Caso 01: enunciado

Descripción: se prueba que funcione correctamente con un caso aleatorio

|  |  |
| --- | --- |
| 01-Enunciado.in | 01-Enunciado.out |
| 4 2  1 4  0 2 4 3  2 0 5 2  4 5 0 1  3 2 1 0 | 3  2 4  3 4 |

Caso 02: circulo de 3

Descripción: se prueba que se encuentre el camino más óptimo ante 2 opciones para llegar a una ciudad

|  |  |
| --- | --- |
| 02-circulo3.in | 02-circulo3.out |
| 3 1  1  0 3 1  3 0 1  1 1 0 | 2  1 3  3 2 |

Caso 03: dos centrales y una ciudad

Descripción: se prueba que, al tener más de una central adyacente a una ciudad, se conecte a la de menor costo

|  |  |
| --- | --- |
| 03-2centrales1ciudad.in | 03-2centrales1ciudad.out |
| 3 2  1 3  0 1 0  1 0 2  0 2 0 | 1  1 2 |

Caso 04: ciudades alineadas

Descripción: se prueba que conecte a todas las ciudades cuando solo existe un camino posible

|  |  |
| --- | --- |
| 04-ciudadesAlineadas.in | 04-ciudadesAlineadas.out |
| 3 1  1  0 1 0  1 0 1  0 1 0 | 2  1 2  2 3 |

Caso 05: todas centrales

Descripción: se prueba que se valide correctamente cuando no es necesario tender líneas eléctricas

|  |  |
| --- | --- |
| 05-todasCentrales.in | 05-todasCentrales.out |
| 2 2  1 2  0 4  4 0 | 0 |

Caso 06: todas a una central

Descripción: se prueba que conecte correctamente las ciudades a la única central disponible

|  |  |
| --- | --- |
| 06-todasAunaCentral.in | 06-todasAunaCentral.out |
| 4 1  4  0 0 0 2  0 0 0 3  0 0 0 1  2 3 1 0 | 6  1 4  2 4  3 4 |

Caso 07: caso de estrés

Descripción: se prueba que el programa responda bien si todas las ciudades están conectadas entre sí con el mismo costo

|  |  |
| --- | --- |
| 07-estres100conexo.in | 07-estres100conexo.in |
|  |  |

Caso 08:

Descripción:

|  |  |
| --- | --- |
|  |  |
|  |  |

Caso 09:

Descripción:

|  |  |
| --- | --- |
|  |  |
|  |  |

Caso 10:

Descripción:

|  |  |
| --- | --- |
|  |  |
|  |  |

Caso 11:

Descripción:

|  |  |
| --- | --- |
|  |  |
|  |  |