Вариант: 1-3-1

1. gcd(64, 100) = 4

1 def gcd(x=64, y=100)

2 if 64 == 0: --- False

4 if 100 == 0: --- False

6 while 100 != 0: --- True

7 rem = 64 % 100

rem = 64

8 x = 100

9 y = 64

6 while 64 != 0: --- True

7 rem = 100 % 64

rem = 36

8 x = 64

9 y = 36

6 while 36 != 0: --- True

7 rem = 64 % 36

rem = 28

8 x = 36

9 y = 28

6 while 28 != 0: --- True

7 rem = 36 % 28

rem = 8

8 x = 28

9 y = 8

6 while 8 != 0: --- True

7 rem = 28 % 8

rem = 4

8 x = 8

9 y = 4

6 while 4 != 0: --- True

7 rem = 8 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, 92) = 92

1 def gcd(x=0, y=92)

2 if 0 == 0: --- True

3 return 92

3. gcd(38, 0) = 38

1 def gcd(x=38, y=0)

2 if 38 == 0: --- False

4 if 0 == 0: --- True

5 return 38

4. hex(186) = 'BA'

3 def hex(number=186)

4 if 186 == 0: --- False

6 res = ''

7 while 186 > 0: --- True

8 digit = 186 % 16

digit = 10

9 res = DIGITS[10] + ''

res = 'A'

10 number = 186 // 16

number = 11

7 while 11 > 0: --- True

8 digit = 11 % 16

digit = 11

9 res = DIGITS[11] + 'A'

res = 'BA'

10 number = 11 // 16

number = 0

7 while 0 > 0: --- False

11 return 'BA'

5. square\_equal(-4, 26, 30) = [7.5, -1.0]

3 def square\_equal(a=-4, b=26, c=30)

4 if -4 != 0: --- True

5 D = 26\*26 - 4\*-4\*30

D = 1156

6 if 1156 > 0: --- True

7 x1 = (-26 - sqrt(1156)) / (2\*-4)

x1 = 7.5

8 x2 = (-26 + sqrt(1156)) / (2\*-4)

x2 = -1.0

9 return [7.5, -1.0]

6. square\_equal(49, 98, 49) = [-1.0]

3 def square\_equal(a=49, b=98, c=49)

4 if 49 != 0: --- True

5 D = 98\*98 - 4\*49\*49

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-1.0]

7. square\_equal(29, 9, 8) = []

3 def square\_equal(a=29, b=9, c=8)

4 if 29 != 0: --- True

5 D = 9\*9 - 4\*29\*8

D = -847

6 if -847 > 0: --- False

10 elif -847 == 0: --- False

12 else:

13 return []

8. square\_equal(0, 40, -86) = [2.15]

3 def square\_equal(a=0, b=40, c=-86)

4 if 0 != 0: --- False

14 else:

15 if 40 != 0: --- True

16 return [2.15]

9. square\_equal(0, 0, 32) = []

3 def square\_equal(a=0, b=0, c=32)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-3-2

1. gcd(55, 90) = 5

1 def gcd(x=55, y=90)

2 if 55 == 0: --- False

4 if 90 == 0: --- False

6 while 90 != 0: --- True

7 rem = 55 % 90

rem = 55

8 x = 90

9 y = 55

6 while 55 != 0: --- True

7 rem = 90 % 55

rem = 35

8 x = 55

9 y = 35

6 while 35 != 0: --- True

7 rem = 55 % 35

rem = 20

8 x = 35

9 y = 20

6 while 20 != 0: --- True

7 rem = 35 % 20

rem = 15

8 x = 20

9 y = 15

6 while 15 != 0: --- True

7 rem = 20 % 15

rem = 5

8 x = 15

9 y = 5

6 while 5 != 0: --- True

7 rem = 15 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

2. gcd(0, 25) = 25

1 def gcd(x=0, y=25)

2 if 0 == 0: --- True

3 return 25

3. gcd(57, 0) = 57

1 def gcd(x=57, y=0)

2 if 57 == 0: --- False

4 if 0 == 0: --- True

5 return 57

4. hex(191) = 'BF'

3 def hex(number=191)

4 if 191 == 0: --- False

6 res = ''

7 while 191 > 0: --- True

8 digit = 191 % 16

digit = 15

9 res = DIGITS[15] + ''

res = 'F'

10 number = 191 // 16

number = 11

7 while 11 > 0: --- True

8 digit = 11 % 16

digit = 11

9 res = DIGITS[11] + 'F'

res = 'BF'

10 number = 11 // 16

number = 0

7 while 0 > 0: --- False

11 return 'BF'

5. square\_equal(-1, -56, 57) = [1.0, -57.0]

3 def square\_equal(a=-1, b=-56, c=57)

4 if -1 != 0: --- True

5 D = -56\*-56 - 4\*-1\*57

D = 3364

6 if 3364 > 0: --- True

7 x1 = (--56 - sqrt(3364)) / (2\*-1)

x1 = 1.0

8 x2 = (--56 + sqrt(3364)) / (2\*-1)

x2 = -57.0

9 return [1.0, -57.0]

6. square\_equal(9, 54, 81) = [-3.0]

3 def square\_equal(a=9, b=54, c=81)

4 if 9 != 0: --- True

5 D = 54\*54 - 4\*9\*81

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-3.0]

7. square\_equal(-5, 16, -43) = []

3 def square\_equal(a=-5, b=16, c=-43)

4 if -5 != 0: --- True

5 D = 16\*16 - 4\*-5\*-43

D = -604

6 if -604 > 0: --- False

10 elif -604 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -75, 33) = [0.44]

3 def square\_equal(a=0, b=-75, c=33)

4 if 0 != 0: --- False

14 else:

15 if -75 != 0: --- True

16 return [0.44]

9. square\_equal(0, 0, 65) = []

3 def square\_equal(a=0, b=0, c=65)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-3-3

1. gcd(54, 96) = 6

1 def gcd(x=54, y=96)

2 if 54 == 0: --- False

4 if 96 == 0: --- False

6 while 96 != 0: --- True

7 rem = 54 % 96

rem = 54

8 x = 96

9 y = 54

6 while 54 != 0: --- True

7 rem = 96 % 54

rem = 42

8 x = 54

9 y = 42

6 while 42 != 0: --- True

7 rem = 54 % 42

rem = 12

8 x = 42

9 y = 12

6 while 12 != 0: --- True

7 rem = 42 % 12

rem = 6

8 x = 12

9 y = 6

6 while 6 != 0: --- True

7 rem = 12 % 6

rem = 0

8 x = 6

9 y = 0

6 while 0 != 0: --- False

10 return 6

2. gcd(0, 5) = 5

1 def gcd(x=0, y=5)

2 if 0 == 0: --- True

3 return 5

3. gcd(30, 0) = 30

1 def gcd(x=30, y=0)

2 if 30 == 0: --- False

4 if 0 == 0: --- True

5 return 30

4. hex(210) = 'D2'

3 def hex(number=210)

4 if 210 == 0: --- False

6 res = ''

7 while 210 > 0: --- True

8 digit = 210 % 16

digit = 2

9 res = DIGITS[2] + ''

res = '2'

10 number = 210 // 16

number = 13

7 while 13 > 0: --- True

8 digit = 13 % 16

digit = 13

9 res = DIGITS[13] + '2'

res = 'D2'

10 number = 13 // 16

number = 0

7 while 0 > 0: --- False

11 return 'D2'

5. square\_equal(-16, -40, 75) = [1.25, -3.75]

3 def square\_equal(a=-16, b=-40, c=75)

4 if -16 != 0: --- True

5 D = -40\*-40 - 4\*-16\*75

D = 6400

6 if 6400 > 0: --- True

7 x1 = (--40 - sqrt(6400)) / (2\*-16)

x1 = 1.25

8 x2 = (--40 + sqrt(6400)) / (2\*-16)

x2 = -3.75

9 return [1.25, -3.75]

6. square\_equal(-25, -20, -4) = [-0.4]

3 def square\_equal(a=-25, b=-20, c=-4)

4 if -25 != 0: --- True

5 D = -20\*-20 - 4\*-25\*-4

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-0.4]

7. square\_equal(5, -19, 19) = []

3 def square\_equal(a=5, b=-19, c=19)

4 if 5 != 0: --- True

5 D = -19\*-19 - 4\*5\*19

D = -19

6 if -19 > 0: --- False

10 elif -19 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -75, 81) = [1.08]

3 def square\_equal(a=0, b=-75, c=81)

4 if 0 != 0: --- False

14 else:

15 if -75 != 0: --- True

16 return [1.08]

9. square\_equal(0, 0, 75) = []

3 def square\_equal(a=0, b=0, c=75)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-3-4

1. gcd(70, 85) = 5

1 def gcd(x=70, y=85)

2 if 70 == 0: --- False

4 if 85 == 0: --- False

6 while 85 != 0: --- True

7 rem = 70 % 85

rem = 70

8 x = 85

9 y = 70

6 while 70 != 0: --- True

7 rem = 85 % 70

rem = 15

8 x = 70

9 y = 15

6 while 15 != 0: --- True

7 rem = 70 % 15

rem = 10

8 x = 15

9 y = 10

6 while 10 != 0: --- True

7 rem = 15 % 10

rem = 5

8 x = 10

9 y = 5

6 while 5 != 0: --- True

7 rem = 10 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

2. gcd(0, 48) = 48

1 def gcd(x=0, y=48)

2 if 0 == 0: --- True

3 return 48

3. gcd(36, 0) = 36

1 def gcd(x=36, y=0)

2 if 36 == 0: --- False

4 if 0 == 0: --- True

5 return 36

4. hex(214) = 'D6'

3 def hex(number=214)

4 if 214 == 0: --- False

6 res = ''

7 while 214 > 0: --- True

8 digit = 214 % 16

digit = 6

9 res = DIGITS[6] + ''

res = '6'

10 number = 214 // 16

number = 13

7 while 13 > 0: --- True

8 digit = 13 % 16

digit = 13

9 res = DIGITS[13] + '6'

res = 'D6'

10 number = 13 // 16

number = 0

7 while 0 > 0: --- False

11 return 'D6'

5. square\_equal(-4, -54, 90) = [1.5, -15.0]

3 def square\_equal(a=-4, b=-54, c=90)

4 if -4 != 0: --- True

5 D = -54\*-54 - 4\*-4\*90

D = 4356

6 if 4356 > 0: --- True

7 x1 = (--54 - sqrt(4356)) / (2\*-4)

x1 = 1.5

8 x2 = (--54 + sqrt(4356)) / (2\*-4)

x2 = -15.0

9 return [1.5, -15.0]

6. square\_equal(-25, -10, -1) = [-0.2]

3 def square\_equal(a=-25, b=-10, c=-1)

4 if -25 != 0: --- True

5 D = -10\*-10 - 4\*-25\*-1

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-0.2]

7. square\_equal(-16, -9, -8) = []

3 def square\_equal(a=-16, b=-9, c=-8)

4 if -16 != 0: --- True

5 D = -9\*-9 - 4\*-16\*-8

D = -431

6 if -431 > 0: --- False

10 elif -431 == 0: --- False

12 else:

13 return []

8. square\_equal(0, 2, 94) = [-47.0]

3 def square\_equal(a=0, b=2, c=94)

4 if 0 != 0: --- False

14 else:

15 if 2 != 0: --- True

16 return [-47.0]

9. square\_equal(0, 0, -32) = []

3 def square\_equal(a=0, b=0, c=-32)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-3-5

1. gcd(55, 85) = 5

1 def gcd(x=55, y=85)

2 if 55 == 0: --- False

4 if 85 == 0: --- False

6 while 85 != 0: --- True

7 rem = 55 % 85

rem = 55

8 x = 85

9 y = 55

6 while 55 != 0: --- True

7 rem = 85 % 55

rem = 30

8 x = 55

9 y = 30

6 while 30 != 0: --- True

7 rem = 55 % 30

rem = 25

8 x = 30

9 y = 25

6 while 25 != 0: --- True

7 rem = 30 % 25

rem = 5

8 x = 25

9 y = 5

6 while 5 != 0: --- True

7 rem = 25 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

2. gcd(0, 90) = 90

1 def gcd(x=0, y=90)

2 if 0 == 0: --- True

3 return 90

3. gcd(51, 0) = 51

1 def gcd(x=51, y=0)

2 if 51 == 0: --- False

4 if 0 == 0: --- True

5 return 51

4. hex(200) = 'C8'

3 def hex(number=200)

4 if 200 == 0: --- False

6 res = ''

7 while 200 > 0: --- True

8 digit = 200 % 16

digit = 8

9 res = DIGITS[8] + ''

res = '8'

10 number = 200 // 16

number = 12

7 while 12 > 0: --- True

8 digit = 12 % 16

digit = 12

9 res = DIGITS[12] + '8'

res = 'C8'

10 number = 12 // 16

number = 0

7 while 0 > 0: --- False

11 return 'C8'

5. square\_equal(-16, 28, 44) = [2.75, -1.0]

3 def square\_equal(a=-16, b=28, c=44)

4 if -16 != 0: --- True

5 D = 28\*28 - 4\*-16\*44

D = 3600

6 if 3600 > 0: --- True

7 x1 = (-28 - sqrt(3600)) / (2\*-16)

x1 = 2.75

8 x2 = (-28 + sqrt(3600)) / (2\*-16)

x2 = -1.0

9 return [2.75, -1.0]

6. square\_equal(-4, -28, -49) = [-3.5]

3 def square\_equal(a=-4, b=-28, c=-49)

4 if -4 != 0: --- True

5 D = -28\*-28 - 4\*-4\*-49

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-3.5]

7. square\_equal(47, -5, 4) = []

3 def square\_equal(a=47, b=-5, c=4)

4 if 47 != 0: --- True

5 D = -5\*-5 - 4\*47\*4

D = -727

6 if -727 > 0: --- False

10 elif -727 == 0: --- False

12 else:

13 return []

8. square\_equal(0, 44, -22) = [0.5]

3 def square\_equal(a=0, b=44, c=-22)

4 if 0 != 0: --- False

14 else:

15 if 44 != 0: --- True

16 return [0.5]

9. square\_equal(0, 0, 24) = []

3 def square\_equal(a=0, b=0, c=24)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-3-6

1. gcd(52, 96) = 4

1 def gcd(x=52, y=96)

2 if 52 == 0: --- False

4 if 96 == 0: --- False

6 while 96 != 0: --- True

7 rem = 52 % 96

rem = 52

8 x = 96

9 y = 52

6 while 52 != 0: --- True

7 rem = 96 % 52

rem = 44

8 x = 52

9 y = 44

6 while 44 != 0: --- True

7 rem = 52 % 44

rem = 8

8 x = 44

9 y = 8

6 while 8 != 0: --- True

7 rem = 44 % 8

rem = 4

8 x = 8

9 y = 4

6 while 4 != 0: --- True

7 rem = 8 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, 23) = 23

1 def gcd(x=0, y=23)

2 if 0 == 0: --- True

3 return 23

3. gcd(4, 0) = 4

1 def gcd(x=4, y=0)

2 if 4 == 0: --- False

4 if 0 == 0: --- True

5 return 4

4. hex(168) = 'A8'

3 def hex(number=168)

4 if 168 == 0: --- False

6 res = ''

7 while 168 > 0: --- True

8 digit = 168 % 16

digit = 8

9 res = DIGITS[8] + ''

res = '8'

10 number = 168 // 16

number = 10

7 while 10 > 0: --- True

8 digit = 10 % 16

digit = 10

9 res = DIGITS[10] + '8'

res = 'A8'

10 number = 10 // 16

number = 0

7 while 0 > 0: --- False

11 return 'A8'

5. square\_equal(-25, 10, 80) = [2.0, -1.6]

3 def square\_equal(a=-25, b=10, c=80)

4 if -25 != 0: --- True

5 D = 10\*10 - 4\*-25\*80

D = 8100

6 if 8100 > 0: --- True

7 x1 = (-10 - sqrt(8100)) / (2\*-25)

x1 = 2.0

8 x2 = (-10 + sqrt(8100)) / (2\*-25)

x2 = -1.6

9 return [2.0, -1.6]

6. square\_equal(16, 16, 4) = [-0.5]

3 def square\_equal(a=16, b=16, c=4)

4 if 16 != 0: --- True

5 D = 16\*16 - 4\*16\*4

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-0.5]

7. square\_equal(-7, -8, -18) = []

3 def square\_equal(a=-7, b=-8, c=-18)

4 if -7 != 0: --- True

5 D = -8\*-8 - 4\*-7\*-18

D = -440

6 if -440 > 0: --- False

10 elif -440 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -2, 92) = [46.0]

3 def square\_equal(a=0, b=-2, c=92)

4 if 0 != 0: --- False

14 else:

15 if -2 != 0: --- True

16 return [46.0]

9. square\_equal(0, 0, -82) = []

3 def square\_equal(a=0, b=0, c=-82)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-3-7

1. gcd(32, 84) = 4

1 def gcd(x=32, y=84)

2 if 32 == 0: --- False

4 if 84 == 0: --- False

6 while 84 != 0: --- True

7 rem = 32 % 84

rem = 32

8 x = 84

9 y = 32

6 while 32 != 0: --- True

7 rem = 84 % 32

rem = 20

8 x = 32

9 y = 20

6 while 20 != 0: --- True

7 rem = 32 % 20

rem = 12

8 x = 20

9 y = 12

6 while 12 != 0: --- True

7 rem = 20 % 12

rem = 8

8 x = 12

9 y = 8

6 while 8 != 0: --- True

7 rem = 12 % 8

rem = 4

8 x = 8

9 y = 4

6 while 4 != 0: --- True

7 rem = 8 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, 32) = 32

1 def gcd(x=0, y=32)

2 if 0 == 0: --- True

3 return 32

3. gcd(9, 0) = 9

1 def gcd(x=9, y=0)

2 if 9 == 0: --- False

4 if 0 == 0: --- True

5 return 9

4. hex(205) = 'CD'

3 def hex(number=205)

4 if 205 == 0: --- False

6 res = ''

7 while 205 > 0: --- True

8 digit = 205 % 16

digit = 13

9 res = DIGITS[13] + ''

res = 'D'

10 number = 205 // 16

number = 12

7 while 12 > 0: --- True

8 digit = 12 % 16

digit = 12

9 res = DIGITS[12] + 'D'

res = 'CD'

10 number = 12 // 16

number = 0

7 while 0 > 0: --- False

11 return 'CD'

5. square\_equal(-25, -85, 18) = [0.2, -3.6]

3 def square\_equal(a=-25, b=-85, c=18)

4 if -25 != 0: --- True

5 D = -85\*-85 - 4\*-25\*18

D = 9025

6 if 9025 > 0: --- True

7 x1 = (--85 - sqrt(9025))/(2\*-25)

x1 = 0.2

8 x2 = (--85 + sqrt(9025))/(2\*-25)

x2 = -3.6

9 return [0.2, -3.6]

6. square\_equal(-1, 14, -49) = [7.0]

3 def square\_equal(a=-1, b=14, c=-49)

4 if -1 != 0: --- True

5 D = 14\*14 - 4\*-1\*-49

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [7.0]

7. square\_equal(-3, 18, -69) = []

3 def square\_equal(a=-3, b=18, c=-69)

4 if -3 != 0: --- True

5 D = 18\*18 - 4\*-3\*-69

D = -504

6 if -504 > 0: --- False

10 elif -504 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -2, -28) = [-14.0]

3 def square\_equal(a=0, b=-2, c=-28)

4 if 0 != 0: --- False

14 else:

15 if -2 != 0: --- True

16 return [-14.0]

9. square\_equal(0, 0, 16) = []

3 def square\_equal(a=0, b=0, c=16)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-3-8

1. gcd(20, 92) = 4

1 def gcd(x=20, y=92)

2 if 20 == 0: --- False

4 if 92 == 0: --- False

6 while 92 != 0: --- True

7 rem = 20 % 92

rem = 20

8 x = 92

9 y = 20

6 while 20 != 0: --- True

7 rem = 92 % 20

rem = 12

8 x = 20

9 y = 12

6 while 12 != 0: --- True

7 rem = 20 % 12

rem = 8

8 x = 12

9 y = 8

6 while 8 != 0: --- True

7 rem = 12 % 8

rem = 4

8 x = 8

9 y = 4

6 while 4 != 0: --- True

7 rem = 8 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, 39) = 39

1 def gcd(x=0, y=39)

2 if 0 == 0: --- True

3 return 39

3. gcd(31, 0) = 31

1 def gcd(x=31, y=0)

2 if 31 == 0: --- False

4 if 0 == 0: --- True

5 return 31

4. hex(246) = 'F6'

3 def hex(number=246)

4 if 246 == 0: --- False

6 res = ''

7 while 246 > 0: --- True

8 digit = 246 % 16

digit = 6

9 res = DIGITS[6] + ''

res = '6'

10 number = 246 // 16

number = 15

7 while 15 > 0: --- True

8 digit = 15 % 16

digit = 15

9 res = DIGITS[15] + '6'

res = 'F6'

10 number = 15 // 16

number = 0

7 while 0 > 0: --- False

11 return 'F6'

5. square\_equal(1, 24, 23) = [-23.0, -1.0]

3 def square\_equal(a=1, b=24, c=23)

4 if 1 != 0: --- True

5 D = 24\*24 - 4\*1\*23

D = 484

6 if 484 > 0: --- True

7 x1 = (-24 - sqrt(484)) / (2\*1)

x1 = -23.0

8 x2 = (-24 + sqrt(484)) / (2\*1)

x2 = -1.0

9 return [-23.0, -1.0]

6. square\_equal(4, -12, 9) = [1.5]

3 def square\_equal(a=4, b=-12, c=9)

4 if 4 != 0: --- True

5 D = -12\*-12 - 4\*4\*9

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [1.5]

7. square\_equal(-3, -15, -55) = []

3 def square\_equal(a=-3, b=-15, c=-55)

4 if -3 != 0: --- True

5 D = -15\*-15 - 4\*-3\*-55

D = -435

6 if -435 > 0: --- False

10 elif -435 == 0: --- False

12 else:

13 return []

8. square\_equal(0, 20, 97) = [-4.85]

3 def square\_equal(a=0, b=20, c=97)

4 if 0 != 0: --- False

14 else:

15 if 20 != 0: --- True

16 return [-4.85]

9. square\_equal(0, 0, 31) = []

3 def square\_equal(a=0, b=0, c=31)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-3-9

1. gcd(44, 56) = 4

1 def gcd(x=44, y=56)

2 if 44 == 0: --- False

4 if 56 == 0: --- False

6 while 56 != 0: --- True

7 rem = 44 % 56

rem = 44

8 x = 56

9 y = 44

6 while 44 != 0: --- True

7 rem = 56 % 44

rem = 12

8 x = 44

9 y = 12

6 while 12 != 0: --- True

7 rem = 44 % 12

rem = 8

8 x = 12

9 y = 8

6 while 8 != 0: --- True

7 rem = 12 % 8

rem = 4

8 x = 8

9 y = 4

6 while 4 != 0: --- True

7 rem = 8 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, 27) = 27

1 def gcd(x=0, y=27)

2 if 0 == 0: --- True

3 return 27

3. gcd(98, 0) = 98

1 def gcd(x=98, y=0)

2 if 98 == 0: --- False

4 if 0 == 0: --- True

5 return 98

4. hex(178) = 'B2'

3 def hex(number=178)

4 if 178 == 0: --- False

6 res = ''

7 while 178 > 0: --- True

8 digit = 178 % 16

digit = 2

9 res = DIGITS[2] + ''

res = '2'

10 number = 178 // 16

number = 11

7 while 11 > 0: --- True

8 digit = 11 % 16

digit = 11

9 res = DIGITS[11] + '2'

res = 'B2'

10 number = 11 // 16

number = 0

7 while 0 > 0: --- False

11 return 'B2'

5. square\_equal(1, 37, 70) = [-35.0, -2.0]

3 def square\_equal(a=1, b=37, c=70)

4 if 1 != 0: --- True

5 D = 37\*37 - 4\*1\*70

D = 1089

6 if 1089 > 0: --- True

7 x1 = (-37 - sqrt(1089)) / (2\*1)

x1 = -35.0

8 x2 = (-37 + sqrt(1089)) / (2\*1)

x2 = -2.0

9 return [-35.0, -2.0]

6. square\_equal(16, 48, 36) = [-1.5]

3 def square\_equal(a=16, b=48, c=36)

4 if 16 != 0: --- True

5 D = 48\*48 - 4\*16\*36

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-1.5]

7. square\_equal(80, -25, 3) = []

3 def square\_equal(a=80, b=-25, c=3)

4 if 80 != 0: --- True

5 D = -25\*-25 - 4\*80\*3

D = -335

6 if -335 > 0: --- False

10 elif -335 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -5, 49) = [9.8]

3 def square\_equal(a=0, b=-5, c=49)

4 if 0 != 0: --- False

14 else:

15 if -5 != 0: --- True

16 return [9.8]

9. square\_equal(0, 0, 87) = []

3 def square\_equal(a=0, b=0, c=87)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-3-10

1. gcd(55, 95) = 5

1 def gcd(x=55, y=95)

2 if 55 == 0: --- False

4 if 95 == 0: --- False

6 while 95 != 0: --- True

7 rem = 55 % 95

rem = 55

8 x = 95

9 y = 55

6 while 55 != 0: --- True

7 rem = 95 % 55

rem = 40

8 x = 55

9 y = 40

6 while 40 != 0: --- True

7 rem = 55 % 40

rem = 15

8 x = 40

9 y = 15

6 while 15 != 0: --- True

7 rem = 40 % 15

rem = 10

8 x = 15

9 y = 10

6 while 10 != 0: --- True

7 rem = 15 % 10

rem = 5

8 x = 10

9 y = 5

6 while 5 != 0: --- True

7 rem = 10 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

2. gcd(0, 49) = 49

1 def gcd(x=0, y=49)

2 if 0 == 0: --- True

3 return 49

3. gcd(17, 0) = 17

1 def gcd(x=17, y=0)

2 if 17 == 0: --- False

4 if 0 == 0: --- True

5 return 17

4. hex(211) = 'D3'

3 def hex(number=211)

4 if 211 == 0: --- False

6 res = ''

7 while 211 > 0: --- True

8 digit = 211 % 16

digit = 3

9 res = DIGITS[3] + ''

res = '3'

10 number = 211 // 16

number = 13

7 while 13 > 0: --- True

8 digit = 13 % 16

digit = 13

9 res = DIGITS[13] + '3'

res = 'D3'

10 number = 13 // 16

number = 0

7 while 0 > 0: --- False

11 return 'D3'

5. square\_equal(16, 28, 6) = [-1.5, -0.25]

3 def square\_equal(a=16, b=28, c=6)

4 if 16 != 0: --- True

5 D = 28\*28 - 4\*16\*6

D = 400

6 if 400 > 0: --- True

7 x1 = (-28 - sqrt(400)) / (2\*16)

x1 = -1.5

8 x2 = (-28 + sqrt(400)) / (2\*16)

x2 = -0.25

9 return [-1.5, -0.25]

6. square\_equal(-64, -32, -4) = [-0.25]

3 def square\_equal(a=-64, b=-32, c=-4)

4 if -64 != 0: --- True

5 D = -32\*-32 - 4\*-64\*-4

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-0.25]

7. square\_equal(98, -6, 2) = []

3 def square\_equal(a=98, b=-6, c=2)

4 if 98 != 0: --- True

5 D = -6\*-6 - 4\*98\*2

D = -748

6 if -748 > 0: --- False

10 elif -748 == 0: --- False

12 else:

13 return []

8. square\_equal(0, 5, -74) = [14.8]

3 def square\_equal(a=0, b=5, c=-74)

4 if 0 != 0: --- False

14 else:

15 if 5 != 0: --- True

16 return [14.8]

9. square\_equal(0, 0, -24) = []

3 def square\_equal(a=0, b=0, c=-24)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-3-11

1. gcd(36, 100) = 4

1 def gcd(x=36, y=100)

2 if 36 == 0: --- False

4 if 100 == 0: --- False

6 while 100 != 0: --- True

7 rem = 36 % 100

rem = 36

8 x = 100

9 y = 36

6 while 36 != 0: --- True

7 rem = 100 % 36

rem = 28

8 x = 36

9 y = 28

6 while 28 != 0: --- True

7 rem = 36 % 28

rem = 8

8 x = 28

9 y = 8

6 while 8 != 0: --- True

7 rem = 28 % 8

rem = 4

8 x = 8

9 y = 4

6 while 4 != 0: --- True

7 rem = 8 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, 96) = 96

1 def gcd(x=0, y=96)

2 if 0 == 0: --- True

3 return 96

3. gcd(23, 0) = 23

1 def gcd(x=23, y=0)

2 if 23 == 0: --- False

4 if 0 == 0: --- True

5 return 23

4. hex(201) = 'C9'

3 def hex(number=201)

4 if 201 == 0: --- False

6 res = ''

7 while 201 > 0: --- True

8 digit = 201 % 16

digit = 9

9 res = DIGITS[9] + ''

res = '9'

10 number = 201 // 16

number = 12

7 while 12 > 0: --- True

8 digit = 12 % 16

digit = 12

9 res = DIGITS[12] + '9'

res = 'C9'

10 number = 12 // 16

number = 0

7 while 0 > 0: --- False

11 return 'C9'

5. square\_equal(4, -28, 45) = [2.5, 4.5]

3 def square\_equal(a=4, b=-28, c=45)

4 if 4 != 0: --- True

5 D = -28\*-28 - 4\*4\*45

D = 64

6 if 64 > 0: --- True

7 x1 = (--28 - sqrt(64)) / (2\*4)

x1 = 2.5

8 x2 = (--28 + sqrt(64)) / (2\*4)

x2 = 4.5

9 return [2.5, 4.5]

6. square\_equal(-4, -12, -9) = [-1.5]

3 def square\_equal(a=-4, b=-12, c=-9)

4 if -4 != 0: --- True

5 D = -12\*-12 - 4\*-4\*-9

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-1.5]

7. square\_equal(7, 17, 28) = []

3 def square\_equal(a=7, b=17, c=28)

4 if 7 != 0: --- True

5 D = 17\*17 - 4\*7\*28

D = -495

6 if -495 > 0: --- False

10 elif -495 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -75, -93) = [-1.24]

3 def square\_equal(a=0, b=-75, c=-93)

4 if 0 != 0: --- False

14 else:

15 if -75 != 0: --- True

16 return [-1.24]

9. square\_equal(0, 0, -2) = []

3 def square\_equal(a=0, b=0, c=-2)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-3-12

1. gcd(49, 84) = 7

1 def gcd(x=49, y=84)

2 if 49 == 0: --- False

4 if 84 == 0: --- False

6 while 84 != 0: --- True

7 rem = 49 % 84

rem = 49

8 x = 84

9 y = 49

6 while 49 != 0: --- True

7 rem = 84 % 49

rem = 35

8 x = 49

9 y = 35

6 while 35 != 0: --- True

7 rem = 49 % 35

rem = 14

8 x = 35

9 y = 14

6 while 14 != 0: --- True

7 rem = 35 % 14

rem = 7

8 x = 14

9 y = 7

6 while 7 != 0: --- True

7 rem = 14 % 7

rem = 0

8 x = 7

9 y = 0

6 while 0 != 0: --- False

10 return 7

2. gcd(0, 89) = 89

1 def gcd(x=0, y=89)

2 if 0 == 0: --- True

3 return 89

3. gcd(15, 0) = 15

1 def gcd(x=15, y=0)

2 if 15 == 0: --- False

4 if 0 == 0: --- True

5 return 15

4. hex(217) = 'D9'

3 def hex(number=217)

4 if 217 == 0: --- False

6 res = ''

7 while 217 > 0: --- True

8 digit = 217 % 16

digit = 9

9 res = DIGITS[9] + ''

res = '9'

10 number = 217 // 16

number = 13

7 while 13 > 0: --- True

8 digit = 13 % 16

digit = 13

9 res = DIGITS[13] + '9'

res = 'D9'

10 number = 13 // 16

number = 0

7 while 0 > 0: --- False

11 return 'D9'

5. square\_equal(-1, -66, 67) = [1.0, -67.0]

3 def square\_equal(a=-1, b=-66, c=67)

4 if -1 != 0: --- True

5 D = -66\*-66 - 4\*-1\*67

D = 4624

6 if 4624 > 0: --- True

7 x1 = (--66 - sqrt(4624)) / (2\*-1)

x1 = 1.0

8 x2 = (--66 + sqrt(4624)) / (2\*-1)

x2 = -67.0

9 return [1.0, -67.0]

6. square\_equal(-25, 30, -9) = [0.6]

3 def square\_equal(a=-25, b=30, c=-9)

4 if -25 != 0: --- True

5 D = 30\*30 - 4\*-25\*-9

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [0.6]

7. square\_equal(3, 7, 9) = []

3 def square\_equal(a=3, b=7, c=9)

4 if 3 != 0: --- True

5 D = 7\*7 - 4\*3\*9

D = -59

6 if -59 > 0: --- False

10 elif -59 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -5, 38) = [7.6]

3 def square\_equal(a=0, b=-5, c=38)

4 if 0 != 0: --- False

14 else:

15 if -5 != 0: --- True

16 return [7.6]

9. square\_equal(0, 0, -26) = []

3 def square\_equal(a=0, b=0, c=-26)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-3-13

1. gcd(36, 92) = 4

1 def gcd(x=36, y=92)

2 if 36 == 0: --- False

4 if 92 == 0: --- False

6 while 92 != 0: --- True

7 rem = 36 % 92

rem = 36

8 x = 92

9 y = 36

6 while 36 != 0: --- True

7 rem = 92 % 36

rem = 20

8 x = 36

9 y = 20

6 while 20 != 0: --- True

7 rem = 36 % 20

rem = 16

8 x = 20

9 y = 16

6 while 16 != 0: --- True

7 rem = 20 % 16

rem = 4

8 x = 16

9 y = 4

6 while 4 != 0: --- True

7 rem = 16 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, 91) = 91

1 def gcd(x=0, y=91)

2 if 0 == 0: --- True

3 return 91

3. gcd(88, 0) = 88

1 def gcd(x=88, y=0)

2 if 88 == 0: --- False

4 if 0 == 0: --- True

5 return 88

4. hex(176) = 'B0'

3 def hex(number=176)

4 if 176 == 0: --- False

6 res = ''

7 while 176 > 0: --- True

8 digit = 176 % 16

digit = 0

9 res = DIGITS[0] + ''

res = '0'

10 number = 176 // 16

number = 11

7 while 11 > 0: --- True

8 digit = 11 % 16

digit = 11

9 res = DIGITS[11] + '0'

res = 'B0'

10 number = 11 // 16

number = 0

7 while 0 > 0: --- False

11 return 'B0'

5. square\_equal(1, 47, 46) = [-46.0, -1.0]

3 def square\_equal(a=1, b=47, c=46)

4 if 1 != 0: --- True

5 D = 47\*47 - 4\*1\*46

D = 2025

6 if 2025 > 0: --- True

7 x1 = (-47 - sqrt(2025)) / (2\*1)

x1 = -46.0

8 x2 = (-47 + sqrt(2025)) / (2\*1)

x2 = -1.0

9 return [-46.0, -1.0]

6. square\_equal(-64, -64, -16) = [-0.5]

3 def square\_equal(a=-64, b=-64, c=-16)

4 if -64 != 0: --- True

5 D = -64\*-64 - 4\*-64\*-16

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-0.5]

7. square\_equal(88, 7, 2) = []

3 def square\_equal(a=88, b=7, c=2)

4 if 88 != 0: --- True

5 D = 7\*7 - 4\*88\*2

D = -655

6 if -655 > 0: --- False

10 elif -655 == 0: --- False

12 else:

13 return []

8. square\_equal(0, 50, -42) = [0.84]

3 def square\_equal(a=0, b=50, c=-42)

4 if 0 != 0: --- False

14 else:

15 if 50 != 0: --- True

16 return [0.84]

9. square\_equal(0, 0, -79) = []

3 def square\_equal(a=0, b=0, c=-79)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-3-14

1. gcd(44, 60) = 4

1 def gcd(x=44, y=60)

2 if 44 == 0: --- False

4 if 60 == 0: --- False

6 while 60 != 0: --- True

7 rem = 44 % 60

rem = 44

8 x = 60

9 y = 44

6 while 44 != 0: --- True

7 rem = 60 % 44

rem = 16

8 x = 44

9 y = 16

6 while 16 != 0: --- True

7 rem = 44 % 16

rem = 12

8 x = 16

9 y = 12

6 while 12 != 0: --- True

7 rem = 16 % 12

rem = 4

8 x = 12

9 y = 4

6 while 4 != 0: --- True

7 rem = 12 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, 19) = 19

1 def gcd(x=0, y=19)

2 if 0 == 0: --- True

3 return 19

3. gcd(47, 0) = 47

1 def gcd(x=47, y=0)

2 if 47 == 0: --- False

4 if 0 == 0: --- True

5 return 47

4. hex(229) = 'E5'

3 def hex(number=229)

4 if 229 == 0: --- False

6 res = ''

7 while 229 > 0: --- True

8 digit = 229 % 16

digit = 5

9 res = DIGITS[5] + ''

res = '5'

10 number = 229 // 16

number = 14

7 while 14 > 0: --- True

8 digit = 14 % 16

digit = 14

9 res = DIGITS[14] + '5'

res = 'E5'

10 number = 14 // 16

number = 0

7 while 0 > 0: --- False

11 return 'E5'

5. square\_equal(4, 76, 37) = [-18.5, -0.5]

3 def square\_equal(a=4, b=76, c=37)

4 if 4 != 0: --- True

5 D = 76\*76 - 4\*4\*37

D = 5184

6 if 5184 > 0: --- True

7 x1 = (-76 - sqrt(5184)) / (2\*4)

x1 = -18.5

8 x2 = (-76 + sqrt(5184)) / (2\*4)

x2 = -0.5

9 return [-18.5, -0.5]

6. square\_equal(16, 40, 25) = [-1.25]

3 def square\_equal(a=16, b=40, c=25)

4 if 16 != 0: --- True

5 D = 40\*40 - 4\*16\*25

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-1.25]

7. square\_equal(79, -8, 2) = []

3 def square\_equal(a=79, b=-8, c=2)

4 if 79 != 0: --- True

5 D = -8\*-8 - 4\*79\*2

D = -568

6 if -568 > 0: --- False

10 elif -568 == 0: --- False

12 else:

13 return []

8. square\_equal(0, 1, -97) = [97.0]

3 def square\_equal(a=0, b=1, c=-97)

4 if 0 != 0: --- False

14 else:

15 if 1 != 0: --- True

16 return [97.0]

9. square\_equal(0, 0, -97) = []

3 def square\_equal(a=0, b=0, c=-97)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-3-15

1. gcd(42, 72) = 6

1 def gcd(x=42, y=72)

2 if 42 == 0: --- False

4 if 72 == 0: --- False

6 while 72 != 0: --- True

7 rem = 42 % 72

rem = 42

8 x = 72

9 y = 42

6 while 42 != 0: --- True

7 rem = 72 % 42

rem = 30

8 x = 42

9 y = 30

6 while 30 != 0: --- True

7 rem = 42 % 30

rem = 12

8 x = 30

9 y = 12

6 while 12 != 0: --- True

7 rem = 30 % 12

rem = 6

8 x = 12

9 y = 6

6 while 6 != 0: --- True

7 rem = 12 % 6

rem = 0

8 x = 6

9 y = 0

6 while 0 != 0: --- False

10 return 6

2. gcd(0, 94) = 94

1 def gcd(x=0, y=94)

2 if 0 == 0: --- True

3 return 94

3. gcd(37, 0) = 37

1 def gcd(x=37, y=0)

2 if 37 == 0: --- False

4 if 0 == 0: --- True

5 return 37

4. hex(245) = 'F5'

3 def hex(number=245)

4 if 245 == 0: --- False

6 res = ''

7 while 245 > 0: --- True

8 digit = 245 % 16

digit = 5

9 res = DIGITS[5] + ''

res = '5'

10 number = 245 // 16

number = 15

7 while 15 > 0: --- True

8 digit = 15 % 16

digit = 15

9 res = DIGITS[15] + '5'

res = 'F5'

10 number = 15 // 16

number = 0

7 while 0 > 0: --- False

11 return 'F5'

5. square\_equal(-16, 56, 72) = [4.5, -1.0]

3 def square\_equal(a=-16, b=56, c=72)

4 if -16 != 0: --- True

5 D = 56\*56 - 4\*-16\*72

D = 7744

6 if 7744 > 0: --- True

7 x1 = (-56 - sqrt(7744)) / (2\*-16)

x1 = 4.5

8 x2 = (-56 + sqrt(7744)) / (2\*-16)

x2 = -1.0

9 return [4.5, -1.0]

6. square\_equal(-64, -96, -36) = [-0.75]

3 def square\_equal(a=-64, b=-96, c=-36)

4 if -64 != 0: --- True

5 D = -96\*-96 - 4\*-64\*-36

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-0.75]

7. square\_equal(-2, 21, -74) = []

3 def square\_equal(a=-2, b=21, c=-74)

4 if -2 != 0: --- True

5 D = 21\*21 - 4\*-2\*-74

D = -151

6 if -151 > 0: --- False

10 elif -151 == 0: --- False

12 else:

13 return []

8. square\_equal(0, 16, -44) = [2.75]

3 def square\_equal(a=0, b=16, c=-44)

4 if 0 != 0: --- False

14 else:

15 if 16 != 0: --- True

16 return [2.75]

9. square\_equal(0, 0, -49) = []

3 def square\_equal(a=0, b=0, c=-49)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-3-16

1. gcd(28, 100) = 4

1 def gcd(x=28, y=100)

2 if 28 == 0: --- False

4 if 100 == 0: --- False

6 while 100 != 0: --- True

7 rem = 28 % 100

rem = 28

8 x = 100

9 y = 28

6 while 28 != 0: --- True

7 rem = 100 % 28

rem = 16

8 x = 28

9 y = 16

6 while 16 != 0: --- True

7 rem = 28 % 16

rem = 12

8 x = 16

9 y = 12

6 while 12 != 0: --- True

7 rem = 16 % 12

rem = 4

8 x = 12

9 y = 4

6 while 4 != 0: --- True

7 rem = 12 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, 4) = 4

1 def gcd(x=0, y=4)

2 if 0 == 0: --- True

3 return 4

3. gcd(72, 0) = 72

1 def gcd(x=72, y=0)

2 if 72 == 0: --- False

4 if 0 == 0: --- True

5 return 72

4. hex(187) = 'BB'

3 def hex(number=187)

4 if 187 == 0: --- False

6 res = ''

7 while 187 > 0: --- True

8 digit = 187 % 16

digit = 11

9 res = DIGITS[11] + ''

res = 'B'

10 number = 187 // 16

number = 11

7 while 11 > 0: --- True

8 digit = 11 % 16

digit = 11

9 res = DIGITS[11] + 'B'

res = 'BB'

10 number = 11 // 16

number = 0

7 while 0 > 0: --- False

11 return 'BB'

5. square\_equal(-100, -10, 6) = [0.2, -0.3]

3 def square\_equal(a=-100, b=-10, c=6)

4 if -100 != 0: --- True

5 D = -10\*-10 - 4\*-100\*6

D = 2500

6 if 2500 > 0: --- True

7 x1 = (--10 - sqrt(2500)) / (2\*-100)

x1 = 0.2

8 x2 = (--10 + sqrt(2500)) / (2\*-100)

x2 = -0.3

9 return [0.2, -0.3]

6. square\_equal(-1, -2, -1) = [-1.0]

3 def square\_equal(a=-1, b=-2, c=-1)

4 if -1 != 0: --- True

5 D = -2\*-2 - 4\*-1\*-1

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-1.0]

7. square\_equal(13, -16, 14) = []

3 def square\_equal(a=13, b=-16, c=14)

4 if 13 != 0: --- True

5 D = -16\*-16 - 4\*13\*14

D = -472

6 if -472 > 0: --- False

10 elif -472 == 0: --- False

12 else:

13 return []

8. square\_equal(0, 6, -15) = [2.5]

3 def square\_equal(a=0, b=6, c=-15)

4 if 0 != 0: --- False

14 else:

15 if 6 != 0: --- True

16 return [2.5]

9. square\_equal(0, 0, 29) = []

3 def square\_equal(a=0, b=0, c=29)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-3-17

1. gcd(52, 88) = 4

1 def gcd(x=52, y=88)

2 if 52 == 0: --- False

4 if 88 == 0: --- False

6 while 88 != 0: --- True

7 rem = 52 % 88

rem = 52

8 x = 88

9 y = 52

6 while 52 != 0: --- True

7 rem = 88 % 52

rem = 36

8 x = 52

9 y = 36

6 while 36 != 0: --- True

7 rem = 52 % 36

rem = 16

8 x = 36

9 y = 16

6 while 16 != 0: --- True

7 rem = 36 % 16

rem = 4

8 x = 16

9 y = 4

6 while 4 != 0: --- True

7 rem = 16 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, 73) = 73

1 def gcd(x=0, y=73)

2 if 0 == 0: --- True

3 return 73

3. gcd(91, 0) = 91

1 def gcd(x=91, y=0)

2 if 91 == 0: --- False

4 if 0 == 0: --- True

5 return 91

4. hex(228) = 'E4'

3 def hex(number=228)

4 if 228 == 0: --- False

6 res = ''

7 while 228 > 0: --- True

8 digit = 228 % 16

digit = 4

9 res = DIGITS[4] + ''

res = '4'

10 number = 228 // 16

number = 14

7 while 14 > 0: --- True

8 digit = 14 % 16

digit = 14

9 res = DIGITS[14] + '4'

res = 'E4'

10 number = 14 // 16

number = 0

7 while 0 > 0: --- False

11 return 'E4'

5. square\_equal(25, 65, 22) = [-2.2, -0.4]

3 def square\_equal(a=25, b=65, c=22)

4 if 25 != 0: --- True

5 D = 65\*65 - 4\*25\*22

D = 2025

6 if 2025 > 0: --- True

7 x1 = (-65 - sqrt(2025)) / (2\*25)

x1 = -2.2

8 x2 = (-65 + sqrt(2025)) / (2\*25)

x2 = -0.4

9 return [-2.2, -0.4]

6. square\_equal(4, -28, 49) = [3.5]

3 def square\_equal(a=4, b=-28, c=49)

4 if 4 != 0: --- True

5 D = -28\*-28 - 4\*4\*49

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [3.5]

7. square\_equal(9, 12, 27) = []

3 def square\_equal(a=9, b=12, c=27)

4 if 9 != 0: --- True

5 D = 12\*12 - 4\*9\*27

D = -828

6 if -828 > 0: --- False

10 elif -828 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -69, -69) = [-1.0]

3 def square\_equal(a=0, b=-69, c=-69)

4 if 0 != 0: --- False

14 else:

15 if -69 != 0: --- True

16 return [-1.0]

9. square\_equal(0, 0, 53) = []

3 def square\_equal(a=0, b=0, c=53)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-3-18

1. gcd(52, 84) = 4

1 def gcd(x=52, y=84)

2 if 52 == 0: --- False

4 if 84 == 0: --- False

6 while 84 != 0: --- True

7 rem = 52 % 84

rem = 52

8 x = 84

9 y = 52

6 while 52 != 0: --- True

7 rem = 84 % 52

rem = 32

8 x = 52

9 y = 32

6 while 32 != 0: --- True

7 rem = 52 % 32

rem = 20

8 x = 32

9 y = 20

6 while 20 != 0: --- True

7 rem = 32 % 20

rem = 12

8 x = 20

9 y = 12

6 while 12 != 0: --- True

7 rem = 20 % 12

rem = 8

8 x = 12

9 y = 8

6 while 8 != 0: --- True

7 rem = 12 % 8

rem = 4

8 x = 8

9 y = 4

6 while 4 != 0: --- True

7 rem = 8 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, 13) = 13

1 def gcd(x=0, y=13)

2 if 0 == 0: --- True

3 return 13

3. gcd(96, 0) = 96

1 def gcd(x=96, y=0)

2 if 96 == 0: --- False

4 if 0 == 0: --- True

5 return 96

4. hex(164) = 'A4'

3 def hex(number=164)

4 if 164 == 0: --- False

6 res = ''

7 while 164 > 0: --- True

8 digit = 164 % 16

digit = 4

9 res = DIGITS[4] + ''

res = '4'

10 number = 164 // 16

number = 10

7 while 10 > 0: --- True

8 digit = 10 % 16

digit = 10

9 res = DIGITS[10] + '4'

res = 'A4'

10 number = 10 // 16

number = 0

7 while 0 > 0: --- False

11 return 'A4'

5. square\_equal(25, 40, 12) = [-1.2, -0.4]

3 def square\_equal(a=25,b=40,c=12)

4 if 25 != 0: --- True

5 D = 40\*40 - 4\*25\*12

D = 400

6 if 400 > 0: --- True

7 x1 = (-40 - sqrt(400)) / (2\*25)

x1 = -1.2

8 x2 = (-40 + sqrt(400)) / (2\*25)

x2 = -0.4

9 return [-1.2, -0.4]

6. square\_equal(-49, 98, -49) = [1.0]

3 def square\_equal(a=-49, b=98, c=-49)

4 if -49 != 0: --- True

5 D = 98\*98 - 4\*-49\*-49

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [1.0]

7. square\_equal(-7, -5, -4) = []

3 def square\_equal(a=-7,b=-5,c=-4)

4 if -7 != 0: --- True

5 D = -5\*-5 - 4\*-7\*-4

D = -87

6 if -87 > 0: --- False

10 elif -87 == 0: --- False

12 else:

13 return []

8. square\_equal(0, 8, 16) = [-2.0]

3 def square\_equal(a=0, b=8, c=16)

4 if 0 != 0: --- False

14 else:

15 if 8 != 0: --- True

16 return [-2.0]

9. square\_equal(0, 0, 52) = []

3 def square\_equal(a=0, b=0, c=52)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-3-19

1. gcd(54, 84) = 6

1 def gcd(x=54, y=84)

2 if 54 == 0: --- False

4 if 84 == 0: --- False

6 while 84 != 0: --- True

7 rem = 54 % 84

rem = 54

8 x = 84

9 y = 54

6 while 54 != 0: --- True

7 rem = 84 % 54

rem = 30

8 x = 54

9 y = 30

6 while 30 != 0: --- True

7 rem = 54 % 30

rem = 24

8 x = 30

9 y = 24

6 while 24 != 0: --- True

7 rem = 30 % 24

rem = 6

8 x = 24

9 y = 6

6 while 6 != 0: --- True

7 rem = 24 % 6

rem = 0

8 x = 6

9 y = 0

6 while 0 != 0: --- False

10 return 6

2. gcd(0, 38) = 38

1 def gcd(x=0, y=38)

2 if 0 == 0: --- True

3 return 38

3. gcd(73, 0) = 73

1 def gcd(x=73, y=0)

2 if 73 == 0: --- False

4 if 0 == 0: --- True

5 return 73

4. hex(167) = 'A7'

3 def hex(number=167)

4 if 167 == 0: --- False

6 res = ''

7 while 167 > 0: --- True

8 digit = 167 % 16

digit = 7

9 res = DIGITS[7] + ''

res = '7'

10 number = 167 // 16

number = 10

7 while 10 > 0: --- True

8 digit = 10 % 16

digit = 10

9 res = DIGITS[10] + '7'

res = 'A7'

10 number = 10 // 16

number = 0

7 while 0 > 0: --- False

11 return 'A7'

5. square\_equal(1, -27, 72) = [3.0, 24.0]

3 def square\_equal(a=1, b=-27, c=72)

4 if 1 != 0: --- True

5 D = -27\*-27 - 4\*1\*72

D = 441

6 if 441 > 0: --- True

7 x1 = (--27 - sqrt(441)) / (2\*1)

x1 = 3.0

8 x2 = (--27 + sqrt(441)) / (2\*1)

x2 = 24.0

9 return [3.0, 24.0]

6. square\_equal(16, -32, 16) = [1.0]

3 def square\_equal(a=16, b=-32, c=16)

4 if 16 != 0: --- True

5 D = -32\*-32 - 4\*16\*16

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [1.0]

7. square\_equal(11, -13, 8) = []

3 def square\_equal(a=11, b=-13, c=8)

4 if 11 != 0: --- True

5 D = -13\*-13 - 4\*11\*8

D = -183

6 if -183 > 0: --- False

10 elif -183 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -12, -51) = [-4.25]

3 def square\_equal(a=0, b=-12, c=-51)

4 if 0 != 0: --- False

14 else:

15 if -12 != 0: --- True

16 return [-4.25]

9. square\_equal(0, 0, 49) = []

3 def square\_equal(a=0, b=0, c=49)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-3-20

1. gcd(60, 85) = 5

1 def gcd(x=60, y=85)

2 if 60 == 0: --- False

4 if 85 == 0: --- False

6 while 85 != 0: --- True

7 rem = 60 % 85

rem = 60

8 x = 85

9 y = 60

6 while 60 != 0: --- True

7 rem = 85 % 60

rem = 25

8 x = 60

9 y = 25

6 while 25 != 0: --- True

7 rem = 60 % 25

rem = 10

8 x = 25

9 y = 10

6 while 10 != 0: --- True

7 rem = 25 % 10

rem = 5

8 x = 10

9 y = 5

6 while 5 != 0: --- True

7 rem = 10 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

2. gcd(0, 59) = 59

1 def gcd(x=0, y=59)

2 if 0 == 0: --- True

3 return 59

3. gcd(99, 0) = 99

1 def gcd(x=99, y=0)

2 if 99 == 0: --- False

4 if 0 == 0: --- True

5 return 99

4. hex(166) = 'A6'

3 def hex(number=166)

4 if 166 == 0: --- False

6 res = ''

7 while 166 > 0: --- True

8 digit = 166 % 16

digit = 6

9 res = DIGITS[6] + ''

res = '6'

10 number = 166 // 16

number = 10

7 while 10 > 0: --- True

8 digit = 10 % 16

digit = 10

9 res = DIGITS[10] + '6'

res = 'A6'

10 number = 10 // 16

number = 0

7 while 0 > 0: --- False

11 return 'A6'

5. square\_equal(-25, -50, 24) = [0.4, -2.4]

3 def square\_equal(a=-25, b=-50, c=24)

4 if -25 != 0: --- True

5 D = -50\*-50 - 4\*-25\*24

D = 4900

6 if 4900 > 0: --- True

7 x1 = (--50 - sqrt(4900)) / (2\*-25)

x1 = 0.4

8 x2 = (--50 + sqrt(4900)) / (2\*-25)

x2 = -2.4

9 return [0.4, -2.4]

6. square\_equal(-25, -90, -81) = [-1.8]

3 def square\_equal(a=-25, b=-90, c=-81)

4 if -25 != 0: --- True

5 D = -90\*-90 - 4\*-25\*-81

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-1.8]

7. square\_equal(-17, -16, -7) = []

3 def square\_equal(a=-17, b=-16, c=-7)

4 if -17 != 0: --- True

5 D = -16\*-16 - 4\*-17\*-7

D = -220

6 if -220 > 0: --- False

10 elif -220 == 0: --- False

12 else:

13 return []

8. square\_equal(0, 49, 98) = [-2.0]

3 def square\_equal(a=0, b=49, c=98)

4 if 0 != 0: --- False

14 else:

15 if 49 != 0: --- True

16 return [-2.0]

9. square\_equal(0, 0, 51) = []

3 def square\_equal(a=0, b=0, c=51)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []