Caso de prueba 01: Caso base.

Se busca que tome la cantidad de aplicaciones consecutiva más corta.

|  |  |
| --- | --- |
| 01CasoBase.IN | 01CasoBase.OUT |
| 10 100  42 2 50 10 1 50 30 24 18 23 | 3  6 |

Caso de prueba 02:

Se busca que encuentre las primeras dos aplicaciones.

|  |  |
| --- | --- |
| 02BusquedaPrimeros.IN | 02BusquedaPrimeros.OUT |
| 5 100  50 50 1 1 1 | 2  1 |

Caso de prueba 03: Búsqueda últimos.

Se busca que encuentre las últimas dos aplicaciones (que lea n y n-1 elementos).

|  |  |
| --- | --- |
| 03 BusquedaUltimos.IN | 03BusquedaUltimos.OUT |
| 5 100  1 1 1 50 50 | 2  4 |

Caso de prueba 04: Búsqueda mayor

Se busque que, si la cantidad de aplicaciones consecutivas es la misma, toma la que libera más espacio.

|  |  |
| --- | --- |
| 04BusquedaMayor.IN | 04BusquedaMayor.OUT |
| 10 40  20 20 1 2 3 11 30 1 2 3 | 2  6 |

Caso de prueba 05: Búsqueda igual

Se busque que, si la cantidad de aplicaciones consecutivas es la misma, y el espacio que libera también, libere las primeras aplicaciones.

|  |  |
| --- | --- |
| 05BusquedaIgual.IN | 05BusquedaIgual.OUT |
| 7 40  20 20 1 2 3 10 30 | 2  1 |

Caso de prueba 06: Memoria insuficiente

Se comprueba que, si no existe espacio suficiente, devuelva el mensaje correspondiente.

|  |  |
| --- | --- |
| 06MemoriaInsuficiente.IN | 06MemoriaInsuficiente.OUT |
| 7 1000  20 20 1 2 3 10 30 | MEMORIA INSUFICIENTE |

Caso de prueba 07: Cadena correcta

Se busque que, si, hay múltiples coincidencias (en cantidad), elija la correcta.

|  |  |
| --- | --- |
| 07CadenaCorrecta.IN | 07CadenaCorrecta.OUT |
| 12 40  20 20 1 2 3 11 30 1 2 3 39 20 | 2  11 |

Caso de prueba 08: Todas las apps

Se busca que elimine todas las aplicaciones para que la aplicación nueva pueda ser instalada.

|  |  |
| --- | --- |
| 08TodasLasApps.IN | 08TodasLasApps.OUT |
| 10 10  1 1 1 1 1 1 1 1 1 1 | 10  1 |