de prueba | Resultado esperado |
| 1-2-4-6 | X=-1,Y=2 | “X e Y deben ser positivos” |
| 1-2-3-4-6 | X=2,Y=-1 | “X e Y deben ser positivos” |
| 1-2-3-5-6 | X=2,Y=2 | 2 |

**3.- Calcular la complejidad ciclomática V(G)**

V(G) = ((7-6)+2) = 3

**4.- Documentar las pruebas siguiendo la plantilla de Pruebas de caja Blanca.**

|  |  |
| --- | --- |
| Columna | Instrucciones |
| Id | 1 |
| Caso de Prueba | Prueba con número negativo |
| Descripción | Se procede a probar el programa introduciendo una de las dos variables negativas. Sólo se ejecutará la primera parte del bucle. La variable “resultado” no debe llegar a mostrarse. |
| Fecha | 07/02/2019 |
| Funcionalidad / Característica | Media con la primera variable negativa. |
| Datos / Acciones de Entrada | Entradas:   * -1 * 2 |
| Resultado Esperado | “X e Y deben ser positivos” |
| Procedimientos especiales requeridos | Se debe introducir dos números. No se admiten caracteres ni espacios en blanco. |
| Dependencias con otros casos de Prueba | [Lista de los identificadores de los casos de prueba que deben ejecutarse antes de este caso. También puede incluir un sumario de la naturaleza de estas dependencias.] |
| **Información para el seguimiento** | |
| Resultado Obtenido | “X e Y deben ser positivos” |
| Estado | Exitoso |
| Última Fecha de Estado | 07/02/2019 |
| Observaciones |  |

|  |  |
| --- | --- |
| Columna | Instrucciones |
| Id | 2 |
| Caso de Prueba | Prueba con número negativo |
| Descripción | Se procede a probar el programa introduciendo una de las dos variables negativas. Sólo se ejecutará la primera parte del bucle. La variable “resultado” no debe llegar a mostrarse. |
| Fecha | 07/02/2019 |
| Funcionalidad / Característica | Media con lasegunda variable negativa. |
| Datos / Acciones de Entrada | Entradas:   * 2 * -1 |
| Resultado Esperado | “X e Y deben ser positivos” |
| Procedimientos especiales requeridos | Se debe introducir dos números. No se admiten caracteres ni espacios en blanco. |
| Dependencias con otros casos de Prueba | [Lista de los identificadores de los casos de prueba que deben ejecutarse antes de este caso. También puede incluir un sumario de la naturaleza de estas dependencias.] |
| **Información para el seguimiento** | |
| Resultado Obtenido | “X e Y deben ser positivos” |
| Estado | Exitoso |
| Última Fecha de Estado | 07/02/2019 |
| Observaciones |  |

|  |  |
| --- | --- |
| Columna | Instrucciones |
| Id | 3 |
| Caso de Prueba | Prueba con los dos números positivos. |
| Descripción | Se procede a probar el programa introduciendo una de las dos variables positivas. Sólo se ejecutará la segunda parte del bucle, almacenando en la variable “resulado”, el valor de la operación ejecutada. La variable “resultado” deberá mostrarse correctamente. |
| Fecha | 07/02/2019 |
| Funcionalidad / Característica | Media con las dos entradas positivas. |
| Datos / Acciones de Entrada | Entradas:   * 2 * 2 |
| Resultado Esperado | 2 |
| Procedimientos especiales requeridos | Se debe introducir dos números. No se admiten caracteres ni espacios en blanco. |
| Dependencias con otros casos de Prueba | [Lista de los identificadores de los casos de prueba que deben ejecutarse antes de este caso. También puede incluir un sumario de la naturaleza de estas dependencias.] |
| **Información para el seguimiento** | |
| Resultado Obtenido | 2 |
| Estado | Exitoso |
| Última Fecha de Estado | 07/02/2019 |
| Observaciones |  |