Вариант: 1-2-1

1. gcd(55, 100) = 5

1 def gcd(x=55, y=100)

2 if 55 == 0: --- False

4 if 100 == 0: --- False

6 while 100 != 0: --- True

7 rem = 55 % 100

rem = 55

8 x = 100

9 y = 55

6 while 55 != 0: --- True

7 rem = 100 % 55

rem = 45

8 x = 55

9 y = 45

6 while 45 != 0: --- True

7 rem = 55 % 45

rem = 10

8 x = 45

9 y = 10

6 while 10 != 0: --- True

7 rem = 45 % 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, 13) = 13

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

2 if 0 == 0: --- True

3 return 13

3. gcd(89, 0) = 89

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

2 if 89 == 0: --- False

4 if 0 == 0: --- True

5 return 89

4. hex(215) = 'D7'

3 def hex(number=215)

4 if 215 == 0: --- False

6 res = ''

7 while 215 > 0: --- True

8 digit = 215 % 16

digit = 7

9 res = DIGITS[7] + ''

res = '7'

10 number = 215 // 16

number = 13

7 while 13 > 0: --- True

8 digit = 13 % 16

digit = 13

9 res = DIGITS[13] + '7'

res = 'D7'

10 number = 13 // 16

number = 0

7 while 0 > 0: --- False

11 return 'D7'

5. square\_equal(-1, -12, 28) = [2.0, -14.0]

3 def square\_equal(a=-1, b=-12, c=28)

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

5 D = -12\*-12 - 4\*-1\*28

D = 256

6 if 256 > 0: --- True

7 x1 = (--12 - sqrt(256)) / (2\*-1)

x1 = 2.0

8 x2 = (--12 + sqrt(256)) / (2\*-1)

x2 = -14.0

9 return [2.0, -14.0]

6. square\_equal(4, -20, 25) = [2.5]

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

4 if 4 != 0: --- True

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [2.5]

7. square\_equal(34, -29, 7) = []

3 def square\_equal(a=34, b=-29, c=7)

4 if 34 != 0: --- True

5 D = -29\*-29 - 4\*34\*7

D = -111

6 if -111 > 0: --- False

10 elif -111 == 0: --- False

12 else:

13 return []

8. square\_equal(0, 26, -65) = [2.5]

3 def square\_equal(a=0, b=26, c=-65)

4 if 0 != 0: --- False

14 else:

15 if 26 != 0: --- True

16 return [2.5]

9. square\_equal(0, 0, -83) = []

3 def square\_equal(a=0, b=0, c=-83)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-2

1. gcd(52, 72) = 4

1 def gcd(x=52, y=72)

2 if 52 == 0: --- False

4 if 72 == 0: --- False

6 while 72 != 0: --- True

7 rem = 52 % 72

rem = 52

8 x = 72

9 y = 52

6 while 52 != 0: --- True

7 rem = 72 % 52

rem = 20

8 x = 52

9 y = 20

6 while 20 != 0: --- True

7 rem = 52 % 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, 52) = 52

1 def gcd(x=0, y=52)

2 if 0 == 0: --- True

3 return 52

3. gcd(45, 0) = 45

1 def gcd(x=45, y=0)

2 if 45 == 0: --- False

4 if 0 == 0: --- True

5 return 45

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(1, -34, 33) = [1.0, 33.0]

3 def square\_equal(a=1, b=-34, c=33)

4 if 1 != 0: --- True

5 D = -34\*-34 - 4\*1\*33

D = 1024

6 if 1024 > 0: --- True

7 x1 = (--34 - sqrt(1024)) / (2\*1)

x1 = 1.0

8 x2 = (--34 + sqrt(1024)) / (2\*1)

x2 = 33.0

9 return [1.0, 33.0]

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(-40, 25, -6) = []

3 def square\_equal(a=-40, b=25, c=-6)

4 if -40 != 0: --- True

5 D = 25\*25 - 4\*-40\*-6

D = -335

6 if -335 > 0: --- False

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

12 else:

13 return []

8. square\_equal(0, -40, -6) = [-0.15]

3 def square\_equal(a=0, b=-40, c=-6)

4 if 0 != 0: --- False

14 else:

15 if -40 != 0: --- True

16 return [-0.15]

9. square\_equal(0, 0, 23) = []

3 def square\_equal(a=0, b=0, c=23)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-3

1. gcd(44, 80) = 4

1 def gcd(x=44, y=80)

2 if 44 == 0: --- False

4 if 80 == 0: --- False

6 while 80 != 0: --- True

7 rem = 44 % 80

rem = 44

8 x = 80

9 y = 44

6 while 44 != 0: --- True

7 rem = 80 % 44

rem = 36

8 x = 44

9 y = 36

6 while 36 != 0: --- True

7 rem = 44 % 36

rem = 8

8 x = 36

9 y = 8

6 while 8 != 0: --- True

7 rem = 36 % 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, 70) = 70

1 def gcd(x=0, y=70)

2 if 0 == 0: --- True

3 return 70

3. gcd(55, 0) = 55

1 def gcd(x=55, y=0)

2 if 55 == 0: --- False

4 if 0 == 0: --- True

5 return 55

4. hex(160) = 'A0'

3 def hex(number=160)

4 if 160 == 0: --- False

6 res = ''

7 while 160 > 0: --- True

8 digit = 160 % 16

digit = 0

9 res = DIGITS[0] + ''

res = '0'

10 number = 160 // 16

number = 10

7 while 10 > 0: --- True

8 digit = 10 % 16

digit = 10

9 res = DIGITS[10] + '0'

res = 'A0'

10 number = 10 // 16

number = 0

7 while 0 > 0: --- False

11 return 'A0'

5. square\_equal(-1, 40, 84) = [42.0, -2.0]

3 def square\_equal(a=-1, b=40, c=84)

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

5 D = 40\*40 - 4\*-1\*84

D = 1936

6 if 1936 > 0: --- True

7 x1 = (-40 - sqrt(1936)) / (2\*-1)

x1 = 42.0

8 x2 = (-40 + sqrt(1936)) / (2\*-1)

x2 = -2.0

9 return [42.0, -2.0]

6. square\_equal(-25, 70, -49) = [1.4]

3 def square\_equal(a=-25, b=70, c=-49)

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

5 D = 70\*70 - 4\*-25\*-49

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [1.4]

7. square\_equal(4, -2, 52) = []

3 def square\_equal(a=4, b=-2, c=52)

4 if 4 != 0: --- True

5 D = -2\*-2 - 4\*4\*52

D = -828

6 if -828 > 0: --- False

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

12 else:

13 return []

8. square\_equal(0, -5, 46) = [9.2]

3 def square\_equal(a=0, b=-5, c=46)

4 if 0 != 0: --- False

14 else:

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

16 return [9.2]

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: 1-2-4

1. gcd(45, 72) = 9

1 def gcd(x=45, y=72)

2 if 45 == 0: --- False

4 if 72 == 0: --- False

6 while 72 != 0: --- True

7 rem = 45 % 72

rem = 45

8 x = 72

9 y = 45

6 while 45 != 0: --- True

7 rem = 72 % 45

rem = 27

8 x = 45

9 y = 27

6 while 27 != 0: --- True

7 rem = 45 % 27

rem = 18

8 x = 27

9 y = 18

6 while 18 != 0: --- True

7 rem = 27 % 18

rem = 9

8 x = 18

9 y = 9

6 while 9 != 0: --- True

7 rem = 18 % 9

rem = 0

8 x = 9

9 y = 0

6 while 0 != 0: --- False

10 return 9

2. gcd(0, 93) = 93

1 def gcd(x=0, y=93)

2 if 0 == 0: --- True

3 return 93

3. gcd(8, 0) = 8

1 def gcd(x=8, y=0)

2 if 8 == 0: --- False

4 if 0 == 0: --- True

5 return 8

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(1, -27, 92) = [4.0, 23.0]

3 def square\_equal(a=1, b=-27, c=92)

4 if 1 != 0: --- True

5 D = -27\*-27 - 4\*1\*92

D = 361

6 if 361 > 0: --- True

7 x1 = (--27 - sqrt(361)) / (2\*1)

x1 = 4.0

8 x2 = (--27 + sqrt(361)) / (2\*1)

x2 = 23.0

9 return [4.0, 23.0]

6. square\_equal(36, -36, 9) = [0.5]

3 def square\_equal(a=36, b=-36, c=9)

4 if 36 != 0: --- True

5 D = -36\*-36 - 4\*36\*9

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [0.5]

7. square\_equal(-3, 5, -50) = []

3 def square\_equal(a=-3, b=5, c=-50)

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

5 D = 5\*5 - 4\*-3\*-50

D = -575

6 if -575 > 0: --- False

10 elif -575 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -24, -60) = [-2.5]

3 def square\_equal(a=0, b=-24, c=-60)

4 if 0 != 0: --- False

14 else:

15 if -24 != 0: --- True

16 return [-2.5]

9. square\_equal(0, 0, -37) = []

3 def square\_equal(a=0, b=0, c=-37)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-5

1. gcd(36, 56) = 4

1 def gcd(x=36, y=56)

2 if 36 == 0: --- False

4 if 56 == 0: --- False

6 while 56 != 0: --- True

7 rem = 36 % 56

rem = 36

8 x = 56

9 y = 36

6 while 36 != 0: --- True

7 rem = 56 % 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, 42) = 42

1 def gcd(x=0, y=42)

2 if 0 == 0: --- True

3 return 42

3. gcd(44, 0) = 44

1 def gcd(x=44, y=0)

2 if 44 == 0: --- False

4 if 0 == 0: --- True

5 return 44

4. hex(165) = 'A5'

3 def hex(number=165)

4 if 165 == 0: --- False

6 res = ''

7 while 165 > 0: --- True

8 digit = 165 % 16

digit = 5

9 res = DIGITS[5] + ''

res = '5'

10 number = 165 // 16

number = 10

7 while 10 > 0: --- True

8 digit = 10 % 16

digit = 10

9 res = DIGITS[10] + '5'

res = 'A5'

10 number = 10 // 16

number = 0

7 while 0 > 0: --- False

11 return 'A5'

5. square\_equal(-4, -24, 13) = [0.5, -6.5]

3 def square\_equal(a=-4, b=-24, c=13)

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

5 D = -24\*-24 - 4\*-4\*13

D = 784

6 if 784 > 0: --- True

7 x1 = (--24 - sqrt(784)) / (2\*-4)

x1 = 0.5

8 x2 = (--24 + sqrt(784)) / (2\*-4)

x2 = -6.5

9 return [0.5, -6.5]

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(-1, -2, -80) = []

3 def square\_equal(a=-1, b=-2, c=-80)

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

5 D = -2\*-2 - 4\*-1\*-80

D = -316

6 if -316 > 0: --- False

10 elif -316 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -12, -24) = [-2.0]

3 def square\_equal(a=0, b=-12, c=-24)

4 if 0 != 0: --- False

14 else:

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

16 return [-2.0]

9. square\_equal(0, 0, 19) = []

3 def square\_equal(a=0, b=0, c=19)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-6

1. gcd(75, 95) = 5

1 def gcd(x=75, y=95)

2 if 75 == 0: --- False

4 if 95 == 0: --- False

6 while 95 != 0: --- True

7 rem = 75 % 95

rem = 75

8 x = 95

9 y = 75

6 while 75 != 0: --- True

7 rem = 95 % 75

rem = 20

8 x = 75

9 y = 20

6 while 20 != 0: --- True

7 rem = 75 % 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, 46) = 46

1 def gcd(x=0, y=46)

2 if 0 == 0: --- True

3 return 46

3. gcd(32, 0) = 32

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

2 if 32 == 0: --- False

4 if 0 == 0: --- True

5 return 32

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(25, -85, 42) = [0.6, 2.8]

3 def square\_equal(a=25, b=-85, c=42)

4 if 25 != 0: --- True

5 D = -85\*-85 - 4\*25\*42

D = 3025

6 if 3025 > 0: --- True

7 x1 = (--85 - sqrt(3025)) / (2\*25)

x1 = 0.6

8 x2 = (--85 + sqrt(3025)) / (2\*25)

x2 = 2.8

9 return [0.6, 2.8]

6. square\_equal(-1, -18, -81) = [-9.0]

3 def square\_equal(a=-1, b=-18, c=-81)

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

5 D = -18\*-18 - 4\*-1\*-81

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-9.0]

7. square\_equal(4, 4, 51) = []

3 def square\_equal(a=4, b=4, c=51)

4 if 4 != 0: --- True

5 D = 4\*4 - 4\*4\*51

D = -800

6 if -800 > 0: --- False

10 elif -800 == 0: --- False

12 else:

13 return []

8. square\_equal(0, 25, -52) = [2.08]

3 def square\_equal(a=0, b=25, c=-52)

4 if 0 != 0: --- False

14 else:

15 if 25 != 0: --- True

16 return [2.08]

9. square\_equal(0, 0, -44) = []

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

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-7

1. gcd(56, 100) = 4

1 def gcd(x=56, y=100)

2 if 56 == 0: --- False

4 if 100 == 0: --- False

6 while 100 != 0: --- True

7 rem = 56 % 100

rem = 56

8 x = 100

9 y = 56

6 while 56 != 0: --- True

7 rem = 100 % 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, 3) = 3

1 def gcd(x=0, y=3)

2 if 0 == 0: --- True

3 return 3

3. gcd(14, 0) = 14

1 def gcd(x=14, y=0)

2 if 14 == 0: --- False

4 if 0 == 0: --- True

5 return 14

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(-4, 58, 96) = [16.0, -1.5]

3 def square\_equal(a=-4, b=58, c=96)

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

5 D = 58\*58 - 4\*-4\*96

D = 4900

6 if 4900 > 0: --- True

7 x1 = (-58 - sqrt(4900)) / (2\*-4)

x1 = 16.0

8 x2 = (-58 + sqrt(4900)) / (2\*-4)

x2 = -1.5

9 return [16.0, -1.5]

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(-53, 24, -3) = []

3 def square\_equal(a=-53, b=24, c=-3)

4 if -53 != 0: --- True

5 D = 24\*24 - 4\*-53\*-3

D = -60

6 if -60 > 0: --- False

10 elif -60 == 0: --- False

12 else:

13 return []

8. square\_equal(0, 5, -47) = [9.4]

3 def square\_equal(a=0, b=5, c=-47)

4 if 0 != 0: --- False

14 else:

15 if 5 != 0: --- True

16 return [9.4]

9. square\_equal(0, 0, -67) = []

3 def square\_equal(a=0, b=0, c=-67)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-8

1. gcd(65, 40) = 5

1 def gcd(x=65, y=40)

2 if 65 == 0: --- False

4 if 40 == 0: --- False

6 while 40 != 0: --- True

7 rem = 65 % 40

rem = 25

8 x = 40

9 y = 25

6 while 25 != 0: --- True

7 rem = 40 % 25

rem = 15

8 x = 25

9 y = 15

6 while 15 != 0: --- True

7 rem = 25 % 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, 99) = 99

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

2 if 0 == 0: --- True

3 return 99

3. gcd(70, 0) = 70

1 def gcd(x=70, y=0)

2 if 70 == 0: --- False

4 if 0 == 0: --- True

5 return 70

4. hex(242) = 'F2'

3 def hex(number=242)

4 if 242 == 0: --- False

6 res = ''

7 while 242 > 0: --- True

8 digit = 242 % 16

digit = 2

9 res = DIGITS[2] + ''

res = '2'

10 number = 242 // 16

number = 15

7 while 15 > 0: --- True

8 digit = 15 % 16

digit = 15

9 res = DIGITS[15] + '2'

res = 'F2'

10 number = 15 // 16

number = 0

7 while 0 > 0: --- False

11 return 'F2'

5. square\_equal(-25, -55, 60) = [0.8, -3.0]

3 def square\_equal(a=-25, b=-55, c=60)

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

5 D = -55\*-55 - 4\*-25\*60

D = 9025

6 if 9025 > 0: --- True

7 x1 = (--55 - sqrt(9025)) / (2\*-25)

x1 = 0.8

8 x2 = (--55 + sqrt(9025)) / (2\*-25)

x2 = -3.0

9 return [0.8, -3.0]

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(2, 7, 76) = []

3 def square\_equal(a=2, b=7, c=76)

4 if 2 != 0: --- True

5 D = 7\*7 - 4\*2\*76

D = -559

6 if -559 > 0: --- False

10 elif -559 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -8, -74) = [-9.25]

3 def square\_equal(a=0, b=-8, c=-74)

4 if 0 != 0: --- False

14 else:

15 if -8 != 0: --- True

16 return [-9.25]

9. square\_equal(0, 0, 48) = []

3 def square\_equal(a=0, b=0, c=48)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-9

1. gcd(25, 65) = 5

1 def gcd(x=25, y=65)

2 if 25 == 0: --- False

4 if 65 == 0: --- False

6 while 65 != 0: --- True

7 rem = 25 % 65

rem = 25

8 x = 65

9 y = 25

6 while 25 != 0: --- True

7 rem = 65 % 25

rem = 15

8 x = 25

9 y = 15

6 while 15 != 0: --- True

7 rem = 25 % 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, 88) = 88

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

2 if 0 == 0: --- True

3 return 88

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(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(4, 24, 32) = [-4.0, -2.0]

3 def square\_equal(a=4, b=24, c=32)

4 if 4 != 0: --- True

5 D = 24\*24 - 4\*4\*32

D = 64

6 if 64 > 0: --- True

7 x1 = (-24 - sqrt(64)) / (2\*4)

x1 = -4.0

8 x2 = (-24 + sqrt(64)) / (2\*4)

x2 = -2.0

9 return [-4.0, -2.0]

6. square\_equal(-1, 10, -25) = [5.0]

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

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

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [5.0]

7. square\_equal(-1, 6, -50) = []

3 def square\_equal(a=-1, b=6, c=-50)

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

5 D = 6\*6 - 4\*-1\*-50

D = -164

6 if -164 > 0: --- False

10 elif -164 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -25, -15) = [-0.6]

3 def square\_equal(a=0, b=-25, c=-15)

4 if 0 != 0: --- False

14 else:

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

16 return [-0.6]

9. square\_equal(0, 0, -63) = []

3 def square\_equal(a=0, b=0, c=-63)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-10

1. gcd(45, 70) = 5

1 def gcd(x=45, y=70)

2 if 45 == 0: --- False

4 if 70 == 0: --- False

6 while 70 != 0: --- True

7 rem = 45 % 70

rem = 45

8 x = 70

9 y = 45

6 while 45 != 0: --- True

7 rem = 70 % 45

rem = 25

8 x = 45

9 y = 25

6 while 25 != 0: --- True

7 rem = 45 % 25

rem = 20

8 x = 25

9 y = 20

6 while 20 != 0: --- True

7 rem = 25 % 20

rem = 5

8 x = 20

9 y = 5

6 while 5 != 0: --- True

7 rem = 20 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

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(236) = 'EC'

3 def hex(number=236)

4 if 236 == 0: --- False

6 res = ''

7 while 236 > 0: --- True

8 digit = 236 % 16

digit = 12

9 res = DIGITS[12] + ''

res = 'C'

10 number = 236 // 16

number = 14

7 while 14 > 0: --- True

8 digit = 14 % 16

digit = 14

9 res = DIGITS[14] + 'C'

res = 'EC'

10 number = 14 // 16

number = 0

7 while 0 > 0: --- False

11 return 'EC'

5. square\_equal(-4, 44, 23) = [11.5, -0.5]

3 def square\_equal(a=-4, b=44, c=23)

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

5 D = 44\*44 - 4\*-4\*23

D = 2304

6 if 2304 > 0: --- True

7 x1 = (-44 - sqrt(2304)) / (2\*-4)

x1 = 11.5

8 x2 = (-44 + sqrt(2304)) / (2\*-4)

x2 = -0.5

9 return [11.5, -0.5]

6. square\_equal(25, 80, 64) = [-1.6]

3 def square\_equal(a=25, b=80, c=64)

4 if 25 != 0: --- True

5 D = 80\*80 - 4\*25\*64

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-1.6]

7. square\_equal(-55, -5, -1) = []

3 def square\_equal(a=-55, b=-5, c=-1)

4 if -55 != 0: --- True

5 D = -5\*-5 - 4\*-55\*-1

D = -195

6 if -195 > 0: --- False

10 elif -195 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -73, 73) = [1.0]

3 def square\_equal(a=0, b=-73, c=73)

4 if 0 != 0: --- False

14 else:

15 if -73 != 0: --- True

16 return [1.0]

9. square\_equal(0, 0, 86) = []

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

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-11

1. gcd(25, 90) = 5

1 def gcd(x=25, y=90)

2 if 25 == 0: --- False

4 if 90 == 0: --- False

6 while 90 != 0: --- True

7 rem = 25 % 90

rem = 25

8 x = 90

9 y = 25

6 while 25 != 0: --- True

7 rem = 90 % 25

rem = 15

8 x = 25

9 y = 15

6 while 15 != 0: --- True

7 rem = 25 % 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, 65) = 65

1 def gcd(x=0, y=65)

2 if 0 == 0: --- True

3 return 65

3. gcd(39, 0) = 39

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

2 if 39 == 0: --- False

4 if 0 == 0: --- True

5 return 39

4. hex(202) = 'CA'

3 def hex(number=202)

4 if 202 == 0: --- False

6 res = ''

7 while 202 > 0: --- True

8 digit = 202 % 16

digit = 10

9 res = DIGITS[10] + ''

res = 'A'

10 number = 202 // 16

number = 12

7 while 12 > 0: --- True

8 digit = 12 % 16

digit = 12

9 res = DIGITS[12] + 'A'

res = 'CA'

10 number = 12 // 16

number = 0

7 while 0 > 0: --- False

11 return 'CA'

5. square\_equal(-1, 43, 44) = [44.0, -1.0]

3 def square\_equal(a=-1, b=43, c=44)

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

5 D = 43\*43 - 4\*-1\*44

D = 2025

6 if 2025 > 0: --- True

7 x1 = (-43 - sqrt(2025)) / (2\*-1)

x1 = 44.0

8 x2 = (-43 + sqrt(2025)) / (2\*-1)

x2 = -1.0

9 return [44.0, -1.0]

6. square\_equal(4, 36, 81) = [-4.5]

3 def square\_equal(a=4, b=36, c=81)

4 if 4 != 0: --- True

5 D = 36\*36 - 4\*4\*81

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-4.5]

7. square\_equal(-1, 18, -91) = []

3 def square\_equal(a=-1, b=18, c=-91)

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

5 D = 18\*18 - 4\*-1\*-91

D = -40

6 if -40 > 0: --- False

10 elif -40 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -40, 88) = [2.2]

3 def square\_equal(a=0, b=-40, c=88)

4 if 0 != 0: --- False

14 else:

15 if -40 != 0: --- True

16 return [2.2]

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: 1-2-12

1. gcd(84, 52) = 4

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

2 if 84 == 0: --- False

4 if 52 == 0: --- False

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, 10) = 10

1 def gcd(x=0, y=10)

2 if 0 == 0: --- True

3 return 10

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(163) = 'A3'

3 def hex(number=163)

4 if 163 == 0: --- False

6 res = ''

7 while 163 > 0: --- True

8 digit = 163 % 16

digit = 3

9 res = DIGITS[3] + ''

res = '3'

10 number = 163 // 16

number = 10

7 while 10 > 0: --- True

8 digit = 10 % 16

digit = 10

9 res = DIGITS[10] + '3'

res = 'A3'

10 number = 10 // 16

number = 0

7 while 0 > 0: --- False

11 return 'A3'

5. square\_equal(-4, 4, 35) = [3.5, -2.5]

3 def square\_equal(a=-4, b=4, c=35)

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

5 D = 4\*4 - 4\*-4\*35

D = 576

6 if 576 > 0: --- True

7 x1 = (-4 - sqrt(576)) / (2\*-4)

x1 = 3.5

8 x2 = (-4 + sqrt(576)) / (2\*-4)

x2 = -2.5

9 return [3.5, -2.5]

6. square\_equal(25, 50, 25) = [-1.0]

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

4 if 25 != 0: --- True

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-1.0]

7. square\_equal(1, 4, 90) = []

3 def square\_equal(a=1, b=4, c=90)

4 if 1 != 0: --- True

5 D = 4\*4 - 4\*1\*90

D = -344

6 if -344 > 0: --- False

10 elif -344 == 0: --- False

12 else:

13 return []

8. square\_equal(0, 2, -19) = [9.5]

3 def square\_equal(a=0, b=2, c=-19)

4 if 0 != 0: --- False

14 else:

15 if 2 != 0: --- True

16 return [9.5]

9. square\_equal(0, 0, -68) = []

3 def square\_equal(a=0, b=0, c=-68)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-13

1. gcd(20, 32) = 4

1 def gcd(x=20, y=32)

2 if 20 == 0: --- False

4 if 32 == 0: --- False

6 while 32 != 0: --- True

7 rem = 20 % 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, 41) = 41

1 def gcd(x=0, y=41)

2 if 0 == 0: --- True

3 return 41

3. gcd(87, 0) = 87

1 def gcd(x=87, y=0)

2 if 87 == 0: --- False

4 if 0 == 0: --- True

5 return 87

4. hex(221) = 'DD'

3 def hex(number=221)

4 if 221 == 0: --- False

6 res = ''

7 while 221 > 0: --- True

8 digit = 221 % 16

digit = 13

9 res = DIGITS[13] + ''

res = 'D'

10 number = 221 // 16

number = 13

7 while 13 > 0: --- True

8 digit = 13 % 16

digit = 13

9 res = DIGITS[13] + 'D'

res = 'DD'

10 number = 13 // 16

number = 0

7 while 0 > 0: --- False

11 return 'DD'

5. square\_equal(-1, 40, 41) = [41.0, -1.0]

3 def square\_equal(a=-1, b=40, c=41)

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

5 D = 40\*40 - 4\*-1\*41

D = 1764

6 if 1764 > 0: --- True

7 x1 = (-40 - sqrt(1764)) / (2\*-1)

x1 = 41.0

8 x2 = (-40 + sqrt(1764)) / (2\*-1)

x2 = -1.0

9 return [41.0, -1.0]

6. square\_equal(25, -40, 16) = [0.8]

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

4 if 25 != 0: --- True

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [0.8]

7. square\_equal(-12, 4, -1) = []

3 def square\_equal(a=-12, b=4, c=-1)

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

5 D = 4\*4 - 4\*-12\*-1

D = -32

6 if -32 > 0: --- False

10 elif -32 == 0: --- False

12 else:

13 return []

8. square\_equal(0, 98, 98) = [-1.0]

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

4 if 0 != 0: --- False

14 else:

15 if 98 != 0: --- True

16 return [-1.0]

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: 1-2-14

1. gcd(90, 65) = 5

1 def gcd(x=90, y=65)

2 if 90 == 0: --- False

4 if 65 == 0: --- False

6 while 65 != 0: --- True

7 rem = 90 % 65

rem = 25

8 x = 65

9 y = 25

6 while 25 != 0: --- True

7 rem = 65 % 25

rem = 15

8 x = 25

9 y = 15

6 while 15 != 0: --- True

7 rem = 25 % 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, 56) = 56

1 def gcd(x=0, y=56)

2 if 0 == 0: --- True

3 return 56

3. gcd(29, 0) = 29

1 def gcd(x=29, y=0)

2 if 29 == 0: --- False

4 if 0 == 0: --- True

5 return 29

4. hex(172) = 'AC'

3 def hex(number=172)

4 if 172 == 0: --- False

6 res = ''

7 while 172 > 0: --- True

8 digit = 172 % 16

digit = 12

9 res = DIGITS[12] + ''

res = 'C'

10 number = 172 // 16

number = 10

7 while 10 > 0: --- True

8 digit = 10 % 16

digit = 10

9 res = DIGITS[10] + 'C'

res = 'AC'

10 number = 10 // 16

number = 0

7 while 0 > 0: --- False

11 return 'AC'

5. square\_equal(-4, -40, 96) = [2.0, -12.0]

3 def square\_equal(a=-4, b=-40, c=96)

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

5 D = -40\*-40 - 4\*-4\*96

D = 3136

6 if 3136 > 0: --- True

7 x1 = (--40 - sqrt(3136)) / (2\*-4)

x1 = 2.0

8 x2 = (--40 + sqrt(3136)) / (2\*-4)

x2 = -12.0

9 return [2.0, -12.0]

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, -8, 54) = []

3 def square\_equal(a=3, b=-8, c=54)

4 if 3 != 0: --- True

5 D = -8\*-8 - 4\*3\*54

D = -584

6 if -584 > 0: --- False

10 elif -584 == 0: --- False

12 else:

13 return []

8. square\_equal(0, 20, 85) = [-4.25]

3 def square\_equal(a=0, b=20, c=85)

4 if 0 != 0: --- False

14 else:

15 if 20 != 0: --- True

16 return [-4.25]

9. square\_equal(0, 0, -99) = []

3 def square\_equal(a=0, b=0, c=-99)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-15

1. gcd(28, 44) = 4

1 def gcd(x=28, y=44)

2 if 28 == 0: --- False

4 if 44 == 0: --- False

6 while 44 != 0: --- True

7 rem = 28 % 44

rem = 28

8 x = 44

9 y = 28

6 while 28 != 0: --- True

7 rem = 44 % 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, 47) = 47

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

2 if 0 == 0: --- True

3 return 47

3. gcd(27, 0) = 27

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

2 if 27 == 0: --- False

4 if 0 == 0: --- True

5 return 27

4. hex(196) = 'C4'

3 def hex(number=196)

4 if 196 == 0: --- False

6 res = ''

7 while 196 > 0: --- True

8 digit = 196 % 16

digit = 4

9 res = DIGITS[4] + ''

res = '4'

10 number = 196 // 16

number = 12

7 while 12 > 0: --- True

8 digit = 12 % 16

digit = 12

9 res = DIGITS[12] + '4'

res = 'C4'

10 number = 12 // 16

number = 0

7 while 0 > 0: --- False

11 return 'C4'

5. square\_equal(-25, -25, 14) = [0.4, -1.4]

3 def square\_equal(a=-25, b=-25, c=14)

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

5 D = -25\*-25 - 4\*-25\*14

D = 2025

6 if 2025 > 0: --- True

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

x1 = 0.4

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

x2 = -1.4

9 return [0.4, -1.4]

6. square\_equal(25, -80, 64) = [1.6]

3 def square\_equal(a=25, b=-80, c=64)

4 if 25 != 0: --- True

5 D = -80\*-80 - 4\*25\*64

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [1.6]

7. square\_equal(-64, 17, -2) = []

3 def square\_equal(a=-64, b=17, c=-2)

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

5 D = 17\*17 - 4\*-64\*-2

D = -223

6 if -223 > 0: --- False

10 elif -223 == 0: --- False

12 else:

13 return []

8. square\_equal(0, 8, -46) = [5.75]

3 def square\_equal(a=0, b=8, c=-46)

4 if 0 != 0: --- False

14 else:

15 if 8 != 0: --- True

16 return [5.75]

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

3 def square\_equal(a=0, b=0, c=9)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-16

1. gcd(60, 76) = 4

1 def gcd(x=60, y=76)

2 if 60 == 0: --- False

4 if 76 == 0: --- False

6 while 76 != 0: --- True

7 rem = 60 % 76

rem = 60

8 x = 76

9 y = 60

6 while 60 != 0: --- True

7 rem = 76 % 60

rem = 16

8 x = 60

9 y = 16

6 while 16 != 0: --- True

7 rem = 60 % 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, 22) = 22

1 def gcd(x=0, y=22)

2 if 0 == 0: --- True

3 return 22

3. gcd(80, 0) = 80

1 def gcd(x=80, y=0)

2 if 80 == 0: --- False

4 if 0 == 0: --- True

5 return 80

4. hex(198) = 'C6'

3 def hex(number=198)

4 if 198 == 0: --- False

6 res = ''

7 while 198 > 0: --- True

8 digit = 198 % 16

digit = 6

9 res = DIGITS[6] + ''

res = '6'

10 number = 198 // 16

number = 12

7 while 12 > 0: --- True

8 digit = 12 % 16

digit = 12

9 res = DIGITS[12] + '6'

res = 'C6'

10 number = 12 // 16

number = 0

7 while 0 > 0: --- False

11 return 'C6'

5. square\_equal(-25, -40, 65) = [1.0, -2.6]

3 def square\_equal(a=-25, b=-40, c=65)

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

5 D = -40\*-40 - 4\*-25\*65

D = 8100

6 if 8100 > 0: --- True

7 x1 = (--40 - sqrt(8100)) / (2\*-25)

x1 = 1.0

8 x2 = (--40 + sqrt(8100)) / (2\*-25)

x2 = -2.6

9 return [1.0, -2.6]

6. square\_equal(16, 24, 9) = [-0.75]

3 def square\_equal(a=16, b=24, c=9)

4 if 16 != 0: --- True

5 D = 24\*24 - 4\*16\*9

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-0.75]

7. square\_equal(1, 8, 66) = []

3 def square\_equal(a=1, b=8, c=66)

4 if 1 != 0: --- True

5 D = 8\*8 - 4\*1\*66

D = -200

6 if -200 > 0: --- False

10 elif -200 == 0: --- False

12 else:

13 return []

8. square\_equal(0, 2, -96) = [48.0]

3 def square\_equal(a=0, b=2, c=-96)

4 if 0 != 0: --- False

14 else:

15 if 2 != 0: --- True

16 return [48.0]

9. square\_equal(0, 0, -10) = []

3 def square\_equal(a=0, b=0, c=-10)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-17

1. gcd(35, 95) = 5

1 def gcd(x=35, y=95)

2 if 35 == 0: --- False

4 if 95 == 0: --- False

6 while 95 != 0: --- True

7 rem = 35 % 95

rem = 35

8 x = 95

9 y = 35

6 while 35 != 0: --- True

7 rem = 95 % 35

rem = 25

8 x = 35

9 y = 25

6 while 25 != 0: --- True

7 rem = 35 % 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, 34) = 34

1 def gcd(x=0, y=34)

2 if 0 == 0: --- True

3 return 34

3. gcd(52, 0) = 52

1 def gcd(x=52, y=0)

2 if 52 == 0: --- False

4 if 0 == 0: --- True

5 return 52

4. hex(182) = 'B6'

3 def hex(number=182)

4 if 182 == 0: --- False

6 res = ''

7 while 182 > 0: --- True

8 digit = 182 % 16

digit = 6

9 res = DIGITS[6] + ''

res = '6'

10 number = 182 // 16

number = 11

7 while 11 > 0: --- True

8 digit = 11 % 16

digit = 11

9 res = DIGITS[11] + '6'

res = 'B6'

10 number = 11 // 16

number = 0

7 while 0 > 0: --- False

11 return 'B6'

5. square\_equal(9, 36, 27) = [-3.0, -1.0]

3 def square\_equal(a=9, b=36, c=27)

4 if 9 != 0: --- True

5 D = 36\*36 - 4\*9\*27

D = 324

6 if 324 > 0: --- True

7 x1 = (-36 - sqrt(324)) / (2\*9)

x1 = -3.0

8 x2 = (-36 + sqrt(324)) / (2\*9)

x2 = -1.0

9 return [-3.0, -1.0]

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(3, -24, 49) = []

3 def square\_equal(a=3, b=-24, c=49)

4 if 3 != 0: --- True

5 D = -24\*-24 - 4\*3\*49

D = -12

6 if -12 > 0: --- False

10 elif -12 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -3, 99) = [33.0]

3 def square\_equal(a=0, b=-3, c=99)

4 if 0 != 0: --- False

14 else:

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

16 return [33.0]

9. square\_equal(0, 0, 38) = []

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

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-18

1. gcd(52, 92) = 4

1 def gcd(x=52, y=92)

2 if 52 == 0: --- False

4 if 92 == 0: --- False

6 while 92 != 0: --- True

7 rem = 52 % 92

rem = 52

8 x = 92

9 y = 52

6 while 52 != 0: --- True

7 rem = 92 % 52

rem = 40

8 x = 52

9 y = 40

6 while 40 != 0: --- True

7 rem = 52 % 40

rem = 12

8 x = 40

9 y = 12

6 while 12 != 0: --- True

7 rem = 40 % 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, 31) = 31

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

2 if 0 == 0: --- True

3 return 31

3. gcd(100, 0) = 100

1 def gcd(x=100, y=0)

2 if 100 == 0: --- False

4 if 0 == 0: --- True

5 return 100

4. hex(189) = 'BD'

3 def hex(number=189)

4 if 189 == 0: --- False

6 res = ''

7 while 189 > 0: --- True

8 digit = 189 % 16

digit = 13

9 res = DIGITS[13] + ''

res = 'D'

10 number = 189 // 16

number = 11

7 while 11 > 0: --- True

8 digit = 11 % 16

digit = 11

9 res = DIGITS[11] + 'D'

res = 'BD'

10 number = 11 // 16

number = 0

7 while 0 > 0: --- False

11 return 'BD'

5. square\_equal(16, 92, 90) = [-4.5, -1.25]

3 def square\_equal(a=16, b=92, c=90)

4 if 16 != 0: --- True

5 D = 92\*92 - 4\*16\*90

D = 2704

6 if 2704 > 0: --- True

7 x1 = (-92 - sqrt(2704)) / (2\*16)

x1 = -4.5

8 x2 = (-92 + sqrt(2704)) / (2\*16)

x2 = -1.25

9 return [-4.5, -1.25]

6. square\_equal(-1, 4, -4) = [2.0]

3 def square\_equal(a=-1, b=4, c=-4)

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

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [2.0]

7. square\_equal(-22, 16, -3) = []

3 def square\_equal(a=-22, b=16, c=-3)

4 if -22 != 0: --- True

5 D = 16\*16 - 4\*-22\*-3

D = -8

6 if -8 > 0: --- False

10 elif -8 == 0: --- False

12 else:

13 return []

8. square\_equal(0, 16, 24) = [-1.5]

3 def square\_equal(a=0, b=16, c=24)

4 if 0 != 0: --- False

14 else:

15 if 16 != 0: --- True

16 return [-1.5]

9. square\_equal(0, 0, 6) = []

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

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-19

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, 38) = 38

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

2 if 0 == 0: --- True

3 return 38

3. gcd(5, 0) = 5

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

2 if 5 == 0: --- False

4 if 0 == 0: --- True

5 return 5

4. hex(218) = 'DA'

3 def hex(number=218)

4 if 218 == 0: --- False

6 res = ''

7 while 218 > 0: --- True

8 digit = 218 % 16

digit = 10

9 res = DIGITS[10] + ''

res = 'A'

10 number = 218 // 16

number = 13

7 while 13 > 0: --- True

8 digit = 13 % 16

digit = 13

9 res = DIGITS[13] + 'A'

res = 'DA'

10 number = 13 // 16

number = 0

7 while 0 > 0: --- False

11 return 'DA'

5. square\_equal(-16, -8, 63) = [1.75, -2.25]

3 def square\_equal(a=-16, b=-8, c=63)

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

5 D = -8\*-8 - 4\*-16\*63

D = 4096

6 if 4096 > 0: --- True

7 x1 = (--8 - sqrt(4096)) / (2\*-16)

x1 = 1.75

8 x2 = (--8 + sqrt(4096)) / (2\*-16)

x2 = -2.25

9 return [1.75, -2.25]

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(15, 14, 5) = []

3 def square\_equal(a=15, b=14, c=5)

4 if 15 != 0: --- True

5 D = 14\*14 - 4\*15\*5

D = -104

6 if -104 > 0: --- False

10 elif -104 == 0: --- False

12 else:

13 return []

8. square\_equal(0, 8, -96) = [12.0]

3 def square\_equal(a=0, b=8, c=-96)

4 if 0 != 0: --- False

14 else:

15 if 8 != 0: --- True

16 return [12.0]

9. square\_equal(0, 0, 36) = []

3 def square\_equal(a=0, b=0, c=36)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-20

1. gcd(32, 76) = 4

1 def gcd(x=32, y=76)

2 if 32 == 0: --- False

4 if 76 == 0: --- False

6 while 76 != 0: --- True

7 rem = 32 % 76

rem = 32

8 x = 76

9 y = 32

6 while 32 != 0: --- True

7 rem = 76 % 32

rem = 12

8 x = 32

9 y = 12

6 while 12 != 0: --- True

7 rem = 32 % 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, 83) = 83

1 def gcd(x=0, y=83)

2 if 0 == 0: --- True

3 return 83

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(227) = 'E3'

3 def hex(number=227)

4 if 227 == 0: --- False

6 res = ''

7 while 227 > 0: --- True

8 digit = 227 % 16

digit = 3

9 res = DIGITS[3] + ''

res = '3'

10 number = 227 // 16

number = 14

7 while 14 > 0: --- True

8 digit = 14 % 16

digit = 14

9 res = DIGITS[14] + '3'

res = 'E3'

10 number = 14 // 16

number = 0

7 while 0 > 0: --- False

11 return 'E3'

5. square\_equal(-4, 8, 21) = [3.5, -1.5]

3 def square\_equal(a=-4, b=8, c=21)

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

5 D = 8\*8 - 4\*-4\*21

D = 400

6 if 400 > 0: --- True

7 x1 = (-8 - sqrt(400)) / (2\*-4)

x1 = 3.5

8 x2 = (-8 + sqrt(400)) / (2\*-4)

x2 = -1.5

9 return [3.5, -1.5]

6. square\_equal(-25, -50, -25) = [-1.0]

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

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

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-1.0]

7. square\_equal(-1, -7, -91) = []

3 def square\_equal(a=-1, b=-7, c=-91)

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

5 D = -7\*-7 - 4\*-1\*-91

D = -315

6 if -315 > 0: --- False

10 elif -315 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -4, 83) = [20.75]

3 def square\_equal(a=0, b=-4, c=83)

4 if 0 != 0: --- False

14 else:

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

16 return [20.75]

9. square\_equal(0, 0, -39) = []

3 def square\_equal(a=0, b=0, c=-39)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-21

1. gcd(48, 68) = 4

1 def gcd(x=48, y=68)

2 if 48 == 0: --- False

4 if 68 == 0: --- False

6 while 68 != 0: --- True

7 rem = 48 % 68

rem = 48

8 x = 68

9 y = 48

6 while 48 != 0: --- True

7 rem = 68 % 48

rem = 20

8 x = 48

9 y = 20

6 while 20 != 0: --- True

7 rem = 48 % 20

rem = 8

8 x = 20

9 y = 8

6 while 8 != 0: --- True

7 rem = 20 % 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, 57) = 57

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

2 if 0 == 0: --- True

3 return 57

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(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(1, 4, 4) = [-2.0]

3 def square\_equal(a=1, b=4, c=4)

4 if 1 != 0: --- True

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-2.0]

6. square\_equal(25, 40, 16) = [-0.8]

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

4 if 25 != 0: --- True

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-0.8]

7. square\_equal(-6, -27, -34) = []

3 def square\_equal(a=-6, b=-27, c=-34)

4 if -6 != 0: --- True

5 D = -27\*-27 - 4\*-6\*-34

D = -87

6 if -87 > 0: --- False

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

12 else:

13 return []

8. square\_equal(0, -2, -14) = [-7.0]

3 def square\_equal(a=0, b=-2, c=-14)

4 if 0 != 0: --- False

14 else:

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

16 return [-7.0]

9. square\_equal(0, 0, 70) = []

3 def square\_equal(a=0, b=0, c=70)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-22

1. gcd(40, 55) = 5

1 def gcd(x=40, y=55)

2 if 40 == 0: --- False

4 if 55 == 0: --- False

6 while 55 != 0: --- True

7 rem = 40 % 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, 40) = 40

1 def gcd(x=0, y=40)

2 if 0 == 0: --- True

3 return 40

3. gcd(92, 0) = 92

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

2 if 92 == 0: --- False

4 if 0 == 0: --- True

5 return 92

4. hex(231) = 'E7'

3 def hex(number=231)

4 if 231 == 0: --- False

6 res = ''

7 while 231 > 0: --- True

8 digit = 231 % 16

digit = 7

9 res = DIGITS[7] + ''

res = '7'

10 number = 231 // 16

number = 14

7 while 14 > 0: --- True

8 digit = 14 % 16

digit = 14

9 res = DIGITS[14] + '7'

res = 'E7'

10 number = 14 // 16

number = 0

7 while 0 > 0: --- False

11 return 'E7'

5. square\_equal(-1, 2, 48) = [8.0, -6.0]

3 def square\_equal(a=-1, b=2, c=48)

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

5 D = 2\*2 - 4\*-1\*48

D = 196

6 if 196 > 0: --- True

7 x1 = (-2 - sqrt(196)) / (2\*-1)

x1 = 8.0

8 x2 = (-2 + sqrt(196)) / (2\*-1)

x2 = -6.0

9 return [8.0, -6.0]

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(-26, -8, -8) = []

3 def square\_equal(a=-26, b=-8, c=-8)

4 if -26 != 0: --- True

5 D = -8\*-8 - 4\*-26\*-8

D = -768

6 if -768 > 0: --- False

10 elif -768 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -1, -96) = [-96.0]

3 def square\_equal(a=0, b=-1, c=-96)

4 if 0 != 0: --- False

14 else:

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

16 return [-96.0]

9. square\_equal(0, 0, -91) = []

3 def square\_equal(a=0, b=0, c=-91)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-23

1. gcd(63, 99) = 9

1 def gcd(x=63, y=99)

2 if 63 == 0: --- False

4 if 99 == 0: --- False

6 while 99 != 0: --- True

7 rem = 63 % 99

rem = 63

8 x = 99

9 y = 63

6 while 63 != 0: --- True

7 rem = 99 % 63

rem = 36

8 x = 63

9 y = 36

6 while 36 != 0: --- True

7 rem = 63 % 36

rem = 27

8 x = 36

9 y = 27

6 while 27 != 0: --- True

7 rem = 36 % 27

rem = 9

8 x = 27

9 y = 9

6 while 9 != 0: --- True

7 rem = 27 % 9

rem = 0

8 x = 9

9 y = 0

6 while 0 != 0: --- False

10 return 9

2. gcd(0, 16) = 16

1 def gcd(x=0, y=16)

2 if 0 == 0: --- True

3 return 16

3. gcd(33, 0) = 33

1 def gcd(x=33, y=0)

2 if 33 == 0: --- False

4 if 0 == 0: --- True

5 return 33

4. hex(171) = 'AB'

3 def hex(number=171)

4 if 171 == 0: --- False

6 res = ''

7 while 171 > 0: --- True

8 digit = 171 % 16

digit = 11

9 res = DIGITS[11] + ''

res = 'B'

10 number = 171 // 16

number = 10

7 while 10 > 0: --- True

8 digit = 10 % 16

digit = 10

9 res = DIGITS[10] + 'B'

res = 'AB'

10 number = 10 // 16

number = 0

7 while 0 > 0: --- False

11 return 'AB'

5. square\_equal(-4, -22, 26) = [1.0, -6.5]

3 def square\_equal(a=-4, b=-22, c=26)

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

5 D = -22\*-22 - 4\*-4\*26

D = 900

6 if 900 > 0: --- True

7 x1 = (--22 - sqrt(900)) / (2\*-4)

x1 = 1.0

8 x2 = (--22 + sqrt(900)) / (2\*-4)

x2 = -6.5

9 return [1.0, -6.5]

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(3, 1, 59) = []

3 def square\_equal(a=3, b=1, c=59)

4 if 3 != 0: --- True

5 D = 1\*1 - 4\*3\*59

D = -707

6 if -707 > 0: --- False

10 elif -707 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -25, 19) = [0.76]

3 def square\_equal(a=0, b=-25, c=19)

4 if 0 != 0: --- False

14 else:

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

16 return [0.76]

9. square\_equal(0, 0, 5) = []

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

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-24

1. gcd(56, 91) = 7

1 def gcd(x=56, y=91)

2 if 56 == 0: --- False

4 if 91 == 0: --- False

6 while 91 != 0: --- True

7 rem = 56 % 91

rem = 56

8 x = 91

9 y = 56

6 while 56 != 0: --- True

7 rem = 91 % 56

rem = 35

8 x = 56

9 y = 35

6 while 35 != 0: --- True

7 rem = 56 % 35

rem = 21

8 x = 35

9 y = 21

6 while 21 != 0: --- True

7 rem = 35 % 21

rem = 14

8 x = 21

9 y = 14

6 while 14 != 0: --- True

7 rem = 21 % 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, 6) = 6

1 def gcd(x=0, y=6)

2 if 0 == 0: --- True

3 return 6

3. gcd(41, 0) = 41

1 def gcd(x=41, y=0)

2 if 41 == 0: --- False

4 if 0 == 0: --- True

5 return 41

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(-25, 35, 8) = [1.6, -0.2]

3 def square\_equal(a=-25, b=35, c=8)

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

5 D = 35\*35 - 4\*-25\*8

D = 2025

6 if 2025 > 0: --- True

7 x1 = (-35 - sqrt(2025)) / (2\*-25)

x1 = 1.6

8 x2 = (-35 + sqrt(2025)) / (2\*-25)

x2 = -0.2

9 return [1.6, -0.2]

6. square\_equal(-4, -32, -64) = [-4.0]

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

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

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-4.0]

7. square\_equal(-18, -7, -4) = []

3 def square\_equal(a=-18, b=-7, c=-4)

4 if -18 != 0: --- True

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

D = -239

6 if -239 > 0: --- False

10 elif -239 == 0: --- False

12 else:

13 return []

8. square\_equal(0, 36, 99) = [-2.75]

3 def square\_equal(a=0, b=36, c=99)

4 if 0 != 0: --- False

14 else:

15 if 36 != 0: --- True

16 return [-2.75]

9. square\_equal(0, 0, 89) = []

3 def square\_equal(a=0, b=0, c=89)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-25

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, 89) = 89

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

2 if 0 == 0: --- True

3 return 89

3. gcd(62, 0) = 62

1 def gcd(x=62, y=0)

2 if 62 == 0: --- False

4 if 0 == 0: --- True

5 return 62

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, 23, 22) = [-22.0, -1.0]

3 def square\_equal(a=1, b=23, c=22)

4 if 1 != 0: --- True

5 D = 23\*23 - 4\*1\*22

D = 441

6 if 441 > 0: --- True

7 x1 = (-23 - sqrt(441)) / (2\*1)

x1 = -22.0

8 x2 = (-23 + sqrt(441)) / (2\*1)

x2 = -1.0

9 return [-22.0, -1.0]

6. square\_equal(16, -72, 81) = [2.25]

3 def square\_equal(a=16, b=-72, c=81)

4 if 16 != 0: --- True

5 D = -72\*-72 - 4\*16\*81

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [2.25]

7. square\_equal(2, -7, 82) = []

3 def square\_equal(a=2, b=-7, c=82)

4 if 2 != 0: --- True

5 D = -7\*-7 - 4\*2\*82

D = -607

6 if -607 > 0: --- False

10 elif -607 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -1, -85) = [-85.0]

3 def square\_equal(a=0, b=-1, c=-85)

4 if 0 != 0: --- False

14 else:

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

16 return [-85.0]

9. square\_equal(0, 0, -41) = []

3 def square\_equal(a=0, b=0, c=-41)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-26

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, 97) = 97

1 def gcd(x=0, y=97)

2 if 0 == 0: --- True

3 return 97

3. gcd(12, 0) = 12

1 def gcd(x=12, y=0)

2 if 12 == 0: --- False

4 if 0 == 0: --- True

5 return 12

4. hex(225) = 'E1'

3 def hex(number=225)

4 if 225 == 0: --- False

6 res = ''

7 while 225 > 0: --- True

8 digit = 225 % 16

digit = 1

9 res = DIGITS[1] + ''

res = '1'

10 number = 225 // 16

number = 14

7 while 14 > 0: --- True

8 digit = 14 % 16

digit = 14

9 res = DIGITS[14] + '1'

res = 'E1'

10 number = 14 // 16

number = 0

7 while 0 > 0: --- False

11 return 'E1'

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

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 = 0.3

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

x2 = -0.2

9 return [0.3, -0.2]

6. square\_equal(-9, 36, -36) = [2.0]

3 def square\_equal(a=-9, b=36, c=-36)

4 if -9 != 0: --- True

5 D = 36\*36 - 4\*-9\*-36

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [2.0]

7. square\_equal(33, -1, 3) = []

3 def square\_equal(a=33, b=-1, c=3)

4 if 33 != 0: --- True

5 D = -1\*-1 - 4\*33\*3

D = -395

6 if -395 > 0: --- False

10 elif -395 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -42, -84) = [-2.0]

3 def square\_equal(a=0, b=-42, c=-84)

4 if 0 != 0: --- False

14 else:

15 if -42 != 0: --- True

16 return [-2.0]

9. square\_equal(0, 0, 50) = []

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

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-27

1. gcd(42, 66) = 6

1 def gcd(x=42, y=66)

2 if 42 == 0: --- False

4 if 66 == 0: --- False

6 while 66 != 0: --- True

7 rem = 42 % 66

rem = 42

8 x = 66

9 y = 42

6 while 42 != 0: --- True

7 rem = 66 % 42

rem = 24

8 x = 42

9 y = 24

6 while 24 != 0: --- True

7 rem = 42 % 24

rem = 18

8 x = 24

9 y = 18

6 while 18 != 0: --- True

7 rem = 24 % 18

rem = 6

8 x = 18

9 y = 6

6 while 6 != 0: --- True

7 rem = 18 % 6

rem = 0

8 x = 6

9 y = 0

6 while 0 != 0: --- False

10 return 6

2. gcd(0, 64) = 64

1 def gcd(x=0, y=64)

2 if 0 == 0: --- True

3 return 64

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(243) = 'F3'

3 def hex(number=243)

4 if 243 == 0: --- False

6 res = ''

7 while 243 > 0: --- True

8 digit = 243 % 16

digit = 3

9 res = DIGITS[3] + ''

res = '3'

10 number = 243 // 16

number = 15

7 while 15 > 0: --- True

8 digit = 15 % 16

digit = 15

9 res = DIGITS[15] + '3'

res = 'F3'

10 number = 15 // 16

number = 0

7 while 0 > 0: --- False

11 return 'F3'

5. square\_equal(100, -50, 4) = [0.1, 0.4]

3 def square\_equal(a=100, b=-50, c=4)

4 if 100 != 0: --- True

5 D = -50\*-50 - 4\*100\*4

D = 900

6 if 900 > 0: --- True

7 x1 = (--50 - sqrt(900)) / (2\*100)

x1 = 0.1

8 x2 = (--50 + sqrt(900)) / (2\*100)

x2 = 0.4

9 return [0.1, 0.4]

6. square\_equal(16, -8, 1) = [0.25]

3 def square\_equal(a=16, b=-8, c=1)

4 if 16 != 0: --- True

5 D = -8\*-8 - 4\*16\*1

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [0.25]

7. square\_equal(28, 19, 6) = []

3 def square\_equal(a=28, b=19, c=6)

4 if 28 != 0: --- True

5 D = 19\*19 - 4\*28\*6

D = -311

6 if -311 > 0: --- False

10 elif -311 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -4, -21) = [-5.25]

3 def square\_equal(a=0, b=-4, c=-21)

4 if 0 != 0: --- False

14 else:

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

16 return [-5.25]

9. square\_equal(0, 0, -40) = []

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

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-28

1. gcd(100, 56) = 4

1 def gcd(x=100, y=56)

2 if 100 == 0: --- False

4 if 56 == 0: --- False

6 while 56 != 0: --- True

7 rem = 100 % 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, 91) = 91

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

2 if 0 == 0: --- True

3 return 91

3. gcd(50, 0) = 50

1 def gcd(x=50, y=0)

2 if 50 == 0: --- False

4 if 0 == 0: --- True

5 return 50

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, 64, 17) = [4.25, -0.25]

3 def square\_equal(a=-16, b=64, c=17)

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

5 D = 64\*64 - 4\*-16\*17

D = 5184

6 if 5184 > 0: --- True

7 x1 = (-64 - sqrt(5184)) / (2\*-16)

x1 = 4.25

8 x2 = (-64 + sqrt(5184)) / (2\*-16)

x2 = -0.25

9 return [4.25, -0.25]

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(-25, -9, -7) = []

3 def square\_equal(a=-25, b=-9, c=-7)

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

5 D = -9\*-9 - 4\*-25\*-7

D = -619

6 if -619 > 0: --- False

10 elif -619 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -14, 63) = [4.5]

3 def square\_equal(a=0, b=-14, c=63)

4 if 0 != 0: --- False

14 else:

15 if -14 != 0: --- True

16 return [4.5]

9. square\_equal(0, 0, -18) = []

3 def square\_equal(a=0, b=0, c=-18)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-29

1. gcd(78, 48) = 6

1 def gcd(x=78, y=48)

2 if 78 == 0: --- False

4 if 48 == 0: --- False

6 while 48 != 0: --- True

7 rem = 78 % 48

rem = 30

8 x = 48

9 y = 30

6 while 30 != 0: --- True

7 rem = 48 % 30

rem = 18

8 x = 30

9 y = 18

6 while 18 != 0: --- True

7 rem = 30 % 18

rem = 12

8 x = 18

9 y = 12

6 while 12 != 0: --- True

7 rem = 18 % 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, 35) = 35

1 def gcd(x=0, y=35)

2 if 0 == 0: --- True

3 return 35

3. gcd(63, 0) = 63

1 def gcd(x=63, y=0)

2 if 63 == 0: --- False

4 if 0 == 0: --- True

5 return 63

4. hex(237) = 'ED'

3 def hex(number=237)

4 if 237 == 0: --- False

6 res = ''

7 while 237 > 0: --- True

8 digit = 237 % 16

digit = 13

9 res = DIGITS[13] + ''

res = 'D'

10 number = 237 // 16

number = 14

7 while 14 > 0: --- True

8 digit = 14 % 16

digit = 14

9 res = DIGITS[14] + 'D'

res = 'ED'

10 number = 14 // 16

number = 0

7 while 0 > 0: --- False

11 return 'ED'

5. square\_equal(-16, 0, 9) = [0.75, -0.75]

3 def square\_equal(a=-16, b=0, c=9)

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

5 D = 0\*0 - 4\*-16\*9

D = 576

6 if 576 > 0: --- True

7 x1 = (-0 - sqrt(576)) / (2\*-16)

x1 = 0.75

8 x2 = (-0 + sqrt(576)) / (2\*-16)

x2 = -0.75

9 return [0.75, -0.75]

6. square\_equal(-1, -16, -64) = [-8.0]

3 def square\_equal(a=-1, b=-16, c=-64)

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

5 D = -16\*-16 - 4\*-1\*-64

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-8.0]

7. square\_equal(-43, 19, -4) = []

3 def square\_equal(a=-43, b=19, c=-4)

4 if -43 != 0: --- True

5 D = 19\*19 - 4\*-43\*-4

D = -327

6 if -327 > 0: --- False

10 elif -327 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -40, -76) = [-1.9]

3 def square\_equal(a=0, b=-40, c=-76)

4 if 0 != 0: --- False

14 else:

15 if -40 != 0: --- True

16 return [-1.9]

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: 1-2-30

1. gcd(44, 72) = 4

1 def gcd(x=44, y=72)

2 if 44 == 0: --- False

4 if 72 == 0: --- False

6 while 72 != 0: --- True

7 rem = 44 % 72

rem = 44

8 x = 72

9 y = 44

6 while 44 != 0: --- True

7 rem = 72 % 44

rem = 28

8 x = 44

9 y = 28

6 while 28 != 0: --- True

7 rem = 44 % 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, 36) = 36

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

2 if 0 == 0: --- True

3 return 36

3. gcd(69, 0) = 69

1 def gcd(x=69, y=0)

2 if 69 == 0: --- False

4 if 0 == 0: --- True

5 return 69

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, -52, 48) = [1.0, 12.0]

3 def square\_equal(a=4, b=-52, c=48)

4 if 4 != 0: --- True

5 D = -52\*-52 - 4\*4\*48

D = 1936

6 if 1936 > 0: --- True

7 x1 = (--52 - sqrt(1936)) / (2\*4)

x1 = 1.0

8 x2 = (--52 + sqrt(1936)) / (2\*4)

x2 = 12.0

9 return [1.0, 12.0]

6. square\_equal(25, -70, 49) = [1.4]

3 def square\_equal(a=25, b=-70, c=49)

4 if 25 != 0: --- True

5 D = -70\*-70 - 4\*25\*49

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [1.4]

7. square\_equal(1, 10, 57) = []

3 def square\_equal(a=1, b=10, c=57)

4 if 1 != 0: --- True

5 D = 10\*10 - 4\*1\*57

D = -128

6 if -128 > 0: --- False

10 elif -128 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -10, 20) = [2.0]

3 def square\_equal(a=0, b=-10, c=20)

4 if 0 != 0: --- False

14 else:

15 if -10 != 0: --- True

16 return [2.0]

9. square\_equal(0, 0, 91) = []

3 def square\_equal(a=0, b=0, c=91)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-31

1. gcd(76, 48) = 4

1 def gcd(x=76, y=48)

2 if 76 == 0: --- False

4 if 48 == 0: --- False

6 while 48 != 0: --- True

7 rem = 76 % 48

rem = 28

8 x = 48

9 y = 28

6 while 28 != 0: --- True

7 rem = 48 % 28

rem = 20

8 x = 28

9 y = 20

6 while 20 != 0: --- True

7 rem = 28 % 20

rem = 8

8 x = 20

9 y = 8

6 while 8 != 0: --- True

7 rem = 20 % 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, 28) = 28

1 def gcd(x=0, y=28)

2 if 0 == 0: --- True

3 return 28

3. gcd(40, 0) = 40

1 def gcd(x=40, y=0)

2 if 40 == 0: --- False

4 if 0 == 0: --- True

5 return 40

4. hex(181) = 'B5'

3 def hex(number=181)

4 if 181 == 0: --- False

6 res = ''

7 while 181 > 0: --- True

8 digit = 181 % 16

digit = 5

9 res = DIGITS[5] + ''

res = '5'

10 number = 181 // 16

number = 11

7 while 11 > 0: --- True

8 digit = 11 % 16

digit = 11

9 res = DIGITS[11] + '5'

res = 'B5'

10 number = 11 // 16

number = 0

7 while 0 > 0: --- False

11 return 'B5'

5. square\_equal(-25, 5, 90) = [2.0, -1.8]

3 def square\_equal(a=-25, b=5, c=90)

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

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

D = 9025

6 if 9025 > 0: --- True

7 x1 = (-5 - sqrt(9025)) / (2\*-25)

x1 = 2.0

8 x2 = (-5 + sqrt(9025)) / (2\*-25)

x2 = -1.8

9 return [2.0, -1.8]

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(-8, 14, -14) = []

3 def square\_equal(a=-8, b=14, c=-14)

4 if -8 != 0: --- True

5 D = 14\*14 - 4\*-8\*-14

D = -252

6 if -252 > 0: --- False

10 elif -252 == 0: --- False

12 else:

13 return []

8. square\_equal(0, 20, 55) = [-2.75]

3 def square\_equal(a=0, b=20, c=55)

4 if 0 != 0: --- False

14 else:

15 if 20 != 0: --- True

16 return [-2.75]

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: 1-2-32

1. gcd(52, 80) = 4

1 def gcd(x=52, y=80)

2 if 52 == 0: --- False

4 if 80 == 0: --- False

6 while 80 != 0: --- True

7 rem = 52 % 80

rem = 52

8 x = 80

9 y = 52

6 while 52 != 0: --- True

7 rem = 80 % 52

rem = 28

8 x = 52

9 y = 28

6 while 28 != 0: --- True

7 rem = 52 % 28

rem = 24

8 x = 28

9 y = 24

6 while 24 != 0: --- True

7 rem = 28 % 24

rem = 4

8 x = 24

9 y = 4

6 while 4 != 0: --- True

7 rem = 24 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, 78) = 78

1 def gcd(x=0, y=78)

2 if 0 == 0: --- True

3 return 78

3. gcd(84, 0) = 84

1 def gcd(x=84, y=0)

2 if 84 == 0: --- False

4 if 0 == 0: --- True

5 return 84

4. hex(206) = 'CE'

3 def hex(number=206)

4 if 206 == 0: --- False

6 res = ''

7 while 206 > 0: --- True

8 digit = 206 % 16

digit = 14

9 res = DIGITS[14] + ''

res = 'E'

10 number = 206 // 16

number = 12

7 while 12 > 0: --- True

8 digit = 12 % 16

digit = 12

9 res = DIGITS[12] + 'E'

res = 'CE'

10 number = 12 // 16

number = 0

7 while 0 > 0: --- False

11 return 'CE'

5. square\_equal(1, 21, 98) = [-14.0, -7.0]

3 def square\_equal(a=1, b=21, c=98)

4 if 1 != 0: --- True

5 D = 21\*21 - 4\*1\*98

D = 49

6 if 49 > 0: --- True

7 x1 = (-21 - sqrt(49)) / (2\*1)

x1 = -14.0

8 x2 = (-21 + sqrt(49)) / (2\*1)

x2 = -7.0

9 return [-14.0, -7.0]

6. square\_equal(-9, -36, -36) = [-2.0]

3 def square\_equal(a=-9, b=-36, c=-36)

4 if -9 != 0: --- True

5 D = -36\*-36 - 4\*-9\*-36

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-2.0]

7. square\_equal(88, 13, 2) = []

3 def square\_equal(a=88, b=13, c=2)

4 if 88 != 0: --- True

5 D = 13\*13 - 4\*88\*2

D = -535

6 if -535 > 0: --- False

10 elif -535 == 0: --- False

12 else:

13 return []

8. square\_equal(0, 34, 68) = [-2.0]

3 def square\_equal(a=0, b=34, c=68)

4 if 0 != 0: --- False

14 else:

15 if 34 != 0: --- True

16 return [-2.0]

9. square\_equal(0, 0, -21) = []

3 def square\_equal(a=0, b=0, c=-21)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-33

1. gcd(40, 68) = 4

1 def gcd(x=40, y=68)

2 if 40 == 0: --- False

4 if 68 == 0: --- False

6 while 68 != 0: --- True

7 rem = 40 % 68

rem = 40

8 x = 68

9 y = 40

6 while 40 != 0: --- True

7 rem = 68 % 40

rem = 28

8 x = 40

9 y = 28

6 while 28 != 0: --- True

7 rem = 40 % 28

rem = 12

8 x = 28

9 y = 12

6 while 12 != 0: --- True

7 rem = 28 % 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, 58) = 58

1 def gcd(x=0, y=58)

2 if 0 == 0: --- True

3 return 58

3. gcd(74, 0) = 74

1 def gcd(x=74, y=0)

2 if 74 == 0: --- False

4 if 0 == 0: --- True

5 return 74

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, -20, 36) = [1.0, -2.25]

3 def square\_equal(a=-16, b=-20, c=36)

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

5 D = -20\*-20 - 4\*-16\*36

D = 2704

6 if 2704 > 0: --- True

7 x1 = (--20 - sqrt(2704)) / (2\*-16)

x1 = 1.0

8 x2 = (--20 + sqrt(2704)) / (2\*-16)

x2 = -2.25

9 return [1.0, -2.25]

6. square\_equal(-16, 64, -64) = [2.0]

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

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

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [2.0]

7. square\_equal(8, -13, 29) = []

3 def square\_equal(a=8, b=-13, c=29)

4 if 8 != 0: --- True

5 D = -13\*-13 - 4\*8\*29

D = -759

6 if -759 > 0: --- False

10 elif -759 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -15, -90) = [-6.0]

3 def square\_equal(a=0, b=-15, c=-90)

4 if 0 != 0: --- False

14 else:

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

16 return [-6.0]

9. square\_equal(0, 0, -60) = []

3 def square\_equal(a=0, b=0, c=-60)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-34

1. gcd(95, 60) = 5

1 def gcd(x=95, y=60)

2 if 95 == 0: --- False

4 if 60 == 0: --- False

6 while 60 != 0: --- True

7 rem = 95 % 60

rem = 35

8 x = 60

9 y = 35

6 while 35 != 0: --- True

7 rem = 60 % 35

rem = 25

8 x = 35

9 y = 25

6 while 25 != 0: --- True

7 rem = 35 % 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, 24) = 24

1 def gcd(x=0, y=24)

2 if 0 == 0: --- True

3 return 24

3. gcd(94, 0) = 94

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

2 if 94 == 0: --- False

4 if 0 == 0: --- True

5 return 94

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(1, 32, 31) = [-31.0, -1.0]

3 def square\_equal(a=1, b=32, c=31)

4 if 1 != 0: --- True

5 D = 32\*32 - 4\*1\*31

D = 900

6 if 900 > 0: --- True

7 x1 = (-32 - sqrt(900)) / (2\*1)

x1 = -31.0

8 x2 = (-32 + sqrt(900)) / (2\*1)

x2 = -1.0

9 return [-31.0, -1.0]

6. square\_equal(-9, 18, -9) = [1.0]

3 def square\_equal(a=-9, b=18, c=-9)

4 if -9 != 0: --- True

5 D = 18\*18 - 4\*-9\*-9

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [1.0]

7. square\_equal(6, 19, 16) = []

3 def square\_equal(a=6, b=19, c=16)

4 if 6 != 0: --- True

5 D = 19\*19 - 4\*6\*16

D = -23

6 if -23 > 0: --- False

10 elif -23 == 0: --- False

12 else:

13 return []

8. square\_equal(0, 25, 6) = [-0.24]

3 def square\_equal(a=0, b=25, c=6)

4 if 0 != 0: --- False

14 else:

15 if 25 != 0: --- True

16 return [-0.24]

9. square\_equal(0, 0, 45) = []

3 def square\_equal(a=0, b=0, c=45)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-35

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, 9) = 9

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

2 if 0 == 0: --- True

3 return 9

3. gcd(46, 0) = 46

1 def gcd(x=46, y=0)

2 if 46 == 0: --- False

4 if 0 == 0: --- True

5 return 46

4. hex(250) = 'FA'

3 def hex(number=250)

4 if 250 == 0: --- False

6 res = ''

7 while 250 > 0: --- True

8 digit = 250 % 16

digit = 10

9 res = DIGITS[10] + ''

res = 'A'

10 number = 250 // 16

number = 15

7 while 15 > 0: --- True

8 digit = 15 % 16

digit = 15

9 res = DIGITS[15] + 'A'

res = 'FA'

10 number = 15 // 16

number = 0

7 while 0 > 0: --- False

11 return 'FA'

5. square\_equal(1, 61, 60) = [-60.0, -1.0]

3 def square\_equal(a=1, b=61, c=60)

4 if 1 != 0: --- True

5 D = 61\*61 - 4\*1\*60

D = 3481

6 if 3481 > 0: --- True

7 x1 = (-61 - sqrt(3481)) / (2\*1)

x1 = -60.0

8 x2 = (-61 + sqrt(3481)) / (2\*1)

x2 = -1.0

9 return [-60.0, -1.0]

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(-8, 16, -20) = []

3 def square\_equal(a=-8, b=16, c=-20)

4 if -8 != 0: --- True

5 D = 16\*16 - 4\*-8\*-20

D = -384

6 if -384 > 0: --- False

10 elif -384 == 0: --- False

12 else:

13 return []

8. square\_equal(0, 75, -45) = [0.6]

3 def square\_equal(a=0, b=75, c=-45)

4 if 0 != 0: --- False

14 else:

15 if 75 != 0: --- True

16 return [0.6]

9. square\_equal(0, 0, 7) = []

3 def square\_equal(a=0, b=0, c=7)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-36

1. gcd(72, 99) = 9

1 def gcd(x=72, y=99)

2 if 72 == 0: --- False

4 if 99 == 0: --- False

6 while 99 != 0: --- True

7 rem = 72 % 99

rem = 72

8 x = 99

9 y = 72

6 while 72 != 0: --- True

7 rem = 99 % 72

rem = 27

8 x = 72

9 y = 27

6 while 27 != 0: --- True

7 rem = 72 % 27

rem = 18

8 x = 27

9 y = 18

6 while 18 != 0: --- True

7 rem = 27 % 18

rem = 9

8 x = 18

9 y = 9

6 while 9 != 0: --- True

7 rem = 18 % 9

rem = 0

8 x = 9

9 y = 0

6 while 0 != 0: --- False

10 return 9

2. gcd(0, 60) = 60

1 def gcd(x=0, y=60)

2 if 0 == 0: --- True

3 return 60

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(247) = 'F7'

3 def hex(number=247)

4 if 247 == 0: --- False

6 res = ''

7 while 247 > 0: --- True

8 digit = 247 % 16

digit = 7

9 res = DIGITS[7] + ''

res = '7'

10 number = 247 // 16

number = 15

7 while 15 > 0: --- True

8 digit = 15 % 16

digit = 15

9 res = DIGITS[15] + '7'

res = 'F7'

10 number = 15 // 16

number = 0

7 while 0 > 0: --- False

11 return 'F7'

5. square\_equal(25, -70, 24) = [0.4, 2.4]

3 def square\_equal(a=25, b=-70, c=24)

4 if 25 != 0: --- True

5 D = -70\*-70 - 4\*25\*24

D = 2500

6 if 2500 > 0: --- True

7 x1 = (--70 - sqrt(2500)) / (2\*25)

x1 = 0.4

8 x2 = (--70 + sqrt(2500)) / (2\*25)

x2 = 2.4

9 return [0.4, 2.4]

6. square\_equal(1, -18, 81) = [9.0]

3 def square\_equal(a=1, b=-18, c=81)

4 if 1 != 0: --- True

5 D = -18\*-18 - 4\*1\*81

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [9.0]

7. square\_equal(-9, -16, -11) = []

3 def square\_equal(a=-9, b=-16, c=-11)

4 if -9 != 0: --- True

5 D = -16\*-16 - 4\*-9\*-11

D = -140

6 if -140 > 0: --- False

10 elif -140 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -10, -16) = [-1.6]

3 def square\_equal(a=0, b=-10, c=-16)

4 if 0 != 0: --- False

14 else:

15 if -10 != 0: --- True

16 return [-1.6]

9. square\_equal(0, 0, -23) = []

3 def square\_equal(a=0, b=0, c=-23)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-37

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, 7) = 7

1 def gcd(x=0, y=7)

2 if 0 == 0: --- True

3 return 7

3. gcd(19, 0) = 19

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

2 if 19 == 0: --- False

4 if 0 == 0: --- True

5 return 19

4. hex(213) = 'D5'

3 def hex(number=213)

4 if 213 == 0: --- False

6 res = ''

7 while 213 > 0: --- True

8 digit = 213 % 16

digit = 5

9 res = DIGITS[5] + ''

res = '5'

10 number = 213 // 16

number = 13

7 while 13 > 0: --- True

8 digit = 13 % 16

digit = 13

9 res = DIGITS[13] + '5'

res = 'D5'

10 number = 13 // 16

number = 0

7 while 0 > 0: --- False

11 return 'D5'

5. square\_equal(16, 56, 49) = [-1.75]

3 def square\_equal(a=16, b=56, c=49)

4 if 16 != 0: --- True

5 D = 56\*56 - 4\*16\*49

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-1.75]

6. square\_equal(-16, 72, -81) = [2.25]

3 def square\_equal(a=-16, b=72, c=-81)

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

5 D = 72\*72 - 4\*-16\*-81

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [2.25]

7. square\_equal(-45, 28, -5) = []

3 def square\_equal(a=-45, b=28, c=-5)

4 if -45 != 0: --- True

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

D = -116

6 if -116 > 0: --- False

10 elif -116 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -1, -23) = [-23.0]

3 def square\_equal(a=0, b=-1, c=-23)

4 if 0 != 0: --- False

14 else:

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

16 return [-23.0]

9. square\_equal(0, 0, -30) = []

3 def square\_equal(a=0, b=0, c=-30)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-38

1. gcd(56, 96) = 8

1 def gcd(x=56, y=96)

2 if 56 == 0: --- False

4 if 96 == 0: --- False

6 while 96 != 0: --- True

7 rem = 56 % 96

rem = 56

8 x = 96

9 y = 56

6 while 56 != 0: --- True

7 rem = 96 % 56

rem = 40

8 x = 56

9 y = 40

6 while 40 != 0: --- True

7 rem = 56 % 40

rem = 16

8 x = 40

9 y = 16

6 while 16 != 0: --- True

7 rem = 40 % 16

rem = 8

8 x = 16

9 y = 8

6 while 8 != 0: --- True

7 rem = 16 % 8

rem = 0

8 x = 8

9 y = 0

6 while 0 != 0: --- False

10 return 8

2. gcd(0, 21) = 21

1 def gcd(x=0, y=21)

2 if 0 == 0: --- True

3 return 21

3. gcd(35, 0) = 35

1 def gcd(x=35, y=0)

2 if 35 == 0: --- False

4 if 0 == 0: --- True

5 return 35

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(-4, -8, 5) = [0.5, -2.5]

3 def square\_equal(a=-4, b=-8, c=5)

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

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

D = 144

6 if 144 > 0: --- True

7 x1 = (--8 - sqrt(144)) / (2\*-4)

x1 = 0.5

8 x2 = (--8 + sqrt(144)) / (2\*-4)

x2 = -2.5

9 return [0.5, -2.5]

6. square\_equal(25, -60, 36) = [1.2]

3 def square\_equal(a=25, b=-60, c=36)

4 if 25 != 0: --- True

5 D = -60\*-60 - 4\*25\*36

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [1.2]

7. square\_equal(77, 1, 3) = []

3 def square\_equal(a=77, b=1, c=3)

4 if 77 != 0: --- True

5 D = 1\*1 - 4\*77\*3

D = -923

6 if -923 > 0: --- False

10 elif -923 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -4, 22) = [5.5]

3 def square\_equal(a=0, b=-4, c=22)

4 if 0 != 0: --- False

14 else:

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

16 return [5.5]

9. square\_equal(0, 0, 57) = []

3 def square\_equal(a=0, b=0, c=57)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-39

1. gcd(28, 76) = 4

1 def gcd(x=28, y=76)

2 if 28 == 0: --- False

4 if 76 == 0: --- False

6 while 76 != 0: --- True

7 rem = 28 % 76

rem = 28

8 x = 76

9 y = 28

6 while 28 != 0: --- True

7 rem = 76 % 28

rem = 20

8 x = 28

9 y = 20

6 while 20 != 0: --- True

7 rem = 28 % 20

rem = 8

8 x = 20

9 y = 8

6 while 8 != 0: --- True

7 rem = 20 % 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, 33) = 33

1 def gcd(x=0, y=33)

2 if 0 == 0: --- True

3 return 33

3. gcd(78, 0) = 78

1 def gcd(x=78, y=0)

2 if 78 == 0: --- False

4 if 0 == 0: --- True

5 return 78

4. hex(238) = 'EE'

3 def hex(number=238)

4 if 238 == 0: --- False

6 res = ''

7 while 238 > 0: --- True

8 digit = 238 % 16

digit = 14

9 res = DIGITS[14] + ''

res = 'E'

10 number = 238 // 16

number = 14

7 while 14 > 0: --- True

8 digit = 14 % 16

digit = 14

9 res = DIGITS[14] + 'E'

res = 'EE'

10 number = 14 // 16

number = 0

7 while 0 > 0: --- False

11 return 'EE'

5. square\_equal(-25, 5, 56) = [1.6, -1.4]

3 def square\_equal(a=-25, b=5, c=56)

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

5 D = 5\*5 - 4\*-25\*56

D = 5625

6 if 5625 > 0: --- True

7 x1 = (-5 - sqrt(5625)) / (2\*-25)

x1 = 1.6

8 x2 = (-5 + sqrt(5625)) / (2\*-25)

x2 = -1.4

9 return [1.6, -1.4]

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(8, -12, 29) = []

3 def square\_equal(a=8, b=-12, c=29)

4 if 8 != 0: --- True

5 D = -12\*-12 - 4\*8\*29

D = -784

6 if -784 > 0: --- False

10 elif -784 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -20, -2) = [-0.1]

3 def square\_equal(a=0, b=-20, c=-2)

4 if 0 != 0: --- False

14 else:

15 if -20 != 0: --- True

16 return [-0.1]

9. square\_equal(0, 0, 80) = []

3 def square\_equal(a=0, b=0, c=80)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-40

1. gcd(56, 77) = 7

1 def gcd(x=56, y=77)

2 if 56 == 0: --- False

4 if 77 == 0: --- False

6 while 77 != 0: --- True

7 rem = 56 % 77

rem = 56

8 x = 77

9 y = 56

6 while 56 != 0: --- True

7 rem = 77 % 56

rem = 21

8 x = 56

9 y = 21

6 while 21 != 0: --- True

7 rem = 56 % 21

rem = 14

8 x = 21

9 y = 14

6 while 14 != 0: --- True

7 rem = 21 % 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, 72) = 72

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

2 if 0 == 0: --- True

3 return 72

3. gcd(11, 0) = 11

1 def gcd(x=11, y=0)

2 if 11 == 0: --- False

4 if 0 == 0: --- True

5 return 11

4. hex(252) = 'FC'

3 def hex(number=252)

4 if 252 == 0: --- False

6 res = ''

7 while 252 > 0: --- True

8 digit = 252 % 16

digit = 12

9 res = DIGITS[12] + ''

res = 'C'

10 number = 252 // 16

number = 15

7 while 15 > 0: --- True

8 digit = 15 % 16

digit = 15

9 res = DIGITS[15] + 'C'

res = 'FC'

10 number = 15 // 16

number = 0

7 while 0 > 0: --- False

11 return 'FC'

5. square\_equal(1, 17, 42) = [-14.0, -3.0]

3 def square\_equal(a=1, b=17, c=42)

4 if 1 != 0: --- True

5 D = 17\*17 - 4\*1\*42

D = 121

6 if 121 > 0: --- True

7 x1 = (-17 - sqrt(121)) / (2\*1)

x1 = -14.0

8 x2 = (-17 + sqrt(121)) / (2\*1)

x2 = -3.0

9 return [-14.0, -3.0]

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(-6, 3, -39) = []

3 def square\_equal(a=-6, b=3, c=-39)

4 if -6 != 0: --- True

5 D = 3\*3 - 4\*-6\*-39

D = -927

6 if -927 > 0: --- False

10 elif -927 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -32, 72) = [2.25]

3 def square\_equal(a=0, b=-32, c=72)

4 if 0 != 0: --- False

14 else:

15 if -32 != 0: --- True

16 return [2.25]

9. square\_equal(0, 0, 35) = []

3 def square\_equal(a=0, b=0, c=35)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-41

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, 63) = 63

1 def gcd(x=0, y=63)

2 if 0 == 0: --- True

3 return 63

3. gcd(83, 0) = 83

1 def gcd(x=83, y=0)

2 if 83 == 0: --- False

4 if 0 == 0: --- True

5 return 83

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(25, 40, 7) = [-1.4, -0.2]

3 def square\_equal(a=25, b=40, c=7)

4 if 25 != 0: --- True

5 D = 40\*40 - 4\*25\*7

D = 900

6 if 900 > 0: --- True

7 x1 = (-40 - sqrt(900)) / (2\*25)

x1 = -1.4

8 x2 = (-40 + sqrt(900)) / (2\*25)

x2 = -0.2

9 return [-1.4, -0.2]

6. square\_equal(-4, -16, -16) = [-2.0]

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

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

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-2.0]

7. square\_equal(-2, 2, -37) = []

3 def square\_equal(a=-2, b=2, c=-37)

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

5 D = 2\*2 - 4\*-2\*-37

D = -292

6 if -292 > 0: --- False

10 elif -292 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -12, -45) = [-3.75]

3 def square\_equal(a=0, b=-12, c=-45)

4 if 0 != 0: --- False

14 else:

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

16 return [-3.75]

9. square\_equal(0, 0, -14) = []

3 def square\_equal(a=0, b=0, c=-14)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-42

1. gcd(52, 32) = 4

1 def gcd(x=52, y=32)

2 if 52 == 0: --- False

4 if 32 == 0: --- False

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, 50) = 50

1 def gcd(x=0, y=50)

2 if 0 == 0: --- True

3 return 50

3. gcd(53, 0) = 53

1 def gcd(x=53, y=0)

2 if 53 == 0: --- False

4 if 0 == 0: --- True

5 return 53

4. hex(223) = 'DF'

3 def hex(number=223)

4 if 223 == 0: --- False

6 res = ''

7 while 223 > 0: --- True

8 digit = 223 % 16

digit = 15

9 res = DIGITS[15] + ''

res = 'F'

10 number = 223 // 16

number = 13

7 while 13 > 0: --- True

8 digit = 13 % 16

digit = 13

9 res = DIGITS[13] + 'F'

res = 'DF'

10 number = 13 // 16

number = 0

7 while 0 > 0: --- False

11 return 'DF'

5. square\_equal(-1, 69, 70) = [70.0, -1.0]

3 def square\_equal(a=-1, b=69, c=70)

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

5 D = 69\*69 - 4\*-1\*70

D = 5041

6 if 5041 > 0: --- True

7 x1 = (-69 - sqrt(5041)) / (2\*-1)

x1 = 70.0

8 x2 = (-69 + sqrt(5041)) / (2\*-1)

x2 = -1.0

9 return [70.0, -1.0]

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(1, -2, 40) = []

3 def square\_equal(a=1, b=-2, c=40)

4 if 1 != 0: --- True

5 D = -2\*-2 - 4\*1\*40

D = -156

6 if -156 > 0: --- False

10 elif -156 == 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, 83) = []

3 def square\_equal(a=0, b=0, c=83)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-43

1. gcd(35, 90) = 5

1 def gcd(x=35, y=90)

2 if 35 == 0: --- False

4 if 90 == 0: --- False

6 while 90 != 0: --- True

7 rem = 35 % 90

rem = 35

8 x = 90

9 y = 35

6 while 35 != 0: --- True

7 rem = 90 % 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, 55) = 55

1 def gcd(x=0, y=55)

2 if 0 == 0: --- True

3 return 55

3. gcd(42, 0) = 42

1 def gcd(x=42, y=0)

2 if 42 == 0: --- False

4 if 0 == 0: --- True

5 return 42

4. hex(185) = 'B9'

3 def hex(number=185)

4 if 185 == 0: --- False

6 res = ''

7 while 185 > 0: --- True

8 digit = 185 % 16

digit = 9

9 res = DIGITS[9] + ''

res = '9'

10 number = 185 // 16

number = 11

7 while 11 > 0: --- True

8 digit = 11 % 16

digit = 11

9 res = DIGITS[11] + '9'

res = 'B9'

10 number = 11 // 16

number = 0

7 while 0 > 0: --- False

11 return 'B9'

5. square\_equal(-25, 15, 70) = [2.0, -1.4]

3 def square\_equal(a=-25, b=15, c=70)

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

5 D = 15\*15 - 4\*-25\*70

D = 7225

6 if 7225 > 0: --- True

7 x1 = (-15 - sqrt(7225)) / (2\*-25)

x1 = 2.0

8 x2 = (-15 + sqrt(7225)) / (2\*-25)

x2 = -1.4

9 return [2.0, -1.4]

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(3, 0, 29) = []

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

4 if 3 != 0: --- True

5 D = 0\*0 - 4\*3\*29

D = -348

6 if -348 > 0: --- False

10 elif -348 == 0: --- False

12 else:

13 return []

8. square\_equal(0, 3, -93) = [31.0]

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

4 if 0 != 0: --- False

14 else:

15 if 3 != 0: --- True

16 return [31.0]

9. square\_equal(0, 0, -89) = []

3 def square\_equal(a=0, b=0, c=-89)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-44

1. gcd(64, 92) = 4

1 def gcd(x=64, y=92)

2 if 64 == 0: --- False

4 if 92 == 0: --- False

6 while 92 != 0: --- True

7 rem = 64 % 92

rem = 64

8 x = 92

9 y = 64

6 while 64 != 0: --- True

7 rem = 92 % 64

rem = 28

8 x = 64

9 y = 28

6 while 28 != 0: --- True

7 rem = 64 % 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, 68) = 68

1 def gcd(x=0, y=68)

2 if 0 == 0: --- True

3 return 68

3. gcd(95, 0) = 95

1 def gcd(x=95, y=0)

2 if 95 == 0: --- False

4 if 0 == 0: --- True

5 return 95

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(1, -23, 42) = [2.0, 21.0]

3 def square\_equal(a=1, b=-23, c=42)

4 if 1 != 0: --- True

5 D = -23\*-23 - 4\*1\*42

D = 361

6 if 361 > 0: --- True

7 x1 = (--23 - sqrt(361)) / (2\*1)

x1 = 2.0

8 x2 = (--23 + sqrt(361)) / (2\*1)

x2 = 21.0

9 return [2.0, 21.0]

6. square\_equal(1, 6, 9) = [-3.0]

3 def square\_equal(a=1, b=6, c=9)

4 if 1 != 0: --- True

5 D = 6\*6 - 4\*1\*9

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-3.0]

7. square\_equal(84, 20, 2) = []

3 def square\_equal(a=84, b=20, c=2)

4 if 84 != 0: --- True

5 D = 20\*20 - 4\*84\*2

D = -272

6 if -272 > 0: --- False

10 elif -272 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -30, -24) = [-0.8]

3 def square\_equal(a=0, b=-30, c=-24)

4 if 0 != 0: --- False

14 else:

15 if -30 != 0: --- True

16 return [-0.8]

9. square\_equal(0, 0, -74) = []

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

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-45

1. gcd(44, 68) = 4

1 def gcd(x=44, y=68)

2 if 44 == 0: --- False

4 if 68 == 0: --- False

6 while 68 != 0: --- True

7 rem = 44 % 68

rem = 44

8 x = 68

9 y = 44

6 while 44 != 0: --- True

7 rem = 68 % 44

rem = 24

8 x = 44

9 y = 24

6 while 24 != 0: --- True

7 rem = 44 % 24

rem = 20

8 x = 24

9 y = 20

6 while 20 != 0: --- True

7 rem = 24 % 20

rem = 4

8 x = 20

9 y = 4

6 while 4 != 0: --- True

7 rem = 20 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, 8) = 8

1 def gcd(x=0, y=8)

2 if 0 == 0: --- True

3 return 8

3. gcd(54, 0) = 54

1 def gcd(x=54, y=0)

2 if 54 == 0: --- False

4 if 0 == 0: --- True

5 return 54

4. hex(222) = 'DE'

3 def hex(number=222)

4 if 222 == 0: --- False

6 res = ''

7 while 222 > 0: --- True

8 digit = 222 % 16

digit = 14

9 res = DIGITS[14] + ''

res = 'E'

10 number = 222 // 16

number = 13

7 while 13 > 0: --- True

8 digit = 13 % 16

digit = 13

9 res = DIGITS[13] + 'E'

res = 'DE'

10 number = 13 // 16

number = 0

7 while 0 > 0: --- False

11 return 'DE'

5. square\_equal(1, 35, 34) = [-34.0, -1.0]

3 def square\_equal(a=1, b=35, c=34)

4 if 1 != 0: --- True

5 D = 35\*35 - 4\*1\*34

D = 1089

6 if 1089 > 0: --- True

7 x1 = (-35 - sqrt(1089)) / (2\*1)

x1 = -34.0

8 x2 = (-35 + sqrt(1089)) / (2\*1)

x2 = -1.0

9 return [-34.0, -1.0]

6. square\_equal(4, -4, 1) = [0.5]

3 def square\_equal(a=4, b=-4, c=1)

4 if 4 != 0: --- True

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [0.5]

7. square\_equal(-2, -4, -27) = []

3 def square\_equal(a=-2, b=-4, c=-27)

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

5 D = -4\*-4 - 4\*-2\*-27

D = -200

6 if -200 > 0: --- False

10 elif -200 == 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, 77) = []

3 def square\_equal(a=0, b=0, c=77)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-46

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, 54) = 54

1 def gcd(x=0, y=54)

2 if 0 == 0: --- True

3 return 54

3. gcd(34, 0) = 34

1 def gcd(x=34, y=0)

2 if 34 == 0: --- False

4 if 0 == 0: --- True

5 return 34

4. hex(226) = 'E2'

3 def hex(number=226)

4 if 226 == 0: --- False

6 res = ''

7 while 226 > 0: --- True

8 digit = 226 % 16

digit = 2

9 res = DIGITS[2] + ''

res = '2'

10 number = 226 // 16

number = 14

7 while 14 > 0: --- True

8 digit = 14 % 16

digit = 14

9 res = DIGITS[14] + '2'

res = 'E2'

10 number = 14 // 16

number = 0

7 while 0 > 0: --- False

11 return 'E2'

5. square\_equal(-1, -40, 84) = [2.0, -42.0]

3 def square\_equal(a=-1, b=-40, c=84)

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

5 D = -40\*-40 - 4\*-1\*84

D = 1936

6 if 1936 > 0: --- True

7 x1 = (--40 - sqrt(1936)) / (2\*-1)

x1 = 2.0

8 x2 = (--40 + sqrt(1936)) / (2\*-1)

x2 = -42.0

9 return [2.0, -42.0]

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(-1, -1, -22) = []

3 def square\_equal(a=-1, b=-1, c=-22)

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

5 D = -1\*-1 - 4\*-1\*-22

D = -87

6 if -87 > 0: --- False

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

12 else:

13 return []

8. square\_equal(0, -25, -92) = [-3.68]

3 def square\_equal(a=0, b=-25, c=-92)

4 if 0 != 0: --- False

14 else:

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

16 return [-3.68]

9. square\_equal(0, 0, 68) = []

3 def square\_equal(a=0, b=0, c=68)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-47

1. gcd(20, 52) = 4

1 def gcd(x=20, y=52)

2 if 20 == 0: --- False

4 if 52 == 0: --- False

6 while 52 != 0: --- True

7 rem = 20 % 52

rem = 20

8 x = 52

9 y = 20

6 while 20 != 0: --- True

7 rem = 52 % 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, 100) = 100

1 def gcd(x=0, y=100)

2 if 0 == 0: --- True

3 return 100

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(240) = 'F0'

3 def hex(number=240)

4 if 240 == 0: --- False

6 res = ''

7 while 240 > 0: --- True

8 digit = 240 % 16

digit = 0

9 res = DIGITS[0] + ''

res = '0'

10 number = 240 // 16

number = 15

7 while 15 > 0: --- True

8 digit = 15 % 16

digit = 15

9 res = DIGITS[15] + '0'

res = 'F0'

10 number = 15 // 16

number = 0

7 while 0 > 0: --- False

11 return 'F0'

5. square\_equal(1, 92, 91) = [-91.0, -1.0]

3 def square\_equal(a=1, b=92, c=91)

4 if 1 != 0: --- True

5 D = 92\*92 - 4\*1\*91

D = 8100

6 if 8100 > 0: --- True

7 x1 = (-92 - sqrt(8100)) / (2\*1)

x1 = -91.0

8 x2 = (-92 + sqrt(8100)) / (2\*1)

x2 = -1.0

9 return [-91.0, -1.0]

6. square\_equal(-9, -18, -9) = [-1.0]

3 def square\_equal(a=-9, b=-18, c=-9)

4 if -9 != 0: --- True

5 D = -18\*-18 - 4\*-9\*-9

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-1.0]

7. square\_equal(5, -6, 40) = []

3 def square\_equal(a=5, b=-6, c=40)

4 if 5 != 0: --- True

5 D = -6\*-6 - 4\*5\*40

D = -764

6 if -764 > 0: --- False

10 elif -764 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -71, 71) = [1.0]

3 def square\_equal(a=0, b=-71, c=71)

4 if 0 != 0: --- False

14 else:

15 if -71 != 0: --- True

16 return [1.0]

9. square\_equal(0, 0, -1) = []

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

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-48

1. gcd(56, 88) = 8

1 def gcd(x=56, y=88)

2 if 56 == 0: --- False

4 if 88 == 0: --- False

6 while 88 != 0: --- True

7 rem = 56 % 88

rem = 56

8 x = 88

9 y = 56

6 while 56 != 0: --- True

7 rem = 88 % 56

rem = 32

8 x = 56

9 y = 32

6 while 32 != 0: --- True

7 rem = 56 % 32

rem = 24

8 x = 32

9 y = 24

6 while 24 != 0: --- True

7 rem = 32 % 24

rem = 8

8 x = 24

9 y = 8

6 while 8 != 0: --- True

7 rem = 24 % 8

rem = 0

8 x = 8

9 y = 0

6 while 0 != 0: --- False

10 return 8

2. gcd(0, 49) = 49

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

2 if 0 == 0: --- True

3 return 49

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(183) = 'B7'

3 def hex(number=183)

4 if 183 == 0: --- False

6 res = ''

7 while 183 > 0: --- True

8 digit = 183 % 16

digit = 7

9 res = DIGITS[7] + ''

res = '7'

10 number = 183 // 16

number = 11

7 while 11 > 0: --- True

8 digit = 11 % 16

digit = 11

9 res = DIGITS[11] + '7'

res = 'B7'

10 number = 11 // 16

number = 0

7 while 0 > 0: --- False

11 return 'B7'

5. square\_equal(-25, 55, 42) = [2.8, -0.6]

3 def square\_equal(a=-25, b=55, c=42)

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

5 D = 55\*55 - 4\*-25\*42

D = 7225

6 if 7225 > 0: --- True

7 x1 = (-55 - sqrt(7225)) / (2\*-25)

x1 = 2.8

8 x2 = (-55 + sqrt(7225)) / (2\*-25)

x2 = -0.6

9 return [2.8, -0.6]

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(53, -18, 3) = []

3 def square\_equal(a=53, b=-18, c=3)

4 if 53 != 0: --- True

5 D = -18\*-18 - 4\*53\*3

D = -312

6 if -312 > 0: --- False

10 elif -312 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -2, 18) = [9.0]

3 def square\_equal(a=0, b=-2, c=18)

4 if 0 != 0: --- False

14 else:

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

16 return [9.0]

9. square\_equal(0, 0, -36) = []

3 def square\_equal(a=0, b=0, c=-36)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-49

1. gcd(30, 48) = 6

1 def gcd(x=30, y=48)

2 if 30 == 0: --- False

4 if 48 == 0: --- False

6 while 48 != 0: --- True

7 rem = 30 % 48

rem = 30

8 x = 48

9 y = 30

6 while 30 != 0: --- True

7 rem = 48 % 30

rem = 18

8 x = 30

9 y = 18

6 while 18 != 0: --- True

7 rem = 30 % 18

rem = 12

8 x = 18

9 y = 12

6 while 12 != 0: --- True

7 rem = 18 % 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, 2) = 2

1 def gcd(x=0, y=2)

2 if 0 == 0: --- True

3 return 2

3. gcd(26, 0) = 26

1 def gcd(x=26, y=0)

2 if 26 == 0: --- False

4 if 0 == 0: --- True

5 return 26

4. hex(249) = 'F9'

3 def hex(number=249)

4 if 249 == 0: --- False

6 res = ''

7 while 249 > 0: --- True

8 digit = 249 % 16

digit = 9

9 res = DIGITS[9] + ''

res = '9'

10 number = 249 // 16

number = 15

7 while 15 > 0: --- True

8 digit = 15 % 16

digit = 15

9 res = DIGITS[15] + '9'

res = 'F9'

10 number = 15 // 16

number = 0

7 while 0 > 0: --- False

11 return 'F9'

5. square\_equal(-16, 68, 60) = [5.0, -0.75]

3 def square\_equal(a=-16, b=68, c=60)

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

5 D = 68\*68 - 4\*-16\*60

D = 8464

6 if 8464 > 0: --- True

7 x1 = (-68 - sqrt(8464)) / (2\*-16)

x1 = 5.0

8 x2 = (-68 + sqrt(8464)) / (2\*-16)

x2 = -0.75

9 return [5.0, -0.75]

6. square\_equal(36, -72, 36) = [1.0]

3 def square\_equal(a=36, b=-72, c=36)

4 if 36 != 0: --- True

5 D = -72\*-72 - 4\*36\*36

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [1.0]

7. square\_equal(7, -17, 30) = []

3 def square\_equal(a=7, b=-17, c=30)

4 if 7 != 0: --- True

5 D = -17\*-17 - 4\*7\*30

D = -551

6 if -551 > 0: --- False

10 elif -551 == 0: --- False

12 else:

13 return []

8. square\_equal(0, 5, 90) = [-18.0]

3 def square\_equal(a=0, b=5, c=90)

4 if 0 != 0: --- False

14 else:

15 if 5 != 0: --- True

16 return [-18.0]

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

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

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

Вариант: 1-2-50

1. gcd(80, 92) = 4

1 def gcd(x=80, y=92)

2 if 80 == 0: --- False

4 if 92 == 0: --- False

6 while 92 != 0: --- True

7 rem = 80 % 92

rem = 80

8 x = 92

9 y = 80

6 while 80 != 0: --- True

7 rem = 92 % 80

rem = 12

8 x = 80

9 y = 12

6 while 12 != 0: --- True

7 rem = 80 % 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, 62) = 62

1 def gcd(x=0, y=62)

2 if 0 == 0: --- True

3 return 62

3. gcd(56, 0) = 56

1 def gcd(x=56, y=0)

2 if 56 == 0: --- False

4 if 0 == 0: --- True

5 return 56

4. hex(212) = 'D4'

3 def hex(number=212)

4 if 212 == 0: --- False

6 res = ''

7 while 212 > 0: --- True

8 digit = 212 % 16

digit = 4

9 res = DIGITS[4] + ''

res = '4'

10 number = 212 // 16

number = 13

7 while 13 > 0: --- True

8 digit = 13 % 16

digit = 13

9 res = DIGITS[13] + '4'

res = 'D4'

10 number = 13 // 16

number = 0

7 while 0 > 0: --- False

11 return 'D4'

5. square\_equal(-4, -12, 7) = [0.5, -3.5]

3 def square\_equal(a=-4, b=-12, c=7)

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

5 D = -12\*-12 - 4\*-4\*7

D = 256

6 if 256 > 0: --- True

7 x1 = (--12 - sqrt(256)) / (2\*-4)

x1 = 0.5

8 x2 = (--12 + sqrt(256)) / (2\*-4)

x2 = -3.5

9 return [0.5, -3.5]

6. square\_equal(16, 56, 49) = [-1.75]

3 def square\_equal(a=16, b=56, c=49)

4 if 16 != 0: --- True

5 D = 56\*56 - 4\*16\*49

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-1.75]

7. square\_equal(7, -14, 30) = []

3 def square\_equal(a=7, b=-14, c=30)

4 if 7 != 0: --- True

5 D = -14\*-14 - 4\*7\*30

D = -644

6 if -644 > 0: --- False

10 elif -644 == 0: --- False

12 else:

13 return []

8. square\_equal(0, -5, -81) = [-16.2]

3 def square\_equal(a=0, b=-5, c=-81)

4 if 0 != 0: --- False

14 else:

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

16 return [-16.2]

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 []