**Ejecutables**

**Lista de casos en los que fallan:**

1. **N tiene 0 dígitos**

|  |  |
| --- | --- |
| Entrada | Salida |
| 0  1  1 | <sin salida> |

|  |
| --- |
| Salida del ejecutable |
| A = B = C = D = E = F = G = I: “NO”  H: genera archivo salida.out vacío |

*Observaciones:* No se valida la condición de entrada N >= 1.

1. **M tiene 0 dígitos**

|  |  |
| --- | --- |
| Entrada | Salida |
| 1  1  0 | <sin salida> |

|  |
| --- |
| Salida de los ejecutables |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | A | B | C | D | E | F | G | H | I | | SI 2  1 2 | NO | NO | SI 1  1 | SI  2 1 2 | SI 2  1 2 | crashea | SI 2  1 2 | NO | |

*Observaciones:* No se valida la condición de entrada M >= N

1. **Entrada larga, con todos los dígitos en M**

|  |  |
| --- | --- |
| Entrada | Salida |
| 100  1234567890543210987643215987603215487609215437609815432609875132460978534210987652143760894325180976  10  1234567890 | SI 25  1 6 11 16 21 26 31 36 41 46 51 53 56 57 58 59 61 62 66 71 72 76 81 86 91 |

|  |
| --- |
| Salida de los ejecutables |
| |  |  | | --- | --- | | A | SI 25  1 6 11 16 21 26 31 36 41 46 51 53 56 57 58 59 61 62 66 71 72 76 81 86 91 | | B | SI 1  1 | | C | SI 2  1 11 | | D | SI 24  1 6 11 16 21 26 31 36 41 46 51 53 56 57 58 59 61 62 66 71 72 76 81 86 | | E | SI 25  1 6 11 16 21 26 31 36 41 46 51 53 56 57 58 59 61 62 66 71 72 76 81 86 91 | | F | SI 26  1 6 11 16 21 26 31 36 41 46 51 53 56 57 58 59 61 62 66 71 72 76 81 83 86 91 | | G | SI 10  1 11 21 31 41 51 61 71 81 91 | | H | SI 25  1 6 11 16 21 26 31 36 41 46 51 53 56 57 58 59 61 62 66 71 72 76 81 86 91 | | I | crashea | |

*Observaciones:* Ejecutable G tratado aparte.

\*Con las observaciones recopiladas hasta el momento, se plantean casos de prueba adicionales que prueben que las deducciones obtenidas hasta ahora sean correctas. Últimamente, se procederá a dar una conclusión final en cuanto al funcionamiento del programa. \*

**CASOS ADICIONALES:**

**.Conclusiones:**

Lo que está haciendo el ejecutable G es tomar la cantidad de dígitos **M** que tiene el valor escondido, y usar esa cantidad para recorrer el número con cantidad de dígitos **N**. En otras palabras, lo que hace es “partir” el número con **N** dígitos en partes de **M** dígitos y valida cada una de esas partes teniendo en cuenta que el número escondido puede tener los dígitos desordenados. Esto provoca que se saltee valores internos que son válidos, lo cual queda especialmente claro cuando se da un caso como **467467467** **(N=9)** siendo el número a buscar **467 (M=3)** ya nunca llega a tener en cuenta el valor **674 (el cual es válido)** debido a que no es un número que forme parte del recorrido de validación que el programa realiza, lo cual es erróneo.