01\_ejemplo\_parcial\_1

Demuestra que jugador A gana con 5 sets de 2 juegos.

|  |  |
| --- | --- |
| 01\_ejemplo\_parcial\_1.in | 01\_ejemplo\_parcial\_1.out |
| 10 5 2 1  AAAAAAAAAA | 5 0 |

02\_ejemplo\_parcial\_2

Demuestra que se puede desempatar.

|  |  |
| --- | --- |
| 02\_ejemplo\_parcial\_2.in | 02\_ejemplo\_parcial\_2.out |
| 21 3 3 2  AABABBBABBBABABABBABB | 1 3 |

03\_unico\_set

Contempla caso de juego minimo, 1 juego -> 1 set: gana.

|  |  |
| --- | --- |
| 03\_unico\_set.in | 02\_ejemplo\_parcial\_2.out |
| 1 1 1 1  A | 1 0 |

04\_desempate\_al\_final

Se busca verificar que el programa desempata luego de una tanda de empates.

|  |  |
| --- | --- |
| 04\_desempate\_al\_final.in | 04\_desempate\_al\_final.out |
| 7 1 1 1  ABABABB | 0 1 |

05\_empate

El programa debe procesar empates (porque no se llega a la cantidad de sets).

|  |  |
| --- | --- |
| 05\_empate.in | 05\_empate.out |
| 8 5 2 1  AABBAABB | 2 2 |

06\_todos\_los\_games\_se\_ganan\_de\_a\_bloque

|  |  |
| --- | --- |
| 06\_todos\_los\_games\_se\_ganan\_de\_a\_bloque.in | 06\_todos\_los\_games\_se\_ganan\_de\_a\_bloque.out |
| 10 3 2 1  AABBAABBAA | 3 2 |

07\_gana\_A

|  |  |
| --- | --- |
| 07\_gana\_A.in | 07\_gana\_A.out |
| 9 3 2 4  ABAABAABA | 3 0 |

08\_ gana\_B

|  |  |
| --- | --- |
| 08\_ gana\_B.in | 08\_ gana\_B.out |
| 9 3 2 1  AABBABBABB | 0 3 |

09\_empate\_a\_cero

|  |  |
| --- | --- |
| 09\_ empate\_a\_cero.in | 09\_ empate\_a\_cero.out |
| 9 3 2 1  AABBABBABB | 0 0 |

10\_fatiga

|  |  |
| --- | --- |
| 10\_fatiga.in | 10\_fatiga.out |
| 105 10 3 3  ABABABABABABABABABABABABABABABABABABABABABABABABABABABABABABABABABABABABABABABABABABABABABABABABABABABABA | 0 0 |

**Complejidad computacional O(n)**

**No es necesario programa probador porque las salidas son únicas.**