[\*] Start mutation process:

- targets: smart\_calc

- tests: test\_smart\_calc

[\*] 0 tests passed:

- test\_smart\_calc [0.00008 s]

[\*] Start mutants generation and execution:

- [# 1] AOR smart\_calc:

--------------------------------------------------------------------------------

14:

15:

16: def lcm(a, b):

17: L = a if a > b else b

- 18: while L <= a \* b:

+ 18: while L <= a / b:

19: if (L % a == 0 and L % b == 0):

20: return L

21: L += 1

22:

--------------------------------------------------------------------------------

[0.00608 s] survived

- [# 2] AOR smart\_calc:

--------------------------------------------------------------------------------

14:

15:

16: def lcm(a, b):

17: L = a if a > b else b

- 18: while L <= a \* b:

+ 18: while L <= a // b:

19: if (L % a == 0 and L % b == 0):

20: return L

21: L += 1

22:

--------------------------------------------------------------------------------

[0.00604 s] survived

- [# 3] AOR smart\_calc:

--------------------------------------------------------------------------------

14:

15:

16: def lcm(a, b):

17: L = a if a > b else b

- 18: while L <= a \* b:

+ 18: while L <= a \*\* b:

19: if (L % a == 0 and L % b == 0):

20: return L

21: L += 1

22:

--------------------------------------------------------------------------------

[0.00638 s] survived

- [# 4] AOR smart\_calc:

--------------------------------------------------------------------------------

15:

16: def lcm(a, b):

17: L = a if a > b else b

18: while L <= a \* b:

- 19: if (L % a == 0 and L % b == 0):

+ 19: if (L \* a == 0 and L % b == 0):

20: return L

21: L += 1

22:

23:

--------------------------------------------------------------------------------

[0.00712 s] survived

- [# 5] AOR smart\_calc:

--------------------------------------------------------------------------------

15:

16: def lcm(a, b):

17: L = a if a > b else b

18: while L <= a \* b:

- 19: if (L % a == 0 and L % b == 0):

+ 19: if (L % a == 0 and L \* b == 0):

20: return L

21: L += 1

22:

23:

--------------------------------------------------------------------------------

[0.00612 s] survived

- [# 6] AOR smart\_calc:

--------------------------------------------------------------------------------

23:

24: def hcf(a, b):

25: H = a if a < b else b

26: while H >= 1:

- 27: if (a % H == 0 and b % H == 0):

+ 27: if (a \* H == 0 and b % H == 0):

28: return H

29: H -= 1

30:

31:

--------------------------------------------------------------------------------

[0.00660 s] survived

- [# 7] AOR smart\_calc:

--------------------------------------------------------------------------------

23:

24: def hcf(a, b):

25: H = a if a < b else b

26: while H >= 1:

- 27: if (a % H == 0 and b % H == 0):

+ 27: if (a % H == 0 and b \* H == 0):

28: return H

29: H -= 1

30:

31:

--------------------------------------------------------------------------------

[0.00749 s] survived

- [# 8] AOR smart\_calc:

--------------------------------------------------------------------------------

29: H -= 1

30:

31:

32: def add(a, b):

- 33: return a + b

+ 33: return a - b

34:

35:

36: def sub(a, b):

37: return a - b

--------------------------------------------------------------------------------

[0.00687 s] survived

- [# 9] AOR smart\_calc:

--------------------------------------------------------------------------------

33: return a + b

34:

35:

36: def sub(a, b):

- 37: return a - b

+ 37: return a + b

38:

39:

40: def mul(a, b):

41: return a \* b

--------------------------------------------------------------------------------

[0.00787 s] survived

- [# 10] AOR smart\_calc:

--------------------------------------------------------------------------------

37: return a - b

38:

39:

40: def mul(a, b):

- 41: return a \* b

+ 41: return a / b

42:

43:

44: def div(a, b):

45: return a / b

--------------------------------------------------------------------------------

[0.00604 s] survived

- [# 11] AOR smart\_calc:

--------------------------------------------------------------------------------

37: return a - b

38:

39:

40: def mul(a, b):

- 41: return a \* b

+ 41: return a // b

42:

43:

44: def div(a, b):

45: return a / b

--------------------------------------------------------------------------------

[0.00939 s] survived

- [# 12] AOR smart\_calc:

--------------------------------------------------------------------------------

37: return a - b

38:

39:

40: def mul(a, b):

- 41: return a \* b

+ 41: return a \*\* b

42:

43:

44: def div(a, b):

45: return a / b

--------------------------------------------------------------------------------

[0.00650 s] survived

- [# 13] AOR smart\_calc:

--------------------------------------------------------------------------------

41: return a \* b

42:

43:

44: def div(a, b):

- 45: return a / b

+ 45: return a // b

46:

47:

48: def mod(a, b):

49: return a % b

--------------------------------------------------------------------------------

[0.00682 s] survived

- [# 14] AOR smart\_calc:

--------------------------------------------------------------------------------

41: return a \* b

42:

43:

44: def div(a, b):

- 45: return a / b

+ 45: return a \* b

46:

47:

48: def mod(a, b):

49: return a % b

--------------------------------------------------------------------------------

[0.00648 s] survived

- [# 15] AOR smart\_calc:

--------------------------------------------------------------------------------

45: return a / b

46:

47:

48: def mod(a, b):

- 49: return a % b

+ 49: return a \* b

50:

51:

52:

53: def end():

--------------------------------------------------------------------------------

[0.00607 s] survived

- [# 16] ASR smart\_calc:

--------------------------------------------------------------------------------

17: L = a if a > b else b

18: while L <= a \* b:

19: if (L % a == 0 and L % b == 0):

20: return L

- 21: L += 1

+ 21: L -= 1

22:

23:

24: def hcf(a, b):

25: H = a if a < b else b

--------------------------------------------------------------------------------

[0.00616 s] survived

- [# 17] ASR smart\_calc:

--------------------------------------------------------------------------------

25: H = a if a < b else b

26: while H >= 1:

27: if (a % H == 0 and b % H == 0):

28: return H

- 29: H -= 1

+ 29: H += 1

30:

31:

32: def add(a, b):

33: return a + b

--------------------------------------------------------------------------------

[0.00618 s] survived

- [# 18] COI smart\_calc:

--------------------------------------------------------------------------------

14:

15:

16: def lcm(a, b):

17: L = a if a > b else b

- 18: while L <= a \* b:

+ 18: while not (L <= a \* b):

19: if (L % a == 0 and L % b == 0):

20: return L

21: L += 1

22:

--------------------------------------------------------------------------------

[0.00621 s] survived

- [# 19] COI smart\_calc:

--------------------------------------------------------------------------------

15:

16: def lcm(a, b):

17: L = a if a > b else b

18: while L <= a \* b:

- 19: if (L % a == 0 and L % b == 0):

+ 19: if not ((L % a == 0 and L % b == 0)):

20: return L

21: L += 1

22:

23:

--------------------------------------------------------------------------------

[0.00627 s] survived

- [# 20] COI smart\_calc:

--------------------------------------------------------------------------------

22:

23:

24: def hcf(a, b):

25: H = a if a < b else b

- 26: while H >= 1:

+ 26: while not (H >= 1):

27: if (a % H == 0 and b % H == 0):

28: return H

29: H -= 1

30:

--------------------------------------------------------------------------------

[0.00624 s] survived

- [# 21] COI smart\_calc:

--------------------------------------------------------------------------------

23:

24: def hcf(a, b):

25: H = a if a < b else b

26: while H >= 1:

- 27: if (a % H == 0 and b % H == 0):

+ 27: if not ((a % H == 0 and b % H == 0)):

28: return H

29: H -= 1

30:

31:

--------------------------------------------------------------------------------

[0.00602 s] survived

- [# 22] EHD smart\_calc:

--------------------------------------------------------------------------------

8: for t in text.split(' '):

9: try:

10: l.append(float(t))

11: except ValueError:

- 12: pass

+ 12: raise

13: return l

14:

15:

16: def lcm(a, b):

--------------------------------------------------------------------------------

[0.00653 s] survived

- [# 23] LCR smart\_calc:

--------------------------------------------------------------------------------

15:

16: def lcm(a, b):

17: L = a if a > b else b

18: while L <= a \* b:

- 19: if (L % a == 0 and L % b == 0):

+ 19: if (L % a == 0 or L % b == 0):

20: return L

21: L += 1

22:

23:

--------------------------------------------------------------------------------

[0.00625 s] survived

- [# 24] LCR smart\_calc:

--------------------------------------------------------------------------------

23:

24: def hcf(a, b):

25: H = a if a < b else b

26: while H >= 1:

- 27: if (a % H == 0 and b % H == 0):

+ 27: if (a % H == 0 or b % H == 0):

28: return H

29: H -= 1

30:

31:

--------------------------------------------------------------------------------

[0.00618 s] survived

- [# 25] ROR smart\_calc:

--------------------------------------------------------------------------------

13: return l

14:

15:

16: def lcm(a, b):

- 17: L = a if a > b else b

+ 17: L = a if a < b else b

18: while L <= a \* b:

19: if (L % a == 0 and L % b == 0):

20: return L

21: L += 1

--------------------------------------------------------------------------------

[0.00625 s] survived

- [# 26] ROR smart\_calc:

--------------------------------------------------------------------------------

13: return l

14:

15:

16: def lcm(a, b):

- 17: L = a if a > b else b

+ 17: L = a if a >= b else b

18: while L <= a \* b:

19: if (L % a == 0 and L % b == 0):

20: return L

21: L += 1

--------------------------------------------------------------------------------

[0.00643 s] survived

- [# 27] ROR smart\_calc:

--------------------------------------------------------------------------------

14:

15:

16: def lcm(a, b):

17: L = a if a > b else b

- 18: while L <= a \* b:

+ 18: while L >= a \* b:

19: if (L % a == 0 and L % b == 0):

20: return L

21: L += 1

22:

--------------------------------------------------------------------------------

[0.00610 s] survived

- [# 28] ROR smart\_calc:

--------------------------------------------------------------------------------

14:

15:

16: def lcm(a, b):

17: L = a if a > b else b

- 18: while L <= a \* b:

+ 18: while L < a \* b:

19: if (L % a == 0 and L % b == 0):

20: return L

21: L += 1

22:

--------------------------------------------------------------------------------

[0.00607 s] survived

- [# 29] ROR smart\_calc:

--------------------------------------------------------------------------------

15:

16: def lcm(a, b):

17: L = a if a > b else b

18: while L <= a \* b:

- 19: if (L % a == 0 and L % b == 0):

+ 19: if (L % a != 0 and L % b == 0):

20: return L

21: L += 1

22:

23:

--------------------------------------------------------------------------------

[0.00712 s] survived

- [# 30] ROR smart\_calc:

--------------------------------------------------------------------------------

15:

16: def lcm(a, b):

17: L = a if a > b else b

18: while L <= a \* b:

- 19: if (L % a == 0 and L % b == 0):

+ 19: if (L % a == 0 and L % b != 0):

20: return L

21: L += 1

22:

23:

--------------------------------------------------------------------------------

[0.00664 s] survived

- [# 31] ROR smart\_calc:

--------------------------------------------------------------------------------

21: L += 1

22:

23:

24: def hcf(a, b):

- 25: H = a if a < b else b

+ 25: H = a if a > b else b

26: while H >= 1:

27: if (a % H == 0 and b % H == 0):

28: return H

29: H -= 1

--------------------------------------------------------------------------------

[0.00654 s] survived

- [# 32] ROR smart\_calc:

--------------------------------------------------------------------------------

21: L += 1

22:

23:

24: def hcf(a, b):

- 25: H = a if a < b else b

+ 25: H = a if a <= b else b

26: while H >= 1:

27: if (a % H == 0 and b % H == 0):

28: return H

29: H -= 1

--------------------------------------------------------------------------------

[0.00602 s] survived

- [# 33] ROR smart\_calc:

--------------------------------------------------------------------------------

22:

23:

24: def hcf(a, b):

25: H = a if a < b else b

- 26: while H >= 1:

+ 26: while H <= 1:

27: if (a % H == 0 and b % H == 0):

28: return H

29: H -= 1

30:

--------------------------------------------------------------------------------

[0.00616 s] survived

- [# 34] ROR smart\_calc:

--------------------------------------------------------------------------------

22:

23:

24: def hcf(a, b):

25: H = a if a < b else b

- 26: while H >= 1:

+ 26: while H > 1:

27: if (a % H == 0 and b % H == 0):

28: return H

29: H -= 1

30:

--------------------------------------------------------------------------------

[0.00689 s] survived

- [# 35] ROR smart\_calc:

--------------------------------------------------------------------------------

23:

24: def hcf(a, b):

25: H = a if a < b else b

26: while H >= 1:

- 27: if (a % H == 0 and b % H == 0):

+ 27: if (a % H != 0 and b % H == 0):

28: return H

29: H -= 1

30:

31:

--------------------------------------------------------------------------------

[0.00639 s] survived

- [# 36] ROR smart\_calc:

--------------------------------------------------------------------------------

23:

24: def hcf(a, b):

25: H = a if a < b else b

26: while H >= 1:

- 27: if (a % H == 0 and b % H == 0):

+ 27: if (a % H == 0 and b % H != 0):

28: return H

29: H -= 1

30:

31:

--------------------------------------------------------------------------------

[0.00647 s] survived

[\*] Mutation score [1.44351 s]: 0.0%

- all: 36

- killed: 0 (0.0%)

- survived: 36 (100.0%)

- incompetent: 0 (0.0%)

- timeout: 0 (0.0%)