**Caso 1. Probar que el programa considere los mayores importes consecutivos vendidos.**

**Premio.in**

3 //3 vendedoras

4 //V1

15

15

50

5

4 //V2

20

240

20

70

5 //V3

1

20

25

100

20

3 //N

**Premio.out**

2

4 350

El output esperado es que gane la vendedora 2 con 4 ventas consecutivas con un total de 350 pesos.

**Caso 2. Probar si el programa puede determinar correctamente cuando no se puede desempatar debido a que ninguna vendedora se sacó ventaja.**

**Premio.in**

2 //vendedoras

3 //v1

30

40

5

3 //v2

5

20

20

2 //N

**Premio.out**

“No se puede desempatar”

**Caso 3. Busca probar que el programa elimine correctamente a una vendedora para un determinado valor de N. No debería considerar de vuelta a esta vendedora eliminada para una próxima ronda (N+1).**

**Premio.in**

3 //vendedoras

3 //v1

10

10

5

3 //v2

10

11

2

3 //v3

10

11

1

2 //N

**Premio.out**

2

3 23

**Caso 4. Caso fatiga, busca ver si el algoritmo funciona bien cuando es sobreexigido, es decir, cuando son varias vendedoras, hay muchos empates consecutivos y el N inicial es el más bajo posible.**

**Premio.in**

6 //vendedoras

6 //v1

10

40

20

9

21

5

7 //v2

50

10

10

9

21

5

25

5 //v3

15

45

10

9

21

5 //v4

10

12

48

9

21

5 //v5

21

9

10

27

33

4 //v6

9

41

19

10

2 //N

**Premio.out**

2

7 130

**Caso 5. Intenta probar si el algoritmo contempla valores con decimales (flotantes)**

**Premio.in**

3 //Vendedoras

3 //V1

10

8

30

100

4 //V2

5.4

80.6

50

29.3

2 //V3

20.5

30

2 //N

**Premio.out**

2

2 130.6

**Caso 6. Intenta probar si el algoritmo es capaz de identificar errores en el input (valores alfabéticos)**

**Premio.in**

2

2

7c

15

3

a

65

4b

2

**Premio.out**

**“**Error en los datos de entrada, no son admisibles datos alfabéticos”

**Caso 7. Intenta probar si el algoritmo es capaz de identificar errores en el input (datos insuficientes)**

**Premio.in**

2

2

64

15

3

25

15

2

**Premio.out**

**“**Datos de entrada insuficientes”

**Caso 8. Intenta probar si el algoritmo identifica correctamente cuando se trata de un caso donde “no hay ganadoras” debido a que ninguna cumple con el N.**

**Premio.in**

2 //Vendedoras

3 //V1

47

84

27

4 //V2

94

24

15

421

5 //N

**Premio.out**

**“**No hay ganadoras”

**Caso 8. Intenta probar si el programa valida correctamente los datos de entrada (cantidad de vendedoras)**

**Premio.in**

130 //Vendedoras

3 //V1

47

84

27

4 //V2

94

24

15

421

3 //N

**Premio.out**

**“**Error en los datos de entrada. Las vendedoras pueden ser de 1 a 100”.

**Caso 9. Intenta probar si el programa valida correctamente los datos de entrada (cantidad de ventas de las vendedoras)**

**Premio.in**

2 //Vendedoras

1394 //V1

47

84

27

4 //V2

94

24

15

421

3 //N

**Premio.out**

**“**Error en los datos de entrada. Las ventas de cada vendedora pueden ser entre 1 y 1000”.

**Caso 10. Intenta probar si el programa valida correctamente los datos de entrada (importe de las ventas)**

**Premio.in**

2 //Vendedoras

3 //V1

47

84

27

4 //V2

94

9514

15

421

3 //N

**Premio.out**

**“**Error en los datos de entrada. Los importes de las ventas pueden ir de 1 a 5000 “

**Caso 11. Intenta probar si el programa valida correctamente los datos de entrada (valor de N)**

**Premio.in**

2 //Vendedoras

3 //V1

47

84

27

4 //V2

94

41

15

421

2410 //N

**Premio.out**

**“**Error en los datos de entrada. El valor de N es inválido (n puede ir de 1 a 1000) “