Вариант: 1-1-1

1. gcd(52, 88) = 4

1 in function gcd(x=52, y=88)

2 if 52 == 0: --- False

4 if 88 == 0: --- False

6 while 88 != 0: --- True

7 rem = x % y

rem = 52 % 88

rem = 52

8 x = y

x = 88

9 y = rem

y = 52

6 while 52 != 0: --- True

7 rem = x % y

rem = 88 % 52

rem = 36

8 x = y

x = 52

9 y = rem

y = 36

6 while 36 != 0: --- True

7 rem = x % y

rem = 52 % 36

rem = 16

8 x = y

x = 36

9 y = rem

y = 16

6 while 16 != 0: --- True

7 rem = x % y

rem = 36 % 16

rem = 4

8 x = y

x = 16

9 y = rem

y = 4

6 while 4 != 0: --- True

7 rem = x % y

rem = 16 % 4

rem = 0

8 x = y

x = 4

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 4

2. gcd(0, 97) = 97

1 in function gcd(x=0, y=97)

2 if 0 == 0: --- True

3 return y

return 97

3. gcd(26, 0) = 26

1 in function gcd(x=26, y=0)

2 if 26 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 26

4. hex(218) = 'DA'

3 in function hex(number=218)

4 if 218 == 0: --- False

6 res = ''

res = ''

7 while 218 > 0: --- True

8 digit = number % 16

digit = 218 % 16

digit = 10

9 res = DIGITS[digit] + res

res = DIGITS[10] + ''

res = 'A'

10 number = number // 16

number = 218 // 16

number = 13

7 while 13 > 0: --- True

8 digit = number % 16

digit = 13 % 16

digit = 13

9 res = DIGITS[digit] + res

res = DIGITS[13] + 'A'

res = 'DA'

10 number = number // 16

number = 13 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return DA

5. square\_equal(-1, -2, 24) = [4.0, -6.0]

3 in function square\_equal(a=-1, b=-2, c=24)

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

5 D = b\*b - 4\*a\*c

D = -2\*-2 - 4\*-1\*24

D = 100

6 if 100 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (--2 - sqrt(100)) / (2\*-1)

x1 = 4.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (--2 + sqrt(100)) / (2\*-1)

x2 = -6.0

9 return [x1, x2]

return [4.0, -6.0]

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

3 in function square\_equal(a=-16, b=48, c=-36)

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

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [1.5]

7. square\_equal(94, 5, 2) = []

3 in function square\_equal(a=94, b=5, c=2)

4 if 94 != 0: --- True

5 D = b\*b - 4\*a\*c

D = 5\*5 - 4\*94\*2

D = -727

6 if -727 > 0: --- False

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

12 else:

13 return []

return []

8. square\_equal(0, 8, -34) = [4.25]

3 in function square\_equal(a=0, b=8, c=-34)

4 if 0 != 0: --- False

14 else:

15 if 8 != 0: --- True

16 return [-c / b]

return [4.25]

9. square\_equal(0, 0, 76) = []

3 in function square\_equal(a=0, b=0, c=76)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-2

1. gcd(70, 85) = 5

1 in function gcd(x=70, y=85)

2 if 70 == 0: --- False

4 if 85 == 0: --- False

6 while 85 != 0: --- True

7 rem = x % y

rem = 70 % 85

rem = 70

8 x = y

x = 85

9 y = rem

y = 70

6 while 70 != 0: --- True

7 rem = x % y

rem = 85 % 70

rem = 15

8 x = y

x = 70

9 y = rem

y = 15

6 while 15 != 0: --- True

7 rem = x % y

rem = 70 % 15

rem = 10

8 x = y

x = 15

9 y = rem

y = 10

6 while 10 != 0: --- True

7 rem = x % y

rem = 15 % 10

rem = 5

8 x = y

x = 10

9 y = rem

y = 5

6 while 5 != 0: --- True

7 rem = x % y

rem = 10 % 5

rem = 0

8 x = y

x = 5

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 5

2. gcd(0, 78) = 78

1 in function gcd(x=0, y=78)

2 if 0 == 0: --- True

3 return y

return 78

3. gcd(50, 0) = 50

1 in function gcd(x=50, y=0)

2 if 50 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 50

4. hex(239) = 'EF'

3 in function hex(number=239)

4 if 239 == 0: --- False

6 res = ''

res = ''

7 while 239 > 0: --- True

8 digit = number % 16

digit = 239 % 16

digit = 15

9 res = DIGITS[digit] + res

res = DIGITS[15] + ''

res = 'F'

10 number = number // 16

number = 239 // 16

number = 14

7 while 14 > 0: --- True

8 digit = number % 16

digit = 14 % 16

digit = 14

9 res = DIGITS[digit] + res

res = DIGITS[14] + 'F'

res = 'EF'

10 number = number // 16

number = 14 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return EF

5. square\_equal(1, 18, 17) = [-17.0, -1.0]

3 in function square\_equal(a=1, b=18, c=17)

4 if 1 != 0: --- True

5 D = b\*b - 4\*a\*c

D = 18\*18 - 4\*1\*17

D = 256

6 if 256 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (-18 - sqrt(256)) / (2\*1)

x1 = -17.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (-18 + sqrt(256)) / (2\*1)

x2 = -1.0

9 return [x1, x2]

return [-17.0, -1.0]

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

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

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

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [-4.0]

7. square\_equal(5, -3, 48) = []

3 in function square\_equal(a=5, b=-3, c=48)

4 if 5 != 0: --- True

5 D = b\*b - 4\*a\*c

D = -3\*-3 - 4\*5\*48

D = -951

6 if -951 > 0: --- False

10 elif -951 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, -12, 81) = [6.75]

3 in function square\_equal(a=0, b=-12, c=81)

4 if 0 != 0: --- False

14 else:

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

16 return [-c / b]

return [6.75]

9. square\_equal(0, 0, -95) = []

3 in function square\_equal(a=0, b=0, c=-95)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-3

1. gcd(52, 80) = 4

1 in function gcd(x=52, y=80)

2 if 52 == 0: --- False

4 if 80 == 0: --- False

6 while 80 != 0: --- True

7 rem = x % y

rem = 52 % 80

rem = 52

8 x = y

x = 80

9 y = rem

y = 52

6 while 52 != 0: --- True

7 rem = x % y

rem = 80 % 52

rem = 28

8 x = y

x = 52

9 y = rem

y = 28

6 while 28 != 0: --- True

7 rem = x % y

rem = 52 % 28

rem = 24

8 x = y

x = 28

9 y = rem

y = 24

6 while 24 != 0: --- True

7 rem = x % y

rem = 28 % 24

rem = 4

8 x = y

x = 24

9 y = rem

y = 4

6 while 4 != 0: --- True

7 rem = x % y

rem = 24 % 4

rem = 0

8 x = y

x = 4

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 4

2. gcd(0, 10) = 10

1 in function gcd(x=0, y=10)

2 if 0 == 0: --- True

3 return y

return 10

3. gcd(97, 0) = 97

1 in function gcd(x=97, y=0)

2 if 97 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 97

4. hex(219) = 'DB'

3 in function hex(number=219)

4 if 219 == 0: --- False

6 res = ''

res = ''

7 while 219 > 0: --- True

8 digit = number % 16

digit = 219 % 16

digit = 11

9 res = DIGITS[digit] + res

res = DIGITS[11] + ''

res = 'B'

10 number = number // 16

number = 219 // 16

number = 13

7 while 13 > 0: --- True

8 digit = number % 16

digit = 13 % 16

digit = 13

9 res = DIGITS[digit] + res

res = DIGITS[13] + 'B'

res = 'DB'

10 number = number // 16

number = 13 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return DB

5. square\_equal(4, -88, 84) = [1.0, 21.0]

3 in function square\_equal(a=4, b=-88, c=84)

4 if 4 != 0: --- True

5 D = b\*b - 4\*a\*c

D = -88\*-88 - 4\*4\*84

D = 6400

6 if 6400 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (--88 - sqrt(6400)) / (2\*4)

x1 = 1.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (--88 + sqrt(6400)) / (2\*4)

x2 = 21.0

9 return [x1, x2]

return [1.0, 21.0]

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

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

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

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [-0.25]

7. square\_equal(4, -22, 42) = []

3 in function square\_equal(a=4, b=-22, c=42)

4 if 4 != 0: --- True

5 D = b\*b - 4\*a\*c

D = -22\*-22 - 4\*4\*42

D = -188

6 if -188 > 0: --- False

10 elif -188 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, 9, 18) = [-2.0]

3 in function square\_equal(a=0, b=9, c=18)

4 if 0 != 0: --- False

14 else:

15 if 9 != 0: --- True

16 return [-c / b]

return [-2.0]

9. square\_equal(0, 0, 47) = []

3 in function square\_equal(a=0, b=0, c=47)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-4

1. gcd(50, 80) = 10

1 in function gcd(x=50, y=80)

2 if 50 == 0: --- False

4 if 80 == 0: --- False

6 while 80 != 0: --- True

7 rem = x % y

rem = 50 % 80

rem = 50

8 x = y

x = 80

9 y = rem

y = 50

6 while 50 != 0: --- True

7 rem = x % y

rem = 80 % 50

rem = 30

8 x = y

x = 50

9 y = rem

y = 30

6 while 30 != 0: --- True

7 rem = x % y

rem = 50 % 30

rem = 20

8 x = y

x = 30

9 y = rem

y = 20

6 while 20 != 0: --- True

7 rem = x % y

rem = 30 % 20

rem = 10

8 x = y

x = 20

9 y = rem

y = 10

6 while 10 != 0: --- True

7 rem = x % y

rem = 20 % 10

rem = 0

8 x = y

x = 10

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 10

2. gcd(0, 73) = 73

1 in function gcd(x=0, y=73)

2 if 0 == 0: --- True

3 return y

return 73

3. gcd(64, 0) = 64

1 in function gcd(x=64, y=0)

2 if 64 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 64

4. hex(244) = 'F4'

3 in function hex(number=244)

4 if 244 == 0: --- False

6 res = ''

res = ''

7 while 244 > 0: --- True

8 digit = number % 16

digit = 244 % 16

digit = 4

9 res = DIGITS[digit] + res

res = DIGITS[4] + ''

res = '4'

10 number = number // 16

number = 244 // 16

number = 15

7 while 15 > 0: --- True

8 digit = number % 16

digit = 15 % 16

digit = 15

9 res = DIGITS[digit] + res

res = DIGITS[15] + '4'

res = 'F4'

10 number = number // 16

number = 15 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return F4

5. square\_equal(-1, 25, 54) = [27.0, -2.0]

3 in function square\_equal(a=-1, b=25, c=54)

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

5 D = b\*b - 4\*a\*c

D = 25\*25 - 4\*-1\*54

D = 841

6 if 841 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (-25 - sqrt(841)) / (2\*-1)

x1 = 27.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (-25 + sqrt(841)) / (2\*-1)

x2 = -2.0

9 return [x1, x2]

return [27.0, -2.0]

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

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

4 if 16 != 0: --- True

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [0.5]

7. square\_equal(7, -5, 25) = []

3 in function square\_equal(a=7, b=-5, c=25)

4 if 7 != 0: --- True

5 D = b\*b - 4\*a\*c

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

D = -675

6 if -675 > 0: --- False

10 elif -675 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, -15, 63) = [4.2]

3 in function square\_equal(a=0, b=-15, c=63)

4 if 0 != 0: --- False

14 else:

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

16 return [-c / b]

return [4.2]

9. square\_equal(0, 0, 20) = []

3 in function square\_equal(a=0, b=0, c=20)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-5

1. gcd(80, 92) = 4

1 in function gcd(x=80, y=92)

2 if 80 == 0: --- False

4 if 92 == 0: --- False

6 while 92 != 0: --- True

7 rem = x % y

rem = 80 % 92

rem = 80

8 x = y

x = 92

9 y = rem

y = 80

6 while 80 != 0: --- True

7 rem = x % y

rem = 92 % 80

rem = 12

8 x = y

x = 80

9 y = rem

y = 12

6 while 12 != 0: --- True

7 rem = x % y

rem = 80 % 12

rem = 8

8 x = y

x = 12

9 y = rem

y = 8

6 while 8 != 0: --- True

7 rem = x % y

rem = 12 % 8

rem = 4

8 x = y

x = 8

9 y = rem

y = 4

6 while 4 != 0: --- True

7 rem = x % y

rem = 8 % 4

rem = 0

8 x = y

x = 4

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 4

2. gcd(0, 2) = 2

1 in function gcd(x=0, y=2)

2 if 0 == 0: --- True

3 return y

return 2

3. gcd(81, 0) = 81

1 in function gcd(x=81, y=0)

2 if 81 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 81

4. hex(224) = 'E0'

3 in function hex(number=224)

4 if 224 == 0: --- False

6 res = ''

res = ''

7 while 224 > 0: --- True

8 digit = number % 16

digit = 224 % 16

digit = 0

9 res = DIGITS[digit] + res

res = DIGITS[0] + ''

res = '0'

10 number = number // 16

number = 224 // 16

number = 14

7 while 14 > 0: --- True

8 digit = number % 16

digit = 14 % 16

digit = 14

9 res = DIGITS[digit] + res

res = DIGITS[14] + '0'

res = 'E0'

10 number = number // 16

number = 14 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return E0

5. square\_equal(1, -22, 96) = [6.0, 16.0]

3 in function square\_equal(a=1, b=-22, c=96)

4 if 1 != 0: --- True

5 D = b\*b - 4\*a\*c

D = -22\*-22 - 4\*1\*96

D = 100

6 if 100 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (--22 - sqrt(100)) / (2\*1)

x1 = 6.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (--22 + sqrt(100)) / (2\*1)

x2 = 16.0

9 return [x1, x2]

return [6.0, 16.0]

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

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

4 if 1 != 0: --- True

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [1.0]

7. square\_equal(4, 8, 56) = []

3 in function square\_equal(a=4, b=8, c=56)

4 if 4 != 0: --- True

5 D = b\*b - 4\*a\*c

D = 8\*8 - 4\*4\*56

D = -832

6 if -832 > 0: --- False

10 elif -832 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, -3, 42) = [14.0]

3 in function square\_equal(a=0, b=-3, c=42)

4 if 0 != 0: --- False

14 else:

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

16 return [-c / b]

return [14.0]

9. square\_equal(0, 0, -62) = []

3 in function square\_equal(a=0, b=0, c=-62)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-6

1. gcd(44, 100) = 4

1 in function gcd(x=44, y=100)

2 if 44 == 0: --- False

4 if 100 == 0: --- False

6 while 100 != 0: --- True

7 rem = x % y

rem = 44 % 100

rem = 44

8 x = y

x = 100

9 y = rem

y = 44

6 while 44 != 0: --- True

7 rem = x % y

rem = 100 % 44

rem = 12

8 x = y

x = 44

9 y = rem

y = 12

6 while 12 != 0: --- True

7 rem = x % y

rem = 44 % 12

rem = 8

8 x = y

x = 12

9 y = rem

y = 8

6 while 8 != 0: --- True

7 rem = x % y

rem = 12 % 8

rem = 4

8 x = y

x = 8

9 y = rem

y = 4

6 while 4 != 0: --- True

7 rem = x % y

rem = 8 % 4

rem = 0

8 x = y

x = 4

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 4

2. gcd(0, 50) = 50

1 in function gcd(x=0, y=50)

2 if 0 == 0: --- True

3 return y

return 50

3. gcd(2, 0) = 2

1 in function gcd(x=2, y=0)

2 if 2 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 2

4. hex(183) = 'B7'

3 in function hex(number=183)

4 if 183 == 0: --- False

6 res = ''

res = ''

7 while 183 > 0: --- True

8 digit = number % 16

digit = 183 % 16

digit = 7

9 res = DIGITS[digit] + res

res = DIGITS[7] + ''

res = '7'

10 number = number // 16

number = 183 // 16

number = 11

7 while 11 > 0: --- True

8 digit = number % 16

digit = 11 % 16

digit = 11

9 res = DIGITS[digit] + res

res = DIGITS[11] + '7'

res = 'B7'

10 number = number // 16

number = 11 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return B7

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

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

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

5 D = b\*b - 4\*a\*c

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

D = 8100

6 if 8100 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

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

x1 = 1.6

8 x2 = (-b + sqrt(D)) / (2\*a)

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

x2 = -2.0

9 return [x1, x2]

return [1.6, -2.0]

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

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

4 if 64 != 0: --- True

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [0.75]

7. square\_equal(-22, -20, -8) = []

3 in function square\_equal(a=-22, b=-20, c=-8)

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

5 D = b\*b - 4\*a\*c

D = -20\*-20 - 4\*-22\*-8

D = -304

6 if -304 > 0: --- False

10 elif -304 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, 50, -64) = [1.28]

3 in function square\_equal(a=0, b=50, c=-64)

4 if 0 != 0: --- False

14 else:

15 if 50 != 0: --- True

16 return [-c / b]

return [1.28]

9. square\_equal(0, 0, -88) = []

3 in function square\_equal(a=0, b=0, c=-88)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-7

1. gcd(56, 88) = 8

1 in function gcd(x=56, y=88)

2 if 56 == 0: --- False

4 if 88 == 0: --- False

6 while 88 != 0: --- True

7 rem = x % y

rem = 56 % 88

rem = 56

8 x = y

x = 88

9 y = rem

y = 56

6 while 56 != 0: --- True

7 rem = x % y

rem = 88 % 56

rem = 32

8 x = y

x = 56

9 y = rem

y = 32

6 while 32 != 0: --- True

7 rem = x % y

rem = 56 % 32

rem = 24

8 x = y

x = 32

9 y = rem

y = 24

6 while 24 != 0: --- True

7 rem = x % y

rem = 32 % 24

rem = 8

8 x = y

x = 24

9 y = rem

y = 8

6 while 8 != 0: --- True

7 rem = x % y

rem = 24 % 8

rem = 0

8 x = y

x = 8

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 8

2. gcd(0, 53) = 53

1 in function gcd(x=0, y=53)

2 if 0 == 0: --- True

3 return y

return 53

3. gcd(72, 0) = 72

1 in function gcd(x=72, y=0)

2 if 72 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 72

4. hex(186) = 'BA'

3 in function hex(number=186)

4 if 186 == 0: --- False

6 res = ''

res = ''

7 while 186 > 0: --- True

8 digit = number % 16

digit = 186 % 16

digit = 10

9 res = DIGITS[digit] + res

res = DIGITS[10] + ''

res = 'A'

10 number = number // 16

number = 186 // 16

number = 11

7 while 11 > 0: --- True

8 digit = number % 16

digit = 11 % 16

digit = 11

9 res = DIGITS[digit] + res

res = DIGITS[11] + 'A'

res = 'BA'

10 number = number // 16

number = 11 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return BA

5. square\_equal(-16, -52, 68) = [1.0, -4.25]

3 in function square\_equal(a=-16, b=-52, c=68)

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

5 D = b\*b - 4\*a\*c

D = -52\*-52 - 4\*-16\*68

D = 7056

6 if 7056 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (--52 - sqrt(7056)) / (2\*-16)

x1 = 1.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (--52 + sqrt(7056)) / (2\*-16)

x2 = -4.25

9 return [x1, x2]

return [1.0, -4.25]

6. square\_equal(4, -4, 1) = [0.5]

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

4 if 4 != 0: --- True

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [0.5]

7. square\_equal(19, -10, 2) = []

3 in function square\_equal(a=19, b=-10, c=2)

4 if 19 != 0: --- True

5 D = b\*b - 4\*a\*c

D = -10\*-10 - 4\*19\*2

D = -52

6 if -52 > 0: --- False

10 elif -52 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, 1, 7) = [-7.0]

3 in function square\_equal(a=0, b=1, c=7)

4 if 0 != 0: --- False

14 else:

15 if 1 != 0: --- True

16 return [-c / b]

return [-7.0]

9. square\_equal(0, 0, -4) = []

3 in function square\_equal(a=0, b=0, c=-4)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-8

1. gcd(35, 56) = 7

1 in function gcd(x=35, y=56)

2 if 35 == 0: --- False

4 if 56 == 0: --- False

6 while 56 != 0: --- True

7 rem = x % y

rem = 35 % 56

rem = 35

8 x = y

x = 56

9 y = rem

y = 35

6 while 35 != 0: --- True

7 rem = x % y

rem = 56 % 35

rem = 21

8 x = y

x = 35

9 y = rem

y = 21

6 while 21 != 0: --- True

7 rem = x % y

rem = 35 % 21

rem = 14

8 x = y

x = 21

9 y = rem

y = 14

6 while 14 != 0: --- True

7 rem = x % y

rem = 21 % 14

rem = 7

8 x = y

x = 14

9 y = rem

y = 7

6 while 7 != 0: --- True

7 rem = x % y

rem = 14 % 7

rem = 0

8 x = y

x = 7

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 7

2. gcd(0, 80) = 80

1 in function gcd(x=0, y=80)

2 if 0 == 0: --- True

3 return y

return 80

3. gcd(100, 0) = 100

1 in function gcd(x=100, y=0)

2 if 100 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 100

4. hex(199) = 'C7'

3 in function hex(number=199)

4 if 199 == 0: --- False

6 res = ''

res = ''

7 while 199 > 0: --- True

8 digit = number % 16

digit = 199 % 16

digit = 7

9 res = DIGITS[digit] + res

res = DIGITS[7] + ''

res = '7'

10 number = number // 16

number = 199 // 16

number = 12

7 while 12 > 0: --- True

8 digit = number % 16

digit = 12 % 16

digit = 12

9 res = DIGITS[digit] + res

res = DIGITS[12] + '7'

res = 'C7'

10 number = number // 16

number = 12 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return C7

5. square\_equal(1, -21, 20) = [1.0, 20.0]

3 in function square\_equal(a=1, b=-21, c=20)

4 if 1 != 0: --- True

5 D = b\*b - 4\*a\*c

D = -21\*-21 - 4\*1\*20

D = 361

6 if 361 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (--21 - sqrt(361)) / (2\*1)

x1 = 1.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (--21 + sqrt(361)) / (2\*1)

x2 = 20.0

9 return [x1, x2]

return [1.0, 20.0]

6. square\_equal(36, -72, 36) = [1.0]

3 in function square\_equal(a=36, b=-72, c=36)

4 if 36 != 0: --- True

5 D = b\*b - 4\*a\*c

D = -72\*-72 - 4\*36\*36

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [1.0]

7. square\_equal(-10, 21, -14) = []

3 in function square\_equal(a=-10, b=21, c=-14)

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

5 D = b\*b - 4\*a\*c

D = 21\*21 - 4\*-10\*-14

D = -119

6 if -119 > 0: --- False

10 elif -119 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, 48, -48) = [1.0]

3 in function square\_equal(a=0, b=48, c=-48)

4 if 0 != 0: --- False

14 else:

15 if 48 != 0: --- True

16 return [-c / b]

return [1.0]

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

3 in function square\_equal(a=0, b=0, c=-98)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-9

1. gcd(55, 85) = 5

1 in function gcd(x=55, y=85)

2 if 55 == 0: --- False

4 if 85 == 0: --- False

6 while 85 != 0: --- True

7 rem = x % y

rem = 55 % 85

rem = 55

8 x = y

x = 85

9 y = rem

y = 55

6 while 55 != 0: --- True

7 rem = x % y

rem = 85 % 55

rem = 30

8 x = y

x = 55

9 y = rem

y = 30

6 while 30 != 0: --- True

7 rem = x % y

rem = 55 % 30

rem = 25

8 x = y

x = 30

9 y = rem

y = 25

6 while 25 != 0: --- True

7 rem = x % y

rem = 30 % 25

rem = 5

8 x = y

x = 25

9 y = rem

y = 5

6 while 5 != 0: --- True

7 rem = x % y

rem = 25 % 5

rem = 0

8 x = y

x = 5

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 5

2. gcd(0, 72) = 72

1 in function gcd(x=0, y=72)

2 if 0 == 0: --- True

3 return y

return 72

3. gcd(30, 0) = 30

1 in function gcd(x=30, y=0)

2 if 30 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 30

4. hex(242) = 'F2'

3 in function hex(number=242)

4 if 242 == 0: --- False

6 res = ''

res = ''

7 while 242 > 0: --- True

8 digit = number % 16

digit = 242 % 16

digit = 2

9 res = DIGITS[digit] + res

res = DIGITS[2] + ''

res = '2'

10 number = number // 16

number = 242 // 16

number = 15

7 while 15 > 0: --- True

8 digit = number % 16

digit = 15 % 16

digit = 15

9 res = DIGITS[digit] + res

res = DIGITS[15] + '2'

res = 'F2'

10 number = number // 16

number = 15 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return F2

5. square\_equal(1, -90, 89) = [1.0, 89.0]

3 in function square\_equal(a=1, b=-90, c=89)

4 if 1 != 0: --- True

5 D = b\*b - 4\*a\*c

D = -90\*-90 - 4\*1\*89

D = 7744

6 if 7744 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (--90 - sqrt(7744)) / (2\*1)

x1 = 1.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (--90 + sqrt(7744)) / (2\*1)

x2 = 89.0

9 return [x1, x2]

return [1.0, 89.0]

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

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

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

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [4.0]

7. square\_equal(-2, -5, -62) = []

3 in function square\_equal(a=-2, b=-5, c=-62)

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

5 D = b\*b - 4\*a\*c

D = -5\*-5 - 4\*-2\*-62

D = -471

6 if -471 > 0: --- False

10 elif -471 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, -58, 58) = [1.0]

3 in function square\_equal(a=0, b=-58, c=58)

4 if 0 != 0: --- False

14 else:

15 if -58 != 0: --- True

16 return [-c / b]

return [1.0]

9. square\_equal(0, 0, -92) = []

3 in function square\_equal(a=0, b=0, c=-92)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-10

1. gcd(35, 60) = 5

1 in function gcd(x=35, y=60)

2 if 35 == 0: --- False

4 if 60 == 0: --- False

6 while 60 != 0: --- True

7 rem = x % y

rem = 35 % 60

rem = 35

8 x = y

x = 60

9 y = rem

y = 35

6 while 35 != 0: --- True

7 rem = x % y

rem = 60 % 35

rem = 25

8 x = y

x = 35

9 y = rem

y = 25

6 while 25 != 0: --- True

7 rem = x % y

rem = 35 % 25

rem = 10

8 x = y

x = 25

9 y = rem

y = 10

6 while 10 != 0: --- True

7 rem = x % y

rem = 25 % 10

rem = 5

8 x = y

x = 10

9 y = rem

y = 5

6 while 5 != 0: --- True

7 rem = x % y

rem = 10 % 5

rem = 0

8 x = y

x = 5

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 5

2. gcd(0, 93) = 93

1 in function gcd(x=0, y=93)

2 if 0 == 0: --- True

3 return y

return 93

3. gcd(67, 0) = 67

1 in function gcd(x=67, y=0)

2 if 67 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 67

4. hex(161) = 'A1'

3 in function hex(number=161)

4 if 161 == 0: --- False

6 res = ''

res = ''

7 while 161 > 0: --- True

8 digit = number % 16

digit = 161 % 16

digit = 1

9 res = DIGITS[digit] + res

res = DIGITS[1] + ''

res = '1'

10 number = number // 16

number = 161 // 16

number = 10

7 while 10 > 0: --- True

8 digit = number % 16

digit = 10 % 16

digit = 10

9 res = DIGITS[digit] + res

res = DIGITS[10] + '1'

res = 'A1'

10 number = number // 16

number = 10 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return A1

5. square\_equal(-16, -24, 91) = [1.75, -3.25]

3 in function square\_equal(a=-16, b=-24, c=91)

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

5 D = b\*b - 4\*a\*c

D = -24\*-24 - 4\*-16\*91

D = 6400

6 if 6400 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (--24 - sqrt(6400)) / (2\*-16)

x1 = 1.75

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (--24 + sqrt(6400)) / (2\*-16)

x2 = -3.25

9 return [x1, x2]

return [1.75, -3.25]

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

3 in function square\_equal(a=25, b=70, c=49)

4 if 25 != 0: --- True

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [-1.4]

7. square\_equal(-12, 3, -19) = []

3 in function square\_equal(a=-12, b=3, c=-19)

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

5 D = b\*b - 4\*a\*c

D = 3\*3 - 4\*-12\*-19

D = -903

6 if -903 > 0: --- False

10 elif -903 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, 30, 63) = [-2.1]

3 in function square\_equal(a=0, b=30, c=63)

4 if 0 != 0: --- False

14 else:

15 if 30 != 0: --- True

16 return [-c / b]

return [-2.1]

9. square\_equal(0, 0, 71) = []

3 in function square\_equal(a=0, b=0, c=71)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-11

1. gcd(36, 92) = 4

1 in function gcd(x=36, y=92)

2 if 36 == 0: --- False

4 if 92 == 0: --- False

6 while 92 != 0: --- True

7 rem = x % y

rem = 36 % 92

rem = 36

8 x = y

x = 92

9 y = rem

y = 36

6 while 36 != 0: --- True

7 rem = x % y

rem = 92 % 36

rem = 20

8 x = y

x = 36

9 y = rem

y = 20

6 while 20 != 0: --- True

7 rem = x % y

rem = 36 % 20

rem = 16

8 x = y

x = 20

9 y = rem

y = 16

6 while 16 != 0: --- True

7 rem = x % y

rem = 20 % 16

rem = 4

8 x = y

x = 16

9 y = rem

y = 4

6 while 4 != 0: --- True

7 rem = x % y

rem = 16 % 4

rem = 0

8 x = y

x = 4

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 4

2. gcd(0, 88) = 88

1 in function gcd(x=0, y=88)

2 if 0 == 0: --- True

3 return y

return 88

3. gcd(24, 0) = 24

1 in function gcd(x=24, y=0)

2 if 24 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 24

4. hex(164) = 'A4'

3 in function hex(number=164)

4 if 164 == 0: --- False

6 res = ''

res = ''

7 while 164 > 0: --- True

8 digit = number % 16

digit = 164 % 16

digit = 4

9 res = DIGITS[digit] + res

res = DIGITS[4] + ''

res = '4'

10 number = number // 16

number = 164 // 16

number = 10

7 while 10 > 0: --- True

8 digit = number % 16

digit = 10 % 16

digit = 10

9 res = DIGITS[digit] + res

res = DIGITS[10] + '4'

res = 'A4'

10 number = number // 16

number = 10 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return A4

5. square\_equal(-1, -76, 77) = [1.0, -77.0]

3 in function square\_equal(a=-1, b=-76, c=77)

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

5 D = b\*b - 4\*a\*c

D = -76\*-76 - 4\*-1\*77

D = 6084

6 if 6084 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (--76 - sqrt(6084)) / (2\*-1)

x1 = 1.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (--76 + sqrt(6084)) / (2\*-1)

x2 = -77.0

9 return [x1, x2]

return [1.0, -77.0]

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

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

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

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [0.4]

7. square\_equal(-8, -4, -29) = []

3 in function square\_equal(a=-8, b=-4, c=-29)

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

5 D = b\*b - 4\*a\*c

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

D = -912

6 if -912 > 0: --- False

10 elif -912 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, -10, -34) = [-3.4]

3 in function square\_equal(a=0, b=-10, c=-34)

4 if 0 != 0: --- False

14 else:

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

16 return [-c / b]

return [-3.4]

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

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

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-12

1. gcd(35, 55) = 5

1 in function gcd(x=35, y=55)

2 if 35 == 0: --- False

4 if 55 == 0: --- False

6 while 55 != 0: --- True

7 rem = x % y

rem = 35 % 55

rem = 35

8 x = y

x = 55

9 y = rem

y = 35

6 while 35 != 0: --- True

7 rem = x % y

rem = 55 % 35

rem = 20

8 x = y

x = 35

9 y = rem

y = 20

6 while 20 != 0: --- True

7 rem = x % y

rem = 35 % 20

rem = 15

8 x = y

x = 20

9 y = rem

y = 15

6 while 15 != 0: --- True

7 rem = x % y

rem = 20 % 15

rem = 5

8 x = y

x = 15

9 y = rem

y = 5

6 while 5 != 0: --- True

7 rem = x % y

rem = 15 % 5

rem = 0

8 x = y

x = 5

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 5

2. gcd(0, 94) = 94

1 in function gcd(x=0, y=94)

2 if 0 == 0: --- True

3 return y

return 94

3. gcd(13, 0) = 13

1 in function gcd(x=13, y=0)

2 if 13 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 13

4. hex(184) = 'B8'

3 in function hex(number=184)

4 if 184 == 0: --- False

6 res = ''

res = ''

7 while 184 > 0: --- True

8 digit = number % 16

digit = 184 % 16

digit = 8

9 res = DIGITS[digit] + res

res = DIGITS[8] + ''

res = '8'

10 number = number // 16

number = 184 // 16

number = 11

7 while 11 > 0: --- True

8 digit = number % 16

digit = 11 % 16

digit = 11

9 res = DIGITS[digit] + res

res = DIGITS[11] + '8'

res = 'B8'

10 number = number // 16

number = 11 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return B8

5. square\_equal(1, 57, 56) = [-56.0, -1.0]

3 in function square\_equal(a=1, b=57, c=56)

4 if 1 != 0: --- True

5 D = b\*b - 4\*a\*c

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

D = 3025

6 if 3025 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (-57 - sqrt(3025)) / (2\*1)

x1 = -56.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (-57 + sqrt(3025)) / (2\*1)

x2 = -1.0

9 return [x1, x2]

return [-56.0, -1.0]

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

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

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

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [-1.0]

7. square\_equal(4, 4, 57) = []

3 in function square\_equal(a=4, b=4, c=57)

4 if 4 != 0: --- True

5 D = b\*b - 4\*a\*c

D = 4\*4 - 4\*4\*57

D = -896

6 if -896 > 0: --- False

10 elif -896 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, -4, 14) = [3.5]

3 in function square\_equal(a=0, b=-4, c=14)

4 if 0 != 0: --- False

14 else:

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

16 return [-c / b]

return [3.5]

9. square\_equal(0, 0, 58) = []

3 in function square\_equal(a=0, b=0, c=58)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-13

1. gcd(55, 70) = 5

1 in function gcd(x=55, y=70)

2 if 55 == 0: --- False

4 if 70 == 0: --- False

6 while 70 != 0: --- True

7 rem = x % y

rem = 55 % 70

rem = 55

8 x = y

x = 70

9 y = rem

y = 55

6 while 55 != 0: --- True

7 rem = x % y

rem = 70 % 55

rem = 15

8 x = y

x = 55

9 y = rem

y = 15

6 while 15 != 0: --- True

7 rem = x % y

rem = 55 % 15

rem = 10

8 x = y

x = 15

9 y = rem

y = 10

6 while 10 != 0: --- True

7 rem = x % y

rem = 15 % 10

rem = 5

8 x = y

x = 10

9 y = rem

y = 5

6 while 5 != 0: --- True

7 rem = x % y

rem = 10 % 5

rem = 0

8 x = y

x = 5

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 5

2. gcd(0, 90) = 90

1 in function gcd(x=0, y=90)

2 if 0 == 0: --- True

3 return y

return 90

3. gcd(49, 0) = 49

1 in function gcd(x=49, y=0)

2 if 49 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 49

4. hex(169) = 'A9'

3 in function hex(number=169)

4 if 169 == 0: --- False

6 res = ''

res = ''

7 while 169 > 0: --- True

8 digit = number % 16

digit = 169 % 16

digit = 9

9 res = DIGITS[digit] + res

res = DIGITS[9] + ''

res = '9'

10 number = number // 16

number = 169 // 16

number = 10

7 while 10 > 0: --- True

8 digit = number % 16

digit = 10 % 16

digit = 10

9 res = DIGITS[digit] + res

res = DIGITS[10] + '9'

res = 'A9'

10 number = number // 16

number = 10 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return A9

5. square\_equal(25, -75, 44) = [0.8, 2.2]

3 in function square\_equal(a=25, b=-75, c=44)

4 if 25 != 0: --- True

5 D = b\*b - 4\*a\*c

D = -75\*-75 - 4\*25\*44

D = 1225

6 if 1225 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (--75 - sqrt(1225)) / (2\*25)

x1 = 0.8

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (--75 + sqrt(1225)) / (2\*25)

x2 = 2.2

9 return [x1, x2]

return [0.8, 2.2]

6. square\_equal(-1, -12, -36) = [-6.0]

3 in function square\_equal(a=-1, b=-12, c=-36)

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

5 D = b\*b - 4\*a\*c

D = -12\*-12 - 4\*-1\*-36

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [-6.0]

7. square\_equal(-25, -9, -1) = []

3 in function square\_equal(a=-25, b=-9, c=-1)

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

5 D = b\*b - 4\*a\*c

D = -9\*-9 - 4\*-25\*-1

D = -19

6 if -19 > 0: --- False

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

12 else:

13 return []

return []

8. square\_equal(0, 20, 30) = [-1.5]

3 in function square\_equal(a=0, b=20, c=30)

4 if 0 != 0: --- False

14 else:

15 if 20 != 0: --- True

16 return [-c / b]

return [-1.5]

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

3 in function square\_equal(a=0, b=0, c=-5)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-14

1. gcd(90, 65) = 5

1 in function gcd(x=90, y=65)

2 if 90 == 0: --- False

4 if 65 == 0: --- False

6 while 65 != 0: --- True

7 rem = x % y

rem = 90 % 65

rem = 25

8 x = y

x = 65

9 y = rem

y = 25

6 while 25 != 0: --- True

7 rem = x % y

rem = 65 % 25

rem = 15

8 x = y

x = 25

9 y = rem

y = 15

6 while 15 != 0: --- True

7 rem = x % y

rem = 25 % 15

rem = 10

8 x = y

x = 15

9 y = rem

y = 10

6 while 10 != 0: --- True

7 rem = x % y

rem = 15 % 10

rem = 5

8 x = y

x = 10

9 y = rem

y = 5

6 while 5 != 0: --- True

7 rem = x % y

rem = 10 % 5

rem = 0

8 x = y

x = 5

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 5

2. gcd(0, 36) = 36

1 in function gcd(x=0, y=36)

2 if 0 == 0: --- True

3 return y

return 36

3. gcd(7, 0) = 7

1 in function gcd(x=7, y=0)

2 if 7 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 7

4. hex(195) = 'C3'

3 in function hex(number=195)

4 if 195 == 0: --- False

6 res = ''

res = ''

7 while 195 > 0: --- True

8 digit = number % 16

digit = 195 % 16

digit = 3

9 res = DIGITS[digit] + res

res = DIGITS[3] + ''

res = '3'

10 number = number // 16

number = 195 // 16

number = 12

7 while 12 > 0: --- True

8 digit = number % 16

digit = 12 % 16

digit = 12

9 res = DIGITS[digit] + res

res = DIGITS[12] + '3'

res = 'C3'

10 number = number // 16

number = 12 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return C3

5. square\_equal(4, -58, 54) = [1.0, 13.5]

3 in function square\_equal(a=4, b=-58, c=54)

4 if 4 != 0: --- True

5 D = b\*b - 4\*a\*c

D = -58\*-58 - 4\*4\*54

D = 2500

6 if 2500 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (--58 - sqrt(2500)) / (2\*4)

x1 = 1.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (--58 + sqrt(2500)) / (2\*4)

x2 = 13.5

9 return [x1, x2]

return [1.0, 13.5]

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

3 in function square\_equal(a=64, b=64, c=16)

4 if 64 != 0: --- True

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [-0.5]

7. square\_equal(6, 29, 38) = []

3 in function square\_equal(a=6, b=29, c=38)

4 if 6 != 0: --- True

5 D = b\*b - 4\*a\*c

D = 29\*29 - 4\*6\*38

D = -71

6 if -71 > 0: --- False

10 elif -71 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, -4, -3) = [-0.75]

3 in function square\_equal(a=0, b=-4, c=-3)

4 if 0 != 0: --- False

14 else:

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

16 return [-c / b]

return [-0.75]

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

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

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-15

1. gcd(36, 56) = 4

1 in function gcd(x=36, y=56)

2 if 36 == 0: --- False

4 if 56 == 0: --- False

6 while 56 != 0: --- True

7 rem = x % y

rem = 36 % 56

rem = 36

8 x = y

x = 56

9 y = rem

y = 36

6 while 36 != 0: --- True

7 rem = x % y

rem = 56 % 36

rem = 20

8 x = y

x = 36

9 y = rem

y = 20

6 while 20 != 0: --- True

7 rem = x % y

rem = 36 % 20

rem = 16

8 x = y

x = 20

9 y = rem

y = 16

6 while 16 != 0: --- True

7 rem = x % y

rem = 20 % 16

rem = 4

8 x = y

x = 16

9 y = rem

y = 4

6 while 4 != 0: --- True

7 rem = x % y

rem = 16 % 4

rem = 0

8 x = y

x = 4

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 4

2. gcd(0, 43) = 43

1 in function gcd(x=0, y=43)

2 if 0 == 0: --- True

3 return y

return 43

3. gcd(96, 0) = 96

1 in function gcd(x=96, y=0)

2 if 96 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 96

4. hex(192) = 'C0'

3 in function hex(number=192)

4 if 192 == 0: --- False

6 res = ''

res = ''

7 while 192 > 0: --- True

8 digit = number % 16

digit = 192 % 16

digit = 0

9 res = DIGITS[digit] + res

res = DIGITS[0] + ''

res = '0'

10 number = number // 16

number = 192 // 16

number = 12

7 while 12 > 0: --- True

8 digit = number % 16

digit = 12 % 16

digit = 12

9 res = DIGITS[digit] + res

res = DIGITS[12] + '0'

res = 'C0'

10 number = number // 16

number = 12 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return C0

5. square\_equal(-25, 15, 40) = [1.6, -1.0]

3 in function square\_equal(a=-25, b=15, c=40)

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

5 D = b\*b - 4\*a\*c

D = 15\*15 - 4\*-25\*40

D = 4225

6 if 4225 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (-15 - sqrt(4225)) / (2\*-25)

x1 = 1.6

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (-15 + sqrt(4225)) / (2\*-25)

x2 = -1.0

9 return [x1, x2]

return [1.6, -1.0]

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

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

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

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [2.0]

7. square\_equal(49, 5, 4) = []

3 in function square\_equal(a=49, b=5, c=4)

4 if 49 != 0: --- True

5 D = b\*b - 4\*a\*c

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

D = -759

6 if -759 > 0: --- False

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

12 else:

13 return []

return []

8. square\_equal(0, 40, -78) = [1.95]

3 in function square\_equal(a=0, b=40, c=-78)

4 if 0 != 0: --- False

14 else:

15 if 40 != 0: --- True

16 return [-c / b]

return [1.95]

9. square\_equal(0, 0, 59) = []

3 in function square\_equal(a=0, b=0, c=59)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-16

1. gcd(68, 92) = 4

1 in function gcd(x=68, y=92)

2 if 68 == 0: --- False

4 if 92 == 0: --- False

6 while 92 != 0: --- True

7 rem = x % y

rem = 68 % 92

rem = 68

8 x = y

x = 92

9 y = rem

y = 68

6 while 68 != 0: --- True

7 rem = x % y

rem = 92 % 68

rem = 24

8 x = y

x = 68

9 y = rem

y = 24

6 while 24 != 0: --- True

7 rem = x % y

rem = 68 % 24

rem = 20

8 x = y

x = 24

9 y = rem

y = 20

6 while 20 != 0: --- True

7 rem = x % y

rem = 24 % 20

rem = 4

8 x = y

x = 20

9 y = rem

y = 4

6 while 4 != 0: --- True

7 rem = x % y

rem = 20 % 4

rem = 0

8 x = y

x = 4

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 4

2. gcd(0, 46) = 46

1 in function gcd(x=0, y=46)

2 if 0 == 0: --- True

3 return y

return 46

3. gcd(82, 0) = 82

1 in function gcd(x=82, y=0)

2 if 82 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 82

4. hex(180) = 'B4'

3 in function hex(number=180)

4 if 180 == 0: --- False

6 res = ''

res = ''

7 while 180 > 0: --- True

8 digit = number % 16

digit = 180 % 16

digit = 4

9 res = DIGITS[digit] + res

res = DIGITS[4] + ''

res = '4'

10 number = number // 16

number = 180 // 16

number = 11

7 while 11 > 0: --- True

8 digit = number % 16

digit = 11 % 16

digit = 11

9 res = DIGITS[digit] + res

res = DIGITS[11] + '4'

res = 'B4'

10 number = number // 16

number = 11 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return B4

5. square\_equal(16, -80, 75) = [1.25, 3.75]

3 in function square\_equal(a=16, b=-80, c=75)

4 if 16 != 0: --- True

5 D = b\*b - 4\*a\*c

D = -80\*-80 - 4\*16\*75

D = 1600

6 if 1600 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (--80 - sqrt(1600)) / (2\*16)

x1 = 1.25

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (--80 + sqrt(1600)) / (2\*16)

x2 = 3.75

9 return [x1, x2]

return [1.25, 3.75]

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

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

4 if 1 != 0: --- True

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [9.0]

7. square\_equal(-1, -10, -70) = []

3 in function square\_equal(a=-1, b=-10, c=-70)

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

5 D = b\*b - 4\*a\*c

D = -10\*-10 - 4\*-1\*-70

D = -180

6 if -180 > 0: --- False

10 elif -180 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, -14, 14) = [1.0]

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

4 if 0 != 0: --- False

14 else:

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

16 return [-c / b]

return [1.0]

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

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

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-17

1. gcd(35, 90) = 5

1 in function gcd(x=35, y=90)

2 if 35 == 0: --- False

4 if 90 == 0: --- False

6 while 90 != 0: --- True

7 rem = x % y

rem = 35 % 90

rem = 35

8 x = y

x = 90

9 y = rem

y = 35

6 while 35 != 0: --- True

7 rem = x % y

rem = 90 % 35

rem = 20

8 x = y

x = 35

9 y = rem

y = 20

6 while 20 != 0: --- True

7 rem = x % y

rem = 35 % 20

rem = 15

8 x = y

x = 20

9 y = rem

y = 15

6 while 15 != 0: --- True

7 rem = x % y

rem = 20 % 15

rem = 5

8 x = y

x = 15

9 y = rem

y = 5

6 while 5 != 0: --- True

7 rem = x % y

rem = 15 % 5

rem = 0

8 x = y

x = 5

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 5

2. gcd(0, 15) = 15

1 in function gcd(x=0, y=15)

2 if 0 == 0: --- True

3 return y

return 15

3. gcd(68, 0) = 68

1 in function gcd(x=68, y=0)

2 if 68 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 68

4. hex(248) = 'F8'

3 in function hex(number=248)

4 if 248 == 0: --- False

6 res = ''

res = ''

7 while 248 > 0: --- True

8 digit = number % 16

digit = 248 % 16

digit = 8

9 res = DIGITS[digit] + res

res = DIGITS[8] + ''

res = '8'

10 number = number // 16

number = 248 // 16

number = 15

7 while 15 > 0: --- True

8 digit = number % 16

digit = 15 % 16

digit = 15

9 res = DIGITS[digit] + res

res = DIGITS[15] + '8'

res = 'F8'

10 number = number // 16

number = 15 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return F8

5. square\_equal(-1, 90, 91) = [91.0, -1.0]

3 in function square\_equal(a=-1, b=90, c=91)

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

5 D = b\*b - 4\*a\*c

D = 90\*90 - 4\*-1\*91

D = 8464

6 if 8464 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (-90 - sqrt(8464)) / (2\*-1)

x1 = 91.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (-90 + sqrt(8464)) / (2\*-1)

x2 = -1.0

9 return [x1, x2]

return [91.0, -1.0]

6. square\_equal(-1, 6, -9) = [3.0]

3 in function square\_equal(a=-1, b=6, c=-9)

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

5 D = b\*b - 4\*a\*c

D = 6\*6 - 4\*-1\*-9

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [3.0]

7. square\_equal(-18, 28, -13) = []

3 in function square\_equal(a=-18, b=28, c=-13)

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

5 D = b\*b - 4\*a\*c

D = 28\*28 - 4\*-18\*-13

D = -152

6 if -152 > 0: --- False

10 elif -152 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, -14, 56) = [4.0]

3 in function square\_equal(a=0, b=-14, c=56)

4 if 0 != 0: --- False

14 else:

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

16 return [-c / b]

return [4.0]

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

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

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-18

1. gcd(54, 96) = 6

1 in function gcd(x=54, y=96)

2 if 54 == 0: --- False

4 if 96 == 0: --- False

6 while 96 != 0: --- True

7 rem = x % y

rem = 54 % 96

rem = 54

8 x = y

x = 96

9 y = rem

y = 54

6 while 54 != 0: --- True

7 rem = x % y

rem = 96 % 54

rem = 42

8 x = y

x = 54

9 y = rem

y = 42

6 while 42 != 0: --- True

7 rem = x % y

rem = 54 % 42

rem = 12

8 x = y

x = 42

9 y = rem

y = 12

6 while 12 != 0: --- True

7 rem = x % y

rem = 42 % 12

rem = 6

8 x = y

x = 12

9 y = rem

y = 6

6 while 6 != 0: --- True

7 rem = x % y

rem = 12 % 6

rem = 0

8 x = y

x = 6

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 6

2. gcd(0, 89) = 89

1 in function gcd(x=0, y=89)

2 if 0 == 0: --- True

3 return y

return 89

3. gcd(25, 0) = 25

1 in function gcd(x=25, y=0)

2 if 25 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 25

4. hex(255) = 'FF'

3 in function hex(number=255)

4 if 255 == 0: --- False

6 res = ''

res = ''

7 while 255 > 0: --- True

8 digit = number % 16

digit = 255 % 16

digit = 15

9 res = DIGITS[digit] + res

res = DIGITS[15] + ''

res = 'F'

10 number = number // 16

number = 255 // 16

number = 15

7 while 15 > 0: --- True

8 digit = number % 16

digit = 15 % 16

digit = 15

9 res = DIGITS[digit] + res

res = DIGITS[15] + 'F'

res = 'FF'

10 number = number // 16

number = 15 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return FF

5. square\_equal(4, 30, 56) = [-4.0, -3.5]

3 in function square\_equal(a=4, b=30, c=56)

4 if 4 != 0: --- True

5 D = b\*b - 4\*a\*c

D = 30\*30 - 4\*4\*56

D = 4

6 if 4 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (-30 - sqrt(4)) / (2\*4)

x1 = -4.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (-30 + sqrt(4)) / (2\*4)

x2 = -3.5

9 return [x1, x2]

return [-4.0, -3.5]

6. square\_equal(-36, 72, -36) = [1.0]

3 in function square\_equal(a=-36, b=72, c=-36)

4 if -36 != 0: --- True

5 D = b\*b - 4\*a\*c

D = 72\*72 - 4\*-36\*-36

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [1.0]

7. square\_equal(-14, 15, -11) = []

3 in function square\_equal(a=-14, b=15, c=-11)

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

5 D = b\*b - 4\*a\*c

D = 15\*15 - 4\*-14\*-11

D = -391

6 if -391 > 0: --- False

10 elif -391 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, 25, 79) = [-3.16]

3 in function square\_equal(a=0, b=25, c=79)

4 if 0 != 0: --- False

14 else:

15 if 25 != 0: --- True

16 return [-c / b]

return [-3.16]

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

3 in function square\_equal(a=0, b=0, c=-46)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-19

1. gcd(32, 76) = 4

1 in function gcd(x=32, y=76)

2 if 32 == 0: --- False

4 if 76 == 0: --- False

6 while 76 != 0: --- True

7 rem = x % y

rem = 32 % 76

rem = 32

8 x = y

x = 76

9 y = rem

y = 32

6 while 32 != 0: --- True

7 rem = x % y

rem = 76 % 32

rem = 12

8 x = y

x = 32

9 y = rem

y = 12

6 while 12 != 0: --- True

7 rem = x % y

rem = 32 % 12

rem = 8

8 x = y

x = 12

9 y = rem

y = 8

6 while 8 != 0: --- True

7 rem = x % y

rem = 12 % 8

rem = 4

8 x = y

x = 8

9 y = rem

y = 4

6 while 4 != 0: --- True

7 rem = x % y

rem = 8 % 4

rem = 0

8 x = y

x = 4

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 4

2. gcd(0, 95) = 95

1 in function gcd(x=0, y=95)

2 if 0 == 0: --- True

3 return y

return 95

3. gcd(69, 0) = 69

1 in function gcd(x=69, y=0)

2 if 69 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 69

4. hex(249) = 'F9'

3 in function hex(number=249)

4 if 249 == 0: --- False

6 res = ''

res = ''

7 while 249 > 0: --- True

8 digit = number % 16

digit = 249 % 16

digit = 9

9 res = DIGITS[digit] + res

res = DIGITS[9] + ''

res = '9'

10 number = number // 16

number = 249 // 16

number = 15

7 while 15 > 0: --- True

8 digit = number % 16

digit = 15 % 16

digit = 15

9 res = DIGITS[digit] + res

res = DIGITS[15] + '9'

res = 'F9'

10 number = number // 16

number = 15 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return F9

5. square\_equal(-1, -25, 26) = [1.0, -26.0]

3 in function square\_equal(a=-1, b=-25, c=26)

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

5 D = b\*b - 4\*a\*c

D = -25\*-25 - 4\*-1\*26

D = 729

6 if 729 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (--25 - sqrt(729)) / (2\*-1)

x1 = 1.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (--25 + sqrt(729)) / (2\*-1)

x2 = -26.0

9 return [x1, x2]

return [1.0, -26.0]

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

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

4 if 25 != 0: --- True

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [-1.0]

7. square\_equal(10, -7, 13) = []

3 in function square\_equal(a=10, b=-7, c=13)

4 if 10 != 0: --- True

5 D = b\*b - 4\*a\*c

D = -7\*-7 - 4\*10\*13

D = -471

6 if -471 > 0: --- False

10 elif -471 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, 4, -6) = [1.5]

3 in function square\_equal(a=0, b=4, c=-6)

4 if 0 != 0: --- False

14 else:

15 if 4 != 0: --- True

16 return [-c / b]

return [1.5]

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

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

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-20

1. gcd(55, 100) = 5

1 in function gcd(x=55, y=100)

2 if 55 == 0: --- False

4 if 100 == 0: --- False

6 while 100 != 0: --- True

7 rem = x % y

rem = 55 % 100

rem = 55

8 x = y

x = 100

9 y = rem

y = 55

6 while 55 != 0: --- True

7 rem = x % y

rem = 100 % 55

rem = 45

8 x = y

x = 55

9 y = rem

y = 45

6 while 45 != 0: --- True

7 rem = x % y

rem = 55 % 45

rem = 10

8 x = y

x = 45

9 y = rem

y = 10

6 while 10 != 0: --- True

7 rem = x % y

rem = 45 % 10

rem = 5

8 x = y

x = 10

9 y = rem

y = 5

6 while 5 != 0: --- True

7 rem = x % y

rem = 10 % 5

rem = 0

8 x = y

x = 5

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 5

2. gcd(0, 20) = 20

1 in function gcd(x=0, y=20)

2 if 0 == 0: --- True

3 return y

return 20

3. gcd(70, 0) = 70

1 in function gcd(x=70, y=0)

2 if 70 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 70

4. hex(216) = 'D8'

3 in function hex(number=216)

4 if 216 == 0: --- False

6 res = ''

res = ''

7 while 216 > 0: --- True

8 digit = number % 16

digit = 216 % 16

digit = 8

9 res = DIGITS[digit] + res

res = DIGITS[8] + ''

res = '8'

10 number = number // 16

number = 216 // 16

number = 13

7 while 13 > 0: --- True

8 digit = number % 16

digit = 13 % 16

digit = 13

9 res = DIGITS[digit] + res

res = DIGITS[13] + '8'

res = 'D8'

10 number = number // 16

number = 13 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return D8

5. square\_equal(-25, -30, 7) = [0.2, -1.4]

3 in function square\_equal(a=-25, b=-30, c=7)

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

5 D = b\*b - 4\*a\*c

D = -30\*-30 - 4\*-25\*7

D = 1600

6 if 1600 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (--30 - sqrt(1600)) / (2\*-25)

x1 = 0.2

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (--30 + sqrt(1600)) / (2\*-25)

x2 = -1.4

9 return [x1, x2]

return [0.2, -1.4]

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

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

4 if 4 != 0: --- True

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [2.5]

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

3 in function square\_equal(a=5, b=-7, c=3)

4 if 5 != 0: --- True

5 D = b\*b - 4\*a\*c

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

D = -11

6 if -11 > 0: --- False

10 elif -11 == 0: --- False

12 else:

13 return []

return []

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

3 in function square\_equal(a=0, b=4, c=83)

4 if 0 != 0: --- False

14 else:

15 if 4 != 0: --- True

16 return [-c / b]

return [-20.75]

9. square\_equal(0, 0, -84) = []

3 in function square\_equal(a=0, b=0, c=-84)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-21

1. gcd(28, 72) = 4

1 in function gcd(x=28, y=72)

2 if 28 == 0: --- False

4 if 72 == 0: --- False

6 while 72 != 0: --- True

7 rem = x % y

rem = 28 % 72

rem = 28

8 x = y

x = 72

9 y = rem

y = 28

6 while 28 != 0: --- True

7 rem = x % y

rem = 72 % 28

rem = 16

8 x = y

x = 28

9 y = rem

y = 16

6 while 16 != 0: --- True

7 rem = x % y

rem = 28 % 16

rem = 12

8 x = y

x = 16

9 y = rem

y = 12

6 while 12 != 0: --- True

7 rem = x % y

rem = 16 % 12

rem = 4

8 x = y

x = 12

9 y = rem

y = 4

6 while 4 != 0: --- True

7 rem = x % y

rem = 12 % 4

rem = 0

8 x = y

x = 4

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 4

2. gcd(0, 23) = 23

1 in function gcd(x=0, y=23)

2 if 0 == 0: --- True

3 return y

return 23

3. gcd(79, 0) = 79

1 in function gcd(x=79, y=0)

2 if 79 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 79

4. hex(170) = 'AA'

3 in function hex(number=170)

4 if 170 == 0: --- False

6 res = ''

res = ''

7 while 170 > 0: --- True

8 digit = number % 16

digit = 170 % 16

digit = 10

9 res = DIGITS[digit] + res

res = DIGITS[10] + ''

res = 'A'

10 number = number // 16

number = 170 // 16

number = 10

7 while 10 > 0: --- True

8 digit = number % 16

digit = 10 % 16

digit = 10

9 res = DIGITS[digit] + res

res = DIGITS[10] + 'A'

res = 'AA'

10 number = number // 16

number = 10 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return AA

5. square\_equal(16, -52, 40) = [1.25, 2.0]

3 in function square\_equal(a=16, b=-52, c=40)

4 if 16 != 0: --- True

5 D = b\*b - 4\*a\*c

D = -52\*-52 - 4\*16\*40

D = 144

6 if 144 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (--52 - sqrt(144)) / (2\*16)

x1 = 1.25

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (--52 + sqrt(144)) / (2\*16)

x2 = 2.0

9 return [x1, x2]

return [1.25, 2.0]

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

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

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

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [-2.0]

7. square\_equal(-7, 3, -31) = []

3 in function square\_equal(a=-7, b=3, c=-31)

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

5 D = b\*b - 4\*a\*c

D = 3\*3 - 4\*-7\*-31

D = -859

6 if -859 > 0: --- False

10 elif -859 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, -4, -14) = [-3.5]

3 in function square\_equal(a=0, b=-4, c=-14)

4 if 0 != 0: --- False

14 else:

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

16 return [-c / b]

return [-3.5]

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

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

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-22

1. gcd(56, 77) = 7

1 in function gcd(x=56, y=77)

2 if 56 == 0: --- False

4 if 77 == 0: --- False

6 while 77 != 0: --- True

7 rem = x % y

rem = 56 % 77

rem = 56

8 x = y

x = 77

9 y = rem

y = 56

6 while 56 != 0: --- True

7 rem = x % y

rem = 77 % 56

rem = 21

8 x = y

x = 56

9 y = rem

y = 21

6 while 21 != 0: --- True

7 rem = x % y

rem = 56 % 21

rem = 14

8 x = y

x = 21

9 y = rem

y = 14

6 while 14 != 0: --- True

7 rem = x % y

rem = 21 % 14

rem = 7

8 x = y

x = 14

9 y = rem

y = 7

6 while 7 != 0: --- True

7 rem = x % y

rem = 14 % 7

rem = 0

8 x = y

x = 7

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 7

2. gcd(0, 47) = 47

1 in function gcd(x=0, y=47)

2 if 0 == 0: --- True

3 return y

return 47

3. gcd(40, 0) = 40

1 in function gcd(x=40, y=0)

2 if 40 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 40

4. hex(189) = 'BD'

3 in function hex(number=189)

4 if 189 == 0: --- False

6 res = ''

res = ''

7 while 189 > 0: --- True

8 digit = number % 16

digit = 189 % 16

digit = 13

9 res = DIGITS[digit] + res

res = DIGITS[13] + ''

res = 'D'

10 number = number // 16

number = 189 // 16

number = 11

7 while 11 > 0: --- True

8 digit = number % 16

digit = 11 % 16

digit = 11

9 res = DIGITS[digit] + res

res = DIGITS[11] + 'D'

res = 'BD'

10 number = number // 16

number = 11 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return BD

5. square\_equal(-1, -8, 20) = [2.0, -10.0]

3 in function square\_equal(a=-1, b=-8, c=20)

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

5 D = b\*b - 4\*a\*c

D = -8\*-8 - 4\*-1\*20

D = 144

6 if 144 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (--8 - sqrt(144)) / (2\*-1)

x1 = 2.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (--8 + sqrt(144)) / (2\*-1)

x2 = -10.0

9 return [x1, x2]

return [2.0, -10.0]

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

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

4 if 1 != 0: --- True

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [7.0]

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

3 in function square\_equal(a=7, b=-13, c=14)

4 if 7 != 0: --- True

5 D = b\*b - 4\*a\*c

D = -13\*-13 - 4\*7\*14

D = -223

6 if -223 > 0: --- False

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

12 else:

13 return []

return []

8. square\_equal(0, -8, -86) = [-10.75]

3 in function square\_equal(a=0, b=-8, c=-86)

4 if 0 != 0: --- False

14 else:

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

16 return [-c / b]

return [-10.75]

9. square\_equal(0, 0, 28) = []

3 in function square\_equal(a=0, b=0, c=28)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-23

1. gcd(45, 70) = 5

1 in function gcd(x=45, y=70)

2 if 45 == 0: --- False

4 if 70 == 0: --- False

6 while 70 != 0: --- True

7 rem = x % y

rem = 45 % 70

rem = 45

8 x = y

x = 70

9 y = rem

y = 45

6 while 45 != 0: --- True

7 rem = x % y

rem = 70 % 45

rem = 25

8 x = y

x = 45

9 y = rem

y = 25

6 while 25 != 0: --- True

7 rem = x % y

rem = 45 % 25

rem = 20

8 x = y

x = 25

9 y = rem

y = 20

6 while 20 != 0: --- True

7 rem = x % y

rem = 25 % 20

rem = 5

8 x = y

x = 20

9 y = rem

y = 5

6 while 5 != 0: --- True

7 rem = x % y

rem = 20 % 5

rem = 0

8 x = y

x = 5

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 5

2. gcd(0, 57) = 57

1 in function gcd(x=0, y=57)

2 if 0 == 0: --- True

3 return y

return 57

3. gcd(56, 0) = 56

1 in function gcd(x=56, y=0)

2 if 56 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 56

4. hex(233) = 'E9'

3 in function hex(number=233)

4 if 233 == 0: --- False

6 res = ''

res = ''

7 while 233 > 0: --- True

8 digit = number % 16

digit = 233 % 16

digit = 9

9 res = DIGITS[digit] + res

res = DIGITS[9] + ''

res = '9'

10 number = number // 16

number = 233 // 16

number = 14

7 while 14 > 0: --- True

8 digit = number % 16

digit = 14 % 16

digit = 14

9 res = DIGITS[digit] + res

res = DIGITS[14] + '9'

res = 'E9'

10 number = number // 16

number = 14 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return E9

5. square\_equal(16, -76, 48) = [0.75, 4.0]

3 in function square\_equal(a=16, b=-76, c=48)

4 if 16 != 0: --- True

5 D = b\*b - 4\*a\*c

D = -76\*-76 - 4\*16\*48

D = 2704

6 if 2704 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (--76 - sqrt(2704)) / (2\*16)

x1 = 0.75

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (--76 + sqrt(2704)) / (2\*16)

x2 = 4.0

9 return [x1, x2]

return [0.75, 4.0]

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

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

4 if 25 != 0: --- True

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [-1.6]

7. square\_equal(-88, 3, -1) = []

3 in function square\_equal(a=-88, b=3, c=-1)

4 if -88 != 0: --- True

5 D = b\*b - 4\*a\*c

D = 3\*3 - 4\*-88\*-1

D = -343

6 if -343 > 0: --- False

10 elif -343 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, -25, -42) = [-1.68]

3 in function square\_equal(a=0, b=-25, c=-42)

4 if 0 != 0: --- False

14 else:

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

16 return [-c / b]

return [-1.68]

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

3 in function square\_equal(a=0, b=0, c=-48)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-24

1. gcd(72, 52) = 4

1 in function gcd(x=72, y=52)

2 if 72 == 0: --- False

4 if 52 == 0: --- False

6 while 52 != 0: --- True

7 rem = x % y

rem = 72 % 52

rem = 20

8 x = y

x = 52

9 y = rem

y = 20

6 while 20 != 0: --- True

7 rem = x % y

rem = 52 % 20

rem = 12

8 x = y

x = 20

9 y = rem

y = 12

6 while 12 != 0: --- True

7 rem = x % y

rem = 20 % 12

rem = 8

8 x = y

x = 12

9 y = rem

y = 8

6 while 8 != 0: --- True

7 rem = x % y

rem = 12 % 8

rem = 4

8 x = y

x = 8

9 y = rem

y = 4

6 while 4 != 0: --- True

7 rem = x % y

rem = 8 % 4

rem = 0

8 x = y

x = 4

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 4

2. gcd(0, 31) = 31

1 in function gcd(x=0, y=31)

2 if 0 == 0: --- True

3 return y

return 31

3. gcd(41, 0) = 41

1 in function gcd(x=41, y=0)

2 if 41 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 41

4. hex(208) = 'D0'

3 in function hex(number=208)

4 if 208 == 0: --- False

6 res = ''

res = ''

7 while 208 > 0: --- True

8 digit = number % 16

digit = 208 % 16

digit = 0

9 res = DIGITS[digit] + res

res = DIGITS[0] + ''

res = '0'

10 number = number // 16

number = 208 // 16

number = 13

7 while 13 > 0: --- True

8 digit = number % 16

digit = 13 % 16

digit = 13

9 res = DIGITS[digit] + res

res = DIGITS[13] + '0'

res = 'D0'

10 number = number // 16

number = 13 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return D0

5. square\_equal(-1, -2, 8) = [2.0, -4.0]

3 in function square\_equal(a=-1, b=-2, c=8)

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

5 D = b\*b - 4\*a\*c

D = -2\*-2 - 4\*-1\*8

D = 36

6 if 36 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (--2 - sqrt(36)) / (2\*-1)

x1 = 2.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (--2 + sqrt(36)) / (2\*-1)

x2 = -4.0

9 return [x1, x2]

return [2.0, -4.0]

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

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

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

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [3.5]

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

3 in function square\_equal(a=-6, b=1, c=-6)

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

5 D = b\*b - 4\*a\*c

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

D = -143

6 if -143 > 0: --- False

10 elif -143 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, -1, -21) = [-21.0]

3 in function square\_equal(a=0, b=-1, c=-21)

4 if 0 != 0: --- False

14 else:

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

16 return [-c / b]

return [-21.0]

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

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

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-25

1. gcd(65, 40) = 5

1 in function gcd(x=65, y=40)

2 if 65 == 0: --- False

4 if 40 == 0: --- False

6 while 40 != 0: --- True

7 rem = x % y

rem = 65 % 40

rem = 25

8 x = y

x = 40

9 y = rem

y = 25

6 while 25 != 0: --- True

7 rem = x % y

rem = 40 % 25

rem = 15

8 x = y

x = 25

9 y = rem

y = 15

6 while 15 != 0: --- True

7 rem = x % y

rem = 25 % 15

rem = 10

8 x = y

x = 15

9 y = rem

y = 10

6 while 10 != 0: --- True

7 rem = x % y

rem = 15 % 10

rem = 5

8 x = y

x = 10

9 y = rem

y = 5

6 while 5 != 0: --- True

7 rem = x % y

rem = 10 % 5

rem = 0

8 x = y

x = 5

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 5

2. gcd(0, 30) = 30

1 in function gcd(x=0, y=30)

2 if 0 == 0: --- True

3 return y

return 30

3. gcd(95, 0) = 95

1 in function gcd(x=95, y=0)

2 if 95 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 95

4. hex(163) = 'A3'

3 in function hex(number=163)

4 if 163 == 0: --- False

6 res = ''

res = ''

7 while 163 > 0: --- True

8 digit = number % 16

digit = 163 % 16

digit = 3

9 res = DIGITS[digit] + res

res = DIGITS[3] + ''

res = '3'

10 number = number // 16

number = 163 // 16

number = 10

7 while 10 > 0: --- True

8 digit = number % 16

digit = 10 % 16

digit = 10

9 res = DIGITS[digit] + res

res = DIGITS[10] + '3'

res = 'A3'

10 number = number // 16

number = 10 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return A3

5. square\_equal(-4, -28, 95) = [2.5, -9.5]

3 in function square\_equal(a=-4, b=-28, c=95)

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

5 D = b\*b - 4\*a\*c

D = -28\*-28 - 4\*-4\*95

D = 2304

6 if 2304 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (--28 - sqrt(2304)) / (2\*-4)

x1 = 2.5

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (--28 + sqrt(2304)) / (2\*-4)

x2 = -9.5

9 return [x1, x2]

return [2.5, -9.5]

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

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

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

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [1.0]

7. square\_equal(16, -3, 12) = []

3 in function square\_equal(a=16, b=-3, c=12)

4 if 16 != 0: --- True

5 D = b\*b - 4\*a\*c

D = -3\*-3 - 4\*16\*12

D = -759

6 if -759 > 0: --- False

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

12 else:

13 return []

return []

8. square\_equal(0, -6, 93) = [15.5]

3 in function square\_equal(a=0, b=-6, c=93)

4 if 0 != 0: --- False

14 else:

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

16 return [-c / b]

return [15.5]

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

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

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-26

1. gcd(40, 55) = 5

1 in function gcd(x=40, y=55)

2 if 40 == 0: --- False

4 if 55 == 0: --- False

6 while 55 != 0: --- True

7 rem = x % y

rem = 40 % 55

rem = 40

8 x = y

x = 55

9 y = rem

y = 40

6 while 40 != 0: --- True

7 rem = x % y

rem = 55 % 40

rem = 15

8 x = y

x = 40

9 y = rem

y = 15

6 while 15 != 0: --- True

7 rem = x % y

rem = 40 % 15

rem = 10

8 x = y

x = 15

9 y = rem

y = 10

6 while 10 != 0: --- True

7 rem = x % y

rem = 15 % 10

rem = 5

8 x = y

x = 10

9 y = rem

y = 5

6 while 5 != 0: --- True

7 rem = x % y

rem = 10 % 5

rem = 0

8 x = y

x = 5

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 5

2. gcd(0, 27) = 27

1 in function gcd(x=0, y=27)

2 if 0 == 0: --- True

3 return y

return 27

3. gcd(38, 0) = 38

1 in function gcd(x=38, y=0)

2 if 38 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 38

4. hex(176) = 'B0'

3 in function hex(number=176)

4 if 176 == 0: --- False

6 res = ''

res = ''

7 while 176 > 0: --- True

8 digit = number % 16

digit = 176 % 16

digit = 0

9 res = DIGITS[digit] + res

res = DIGITS[0] + ''

res = '0'

10 number = number // 16

number = 176 // 16

number = 11

7 while 11 > 0: --- True

8 digit = number % 16

digit = 11 % 16

digit = 11

9 res = DIGITS[digit] + res

res = DIGITS[11] + '0'

res = 'B0'

10 number = number // 16

number = 11 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return B0

5. square\_equal(-16, -52, 30) = [0.5, -3.75]

3 in function square\_equal(a=-16, b=-52, c=30)

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

5 D = b\*b - 4\*a\*c

D = -52\*-52 - 4\*-16\*30

D = 4624

6 if 4624 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (--52 - sqrt(4624)) / (2\*-16)

x1 = 0.5

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (--52 + sqrt(4624)) / (2\*-16)

x2 = -3.75

9 return [x1, x2]

return [0.5, -3.75]

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

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

4 if 1 != 0: --- True

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [2.0]

7. square\_equal(9, -21, 19) = []

3 in function square\_equal(a=9, b=-21, c=19)

4 if 9 != 0: --- True

5 D = b\*b - 4\*a\*c

D = -21\*-21 - 4\*9\*19

D = -243

6 if -243 > 0: --- False

10 elif -243 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, 2, 11) = [-5.5]

3 in function square\_equal(a=0, b=2, c=11)

4 if 0 != 0: --- False

14 else:

15 if 2 != 0: --- True

16 return [-c / b]

return [-5.5]

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

3 in function square\_equal(a=0, b=0, c=-52)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-27

1. gcd(36, 100) = 4

1 in function gcd(x=36, y=100)

2 if 36 == 0: --- False

4 if 100 == 0: --- False

6 while 100 != 0: --- True

7 rem = x % y

rem = 36 % 100

rem = 36

8 x = y

x = 100

9 y = rem

y = 36

6 while 36 != 0: --- True

7 rem = x % y

rem = 100 % 36

rem = 28

8 x = y

x = 36

9 y = rem

y = 28

6 while 28 != 0: --- True

7 rem = x % y

rem = 36 % 28

rem = 8

8 x = y

x = 28

9 y = rem

y = 8

6 while 8 != 0: --- True

7 rem = x % y

rem = 28 % 8

rem = 4

8 x = y

x = 8

9 y = rem

y = 4

6 while 4 != 0: --- True

7 rem = x % y

rem = 8 % 4

rem = 0

8 x = y

x = 4

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 4

2. gcd(0, 25) = 25

1 in function gcd(x=0, y=25)

2 if 0 == 0: --- True

3 return y

return 25

3. gcd(28, 0) = 28

1 in function gcd(x=28, y=0)

2 if 28 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 28

4. hex(221) = 'DD'

3 in function hex(number=221)

4 if 221 == 0: --- False

6 res = ''

res = ''

7 while 221 > 0: --- True

8 digit = number % 16

digit = 221 % 16

digit = 13

9 res = DIGITS[digit] + res

res = DIGITS[13] + ''

res = 'D'

10 number = number // 16

number = 221 // 16

number = 13

7 while 13 > 0: --- True

8 digit = number % 16

digit = 13 % 16

digit = 13

9 res = DIGITS[digit] + res

res = DIGITS[13] + 'D'

res = 'DD'

10 number = number // 16

number = 13 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return DD

5. square\_equal(-4, -36, 40) = [1.0, -10.0]

3 in function square\_equal(a=-4, b=-36, c=40)

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

5 D = b\*b - 4\*a\*c

D = -36\*-36 - 4\*-4\*40

D = 1936

6 if 1936 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (--36 - sqrt(1936)) / (2\*-4)

x1 = 1.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (--36 + sqrt(1936)) / (2\*-4)

x2 = -10.0

9 return [x1, x2]

return [1.0, -10.0]

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

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

4 if 25 != 0: --- True

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [0.6]

7. square\_equal(-62, -24, -3) = []

3 in function square\_equal(a=-62, b=-24, c=-3)

4 if -62 != 0: --- True

5 D = b\*b - 4\*a\*c

D = -24\*-24 - 4\*-62\*-3

D = -168

6 if -168 > 0: --- False

10 elif -168 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, -6, 51) = [8.5]

3 in function square\_equal(a=0, b=-6, c=51)

4 if 0 != 0: --- False

14 else:

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

16 return [-c / b]

return [8.5]

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

3 in function square\_equal(a=0, b=0, c=-57)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-28

1. gcd(49, 77) = 7

1 in function gcd(x=49, y=77)

2 if 49 == 0: --- False

4 if 77 == 0: --- False

6 while 77 != 0: --- True

7 rem = x % y

rem = 49 % 77

rem = 49

8 x = y

x = 77

9 y = rem

y = 49

6 while 49 != 0: --- True

7 rem = x % y

rem = 77 % 49

rem = 28

8 x = y

x = 49

9 y = rem

y = 28

6 while 28 != 0: --- True

7 rem = x % y

rem = 49 % 28

rem = 21

8 x = y

x = 28

9 y = rem

y = 21

6 while 21 != 0: --- True

7 rem = x % y

rem = 28 % 21

rem = 7

8 x = y

x = 21

9 y = rem

y = 7

6 while 7 != 0: --- True

7 rem = x % y

rem = 21 % 7

rem = 0

8 x = y

x = 7

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 7

2. gcd(0, 42) = 42

1 in function gcd(x=0, y=42)

2 if 0 == 0: --- True

3 return y

return 42

3. gcd(74, 0) = 74

1 in function gcd(x=74, y=0)

2 if 74 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 74

4. hex(217) = 'D9'

3 in function hex(number=217)

4 if 217 == 0: --- False

6 res = ''

res = ''

7 while 217 > 0: --- True

8 digit = number % 16

digit = 217 % 16

digit = 9

9 res = DIGITS[digit] + res

res = DIGITS[9] + ''

res = '9'

10 number = number // 16

number = 217 // 16

number = 13

7 while 13 > 0: --- True

8 digit = number % 16

digit = 13 % 16

digit = 13

9 res = DIGITS[digit] + res

res = DIGITS[13] + '9'

res = 'D9'

10 number = number // 16

number = 13 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return D9

5. square\_equal(1, 65, 64) = [-64.0, -1.0]

3 in function square\_equal(a=1, b=65, c=64)

4 if 1 != 0: --- True

5 D = b\*b - 4\*a\*c

D = 65\*65 - 4\*1\*64

D = 3969

6 if 3969 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (-65 - sqrt(3969)) / (2\*1)

x1 = -64.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (-65 + sqrt(3969)) / (2\*1)

x2 = -1.0

9 return [x1, x2]

return [-64.0, -1.0]

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

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

4 if 16 != 0: --- True

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [2.0]

7. square\_equal(12, -18, 10) = []

3 in function square\_equal(a=12, b=-18, c=10)

4 if 12 != 0: --- True

5 D = b\*b - 4\*a\*c

D = -18\*-18 - 4\*12\*10

D = -156

6 if -156 > 0: --- False

10 elif -156 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, 5, 11) = [-2.2]

3 in function square\_equal(a=0, b=5, c=11)

4 if 0 != 0: --- False

14 else:

15 if 5 != 0: --- True

16 return [-c / b]

return [-2.2]

9. square\_equal(0, 0, -93) = []

3 in function square\_equal(a=0, b=0, c=-93)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-29

1. gcd(55, 75) = 5

1 in function gcd(x=55, y=75)

2 if 55 == 0: --- False

4 if 75 == 0: --- False

6 while 75 != 0: --- True

7 rem = x % y

rem = 55 % 75

rem = 55

8 x = y

x = 75

9 y = rem

y = 55

6 while 55 != 0: --- True

7 rem = x % y

rem = 75 % 55

rem = 20

8 x = y

x = 55

9 y = rem

y = 20

6 while 20 != 0: --- True

7 rem = x % y

rem = 55 % 20

rem = 15

8 x = y

x = 20

9 y = rem

y = 15

6 while 15 != 0: --- True

7 rem = x % y

rem = 20 % 15

rem = 5

8 x = y

x = 15

9 y = rem

y = 5

6 while 5 != 0: --- True

7 rem = x % y

rem = 15 % 5

rem = 0

8 x = y

x = 5

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 5

2. gcd(0, 38) = 38

1 in function gcd(x=0, y=38)

2 if 0 == 0: --- True

3 return y

return 38

3. gcd(45, 0) = 45

1 in function gcd(x=45, y=0)

2 if 45 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 45

4. hex(237) = 'ED'

3 in function hex(number=237)

4 if 237 == 0: --- False

6 res = ''

res = ''

7 while 237 > 0: --- True

8 digit = number % 16

digit = 237 % 16

digit = 13

9 res = DIGITS[digit] + res

res = DIGITS[13] + ''

res = 'D'

10 number = number // 16

number = 237 // 16

number = 14

7 while 14 > 0: --- True

8 digit = number % 16

digit = 14 % 16

digit = 14

9 res = DIGITS[digit] + res

res = DIGITS[14] + 'D'

res = 'ED'

10 number = number // 16

number = 14 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return ED

5. square\_equal(1, -15, 50) = [5.0, 10.0]

3 in function square\_equal(a=1, b=-15, c=50)

4 if 1 != 0: --- True

5 D = b\*b - 4\*a\*c

D = -15\*-15 - 4\*1\*50

D = 25

6 if 25 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (--15 - sqrt(25)) / (2\*1)

x1 = 5.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (--15 + sqrt(25)) / (2\*1)

x2 = 10.0

9 return [x1, x2]

return [5.0, 10.0]

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

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

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

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [-1.0]

7. square\_equal(-2, 5, -68) = []

3 in function square\_equal(a=-2, b=5, c=-68)

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

5 D = b\*b - 4\*a\*c

D = 5\*5 - 4\*-2\*-68

D = -519

6 if -519 > 0: --- False

10 elif -519 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, -1, 39) = [39.0]

3 in function square\_equal(a=0, b=-1, c=39)

4 if 0 != 0: --- False

14 else:

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

16 return [-c / b]

return [39.0]

9. square\_equal(0, 0, -27) = []

3 in function square\_equal(a=0, b=0, c=-27)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-30

1. gcd(95, 55) = 5

1 in function gcd(x=95, y=55)

2 if 95 == 0: --- False

4 if 55 == 0: --- False

6 while 55 != 0: --- True

7 rem = x % y

rem = 95 % 55

rem = 40

8 x = y

x = 55

9 y = rem

y = 40

6 while 40 != 0: --- True

7 rem = x % y

rem = 55 % 40

rem = 15

8 x = y

x = 40

9 y = rem

y = 15

6 while 15 != 0: --- True

7 rem = x % y

rem = 40 % 15

rem = 10

8 x = y

x = 15

9 y = rem

y = 10

6 while 10 != 0: --- True

7 rem = x % y

rem = 15 % 10

rem = 5

8 x = y

x = 10

9 y = rem

y = 5

6 while 5 != 0: --- True

7 rem = x % y

rem = 10 % 5

rem = 0

8 x = y

x = 5

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 5

2. gcd(0, 76) = 76

1 in function gcd(x=0, y=76)

2 if 0 == 0: --- True

3 return y

return 76

3. gcd(17, 0) = 17

1 in function gcd(x=17, y=0)

2 if 17 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 17

4. hex(166) = 'A6'

3 in function hex(number=166)

4 if 166 == 0: --- False

6 res = ''

res = ''

7 while 166 > 0: --- True

8 digit = number % 16

digit = 166 % 16

digit = 6

9 res = DIGITS[digit] + res

res = DIGITS[6] + ''

res = '6'

10 number = number // 16

number = 166 // 16

number = 10

7 while 10 > 0: --- True

8 digit = number % 16

digit = 10 % 16

digit = 10

9 res = DIGITS[digit] + res

res = DIGITS[10] + '6'

res = 'A6'

10 number = number // 16

number = 10 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return A6

5. square\_equal(-16, -88, 23) = [0.25, -5.75]

3 in function square\_equal(a=-16, b=-88, c=23)

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

5 D = b\*b - 4\*a\*c

D = -88\*-88 - 4\*-16\*23

D = 9216

6 if 9216 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (--88 - sqrt(9216)) / (2\*-16)

x1 = 0.25

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (--88 + sqrt(9216)) / (2\*-16)

x2 = -5.75

9 return [x1, x2]

return [0.25, -5.75]

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

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

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

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [-2.0]

7. square\_equal(1, -5, 32) = []

3 in function square\_equal(a=1, b=-5, c=32)

4 if 1 != 0: --- True

5 D = b\*b - 4\*a\*c

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

D = -103

6 if -103 > 0: --- False

10 elif -103 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, -1, -11) = [-11.0]

3 in function square\_equal(a=0, b=-1, c=-11)

4 if 0 != 0: --- False

14 else:

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

16 return [-c / b]

return [-11.0]

9. square\_equal(0, 0, -90) = []

3 in function square\_equal(a=0, b=0, c=-90)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-31

1. gcd(56, 68) = 4

1 in function gcd(x=56, y=68)

2 if 56 == 0: --- False

4 if 68 == 0: --- False

6 while 68 != 0: --- True

7 rem = x % y

rem = 56 % 68

rem = 56

8 x = y

x = 68

9 y = rem

y = 56

6 while 56 != 0: --- True

7 rem = x % y

rem = 68 % 56

rem = 12

8 x = y

x = 56

9 y = rem

y = 12

6 while 12 != 0: --- True

7 rem = x % y

rem = 56 % 12

rem = 8

8 x = y

x = 12

9 y = rem

y = 8

6 while 8 != 0: --- True

7 rem = x % y

rem = 12 % 8

rem = 4

8 x = y

x = 8

9 y = rem

y = 4

6 while 4 != 0: --- True

7 rem = x % y

rem = 8 % 4

rem = 0

8 x = y

x = 4

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 4

2. gcd(0, 16) = 16

1 in function gcd(x=0, y=16)

2 if 0 == 0: --- True

3 return y

return 16

3. gcd(9, 0) = 9

1 in function gcd(x=9, y=0)

2 if 9 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 9

4. hex(162) = 'A2'

3 in function hex(number=162)

4 if 162 == 0: --- False

6 res = ''

res = ''

7 while 162 > 0: --- True

8 digit = number % 16

digit = 162 % 16

digit = 2

9 res = DIGITS[digit] + res

res = DIGITS[2] + ''

res = '2'

10 number = number // 16

number = 162 // 16

number = 10

7 while 10 > 0: --- True

8 digit = number % 16

digit = 10 % 16

digit = 10

9 res = DIGITS[digit] + res

res = DIGITS[10] + '2'

res = 'A2'

10 number = number // 16

number = 10 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return A2

5. square\_equal(-16, 40, 24) = [3.0, -0.5]

3 in function square\_equal(a=-16, b=40, c=24)

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

5 D = b\*b - 4\*a\*c

D = 40\*40 - 4\*-16\*24

D = 3136

6 if 3136 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (-40 - sqrt(3136)) / (2\*-16)

x1 = 3.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (-40 + sqrt(3136)) / (2\*-16)

x2 = -0.5

9 return [x1, x2]

return [3.0, -0.5]

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

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

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

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [-0.4]

7. square\_equal(54, 24, 3) = []

3 in function square\_equal(a=54, b=24, c=3)

4 if 54 != 0: --- True

5 D = b\*b - 4\*a\*c

D = 24\*24 - 4\*54\*3

D = -72

6 if -72 > 0: --- False

10 elif -72 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, 2, 75) = [-37.5]

3 in function square\_equal(a=0, b=2, c=75)

4 if 0 != 0: --- False

14 else:

15 if 2 != 0: --- True

16 return [-c / b]

return [-37.5]

9. square\_equal(0, 0, 46) = []

3 in function square\_equal(a=0, b=0, c=46)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-32

1. gcd(56, 96) = 8

1 in function gcd(x=56, y=96)

2 if 56 == 0: --- False

4 if 96 == 0: --- False

6 while 96 != 0: --- True

7 rem = x % y

rem = 56 % 96

rem = 56

8 x = y

x = 96

9 y = rem

y = 56

6 while 56 != 0: --- True

7 rem = x % y

rem = 96 % 56

rem = 40

8 x = y

x = 56

9 y = rem

y = 40

6 while 40 != 0: --- True

7 rem = x % y

rem = 56 % 40

rem = 16

8 x = y

x = 40

9 y = rem

y = 16

6 while 16 != 0: --- True

7 rem = x % y

rem = 40 % 16

rem = 8

8 x = y

x = 16

9 y = rem

y = 8

6 while 8 != 0: --- True

7 rem = x % y

rem = 16 % 8

rem = 0

8 x = y

x = 8

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 8

2. gcd(0, 11) = 11

1 in function gcd(x=0, y=11)

2 if 0 == 0: --- True

3 return y

return 11

3. gcd(54, 0) = 54

1 in function gcd(x=54, y=0)

2 if 54 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 54

4. hex(171) = 'AB'

3 in function hex(number=171)

4 if 171 == 0: --- False

6 res = ''

res = ''

7 while 171 > 0: --- True

8 digit = number % 16

digit = 171 % 16

digit = 11

9 res = DIGITS[digit] + res

res = DIGITS[11] + ''

res = 'B'

10 number = number // 16

number = 171 // 16

number = 10

7 while 10 > 0: --- True

8 digit = number % 16

digit = 10 % 16

digit = 10

9 res = DIGITS[digit] + res

res = DIGITS[10] + 'B'

res = 'AB'

10 number = number // 16

number = 10 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return AB

5. square\_equal(-4, 6, 40) = [4.0, -2.5]

3 in function square\_equal(a=-4, b=6, c=40)

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

5 D = b\*b - 4\*a\*c

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

D = 676

6 if 676 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (-6 - sqrt(676)) / (2\*-4)

x1 = 4.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (-6 + sqrt(676)) / (2\*-4)

x2 = -2.5

9 return [x1, x2]

return [4.0, -2.5]

6. square\_equal(-1, -6, -9) = [-3.0]

3 in function square\_equal(a=-1, b=-6, c=-9)

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

5 D = b\*b - 4\*a\*c

D = -6\*-6 - 4\*-1\*-9

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [-3.0]

7. square\_equal(-3, -11, -21) = []

3 in function square\_equal(a=-3, b=-11, c=-21)

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

5 D = b\*b - 4\*a\*c

D = -11\*-11 - 4\*-3\*-21

D = -131

6 if -131 > 0: --- False

10 elif -131 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, -14, -91) = [-6.5]

3 in function square\_equal(a=0, b=-14, c=-91)

4 if 0 != 0: --- False

14 else:

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

16 return [-c / b]

return [-6.5]

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

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

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-33

1. gcd(52, 32) = 4

1 in function gcd(x=52, y=32)

2 if 52 == 0: --- False

4 if 32 == 0: --- False

6 while 32 != 0: --- True

7 rem = x % y

rem = 52 % 32

rem = 20

8 x = y

x = 32

9 y = rem

y = 20

6 while 20 != 0: --- True

7 rem = x % y

rem = 32 % 20

rem = 12

8 x = y

x = 20

9 y = rem

y = 12

6 while 12 != 0: --- True

7 rem = x % y

rem = 20 % 12

rem = 8

8 x = y

x = 12

9 y = rem

y = 8

6 while 8 != 0: --- True

7 rem = x % y

rem = 12 % 8

rem = 4

8 x = y

x = 8

9 y = rem

y = 4

6 while 4 != 0: --- True

7 rem = x % y

rem = 8 % 4

rem = 0

8 x = y

x = 4

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 4

2. gcd(0, 99) = 99

1 in function gcd(x=0, y=99)

2 if 0 == 0: --- True

3 return y

return 99

3. gcd(57, 0) = 57

1 in function gcd(x=57, y=0)

2 if 57 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 57

4. hex(203) = 'CB'

3 in function hex(number=203)

4 if 203 == 0: --- False

6 res = ''

res = ''

7 while 203 > 0: --- True

8 digit = number % 16

digit = 203 % 16

digit = 11

9 res = DIGITS[digit] + res

res = DIGITS[11] + ''

res = 'B'

10 number = number // 16

number = 203 // 16

number = 12

7 while 12 > 0: --- True

8 digit = number % 16

digit = 12 % 16

digit = 12

9 res = DIGITS[digit] + res

res = DIGITS[12] + 'B'

res = 'CB'

10 number = number // 16

number = 12 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return CB

5. square\_equal(-625, 0, 1) = [0.04, -0.04]

3 in function square\_equal(a=-625, b=0, c=1)

4 if -625 != 0: --- True

5 D = b\*b - 4\*a\*c

D = 0\*0 - 4\*-625\*1

D = 2500

6 if 2500 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (-0 - sqrt(2500)) / (2\*-625)

x1 = 0.04

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (-0 + sqrt(2500)) / (2\*-625)

x2 = -0.04

9 return [x1, x2]

return [0.04, -0.04]

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

3 in function square\_equal(a=-4, b=-36, c=-81)

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

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [-4.5]

7. square\_equal(19, 2, 6) = []

3 in function square\_equal(a=19, b=2, c=6)

4 if 19 != 0: --- True

5 D = b\*b - 4\*a\*c

D = 2\*2 - 4\*19\*6

D = -452

6 if -452 > 0: --- False

10 elif -452 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, -70, -63) = [-0.9]

3 in function square\_equal(a=0, b=-70, c=-63)

4 if 0 != 0: --- False

14 else:

15 if -70 != 0: --- True

16 return [-c / b]

return [-0.9]

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

3 in function square\_equal(a=0, b=0, c=39)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-34

1. gcd(30, 78) = 6

1 in function gcd(x=30, y=78)

2 if 30 == 0: --- False

4 if 78 == 0: --- False

6 while 78 != 0: --- True

7 rem = x % y

rem = 30 % 78

rem = 30

8 x = y

x = 78

9 y = rem

y = 30

6 while 30 != 0: --- True

7 rem = x % y

rem = 78 % 30

rem = 18

8 x = y

x = 30

9 y = rem

y = 18

6 while 18 != 0: --- True

7 rem = x % y

rem = 30 % 18

rem = 12

8 x = y

x = 18

9 y = rem

y = 12

6 while 12 != 0: --- True

7 rem = x % y

rem = 18 % 12

rem = 6

8 x = y

x = 12

9 y = rem

y = 6

6 while 6 != 0: --- True

7 rem = x % y

rem = 12 % 6

rem = 0

8 x = y

x = 6

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 6

2. gcd(0, 19) = 19

1 in function gcd(x=0, y=19)

2 if 0 == 0: --- True

3 return y

return 19

3. gcd(29, 0) = 29

1 in function gcd(x=29, y=0)

2 if 29 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 29

4. hex(240) = 'F0'

3 in function hex(number=240)

4 if 240 == 0: --- False

6 res = ''

res = ''

7 while 240 > 0: --- True

8 digit = number % 16

digit = 240 % 16

digit = 0

9 res = DIGITS[digit] + res

res = DIGITS[0] + ''

res = '0'

10 number = number // 16

number = 240 // 16

number = 15

7 while 15 > 0: --- True

8 digit = number % 16

digit = 15 % 16

digit = 15

9 res = DIGITS[digit] + res

res = DIGITS[15] + '0'

res = 'F0'

10 number = number // 16

number = 15 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return F0

5. square\_equal(-1, -80, 81) = [1.0, -81.0]

3 in function square\_equal(a=-1, b=-80, c=81)

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

5 D = b\*b - 4\*a\*c

D = -80\*-80 - 4\*-1\*81

D = 6724

6 if 6724 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (--80 - sqrt(6724)) / (2\*-1)

x1 = 1.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (--80 + sqrt(6724)) / (2\*-1)

x2 = -81.0

9 return [x1, x2]

return [1.0, -81.0]

6. square\_equal(4, 24, 36) = [-3.0]

3 in function square\_equal(a=4, b=24, c=36)

4 if 4 != 0: --- True

5 D = b\*b - 4\*a\*c

D = 24\*24 - 4\*4\*36

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [-3.0]

7. square\_equal(4, 8, 36) = []

3 in function square\_equal(a=4, b=8, c=36)

4 if 4 != 0: --- True

5 D = b\*b - 4\*a\*c

D = 8\*8 - 4\*4\*36

D = -512

6 if -512 > 0: --- False

10 elif -512 == 0: --- False

12 else:

13 return []

return []

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

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

4 if 0 != 0: --- False

14 else:

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

16 return [-c / b]

return [1.0]

9. square\_equal(0, 0, 13) = []

3 in function square\_equal(a=0, b=0, c=13)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-35

1. gcd(76, 44) = 4

1 in function gcd(x=76, y=44)

2 if 76 == 0: --- False

4 if 44 == 0: --- False

6 while 44 != 0: --- True

7 rem = x % y

rem = 76 % 44

rem = 32

8 x = y

x = 44

9 y = rem

y = 32

6 while 32 != 0: --- True

7 rem = x % y

rem = 44 % 32

rem = 12

8 x = y

x = 32

9 y = rem

y = 12

6 while 12 != 0: --- True

7 rem = x % y

rem = 32 % 12

rem = 8

8 x = y

x = 12

9 y = rem

y = 8

6 while 8 != 0: --- True

7 rem = x % y

rem = 12 % 8

rem = 4

8 x = y

x = 8

9 y = rem

y = 4

6 while 4 != 0: --- True

7 rem = x % y

rem = 8 % 4

rem = 0

8 x = y

x = 4

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 4

2. gcd(0, 18) = 18

1 in function gcd(x=0, y=18)

2 if 0 == 0: --- True

3 return y

return 18

3. gcd(52, 0) = 52

1 in function gcd(x=52, y=0)

2 if 52 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 52

4. hex(175) = 'AF'

3 in function hex(number=175)

4 if 175 == 0: --- False

6 res = ''

res = ''

7 while 175 > 0: --- True

8 digit = number % 16

digit = 175 % 16

digit = 15

9 res = DIGITS[digit] + res

res = DIGITS[15] + ''

res = 'F'

10 number = number // 16

number = 175 // 16

number = 10

7 while 10 > 0: --- True

8 digit = number % 16

digit = 10 % 16

digit = 10

9 res = DIGITS[digit] + res

res = DIGITS[10] + 'F'

res = 'AF'

10 number = number // 16

number = 10 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return AF

5. square\_equal(-16, -60, 16) = [0.25, -4.0]

3 in function square\_equal(a=-16, b=-60, c=16)

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

5 D = b\*b - 4\*a\*c

D = -60\*-60 - 4\*-16\*16

D = 4624

6 if 4624 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (--60 - sqrt(4624)) / (2\*-16)

x1 = 0.25

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (--60 + sqrt(4624)) / (2\*-16)

x2 = -4.0

9 return [x1, x2]

return [0.25, -4.0]

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

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

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

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [0.75]

7. square\_equal(-16, 18, -13) = []

3 in function square\_equal(a=-16, b=18, c=-13)

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

5 D = b\*b - 4\*a\*c

D = 18\*18 - 4\*-16\*-13

D = -508

6 if -508 > 0: --- False

10 elif -508 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, -64, 96) = [1.5]

3 in function square\_equal(a=0, b=-64, c=96)

4 if 0 != 0: --- False

14 else:

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

16 return [-c / b]

return [1.5]

9. square\_equal(0, 0, -76) = []

3 in function square\_equal(a=0, b=0, c=-76)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-36

1. gcd(50, 85) = 5

1 in function gcd(x=50, y=85)

2 if 50 == 0: --- False

4 if 85 == 0: --- False

6 while 85 != 0: --- True

7 rem = x % y

rem = 50 % 85

rem = 50

8 x = y

x = 85

9 y = rem

y = 50

6 while 50 != 0: --- True

7 rem = x % y

rem = 85 % 50

rem = 35

8 x = y

x = 50

9 y = rem

y = 35

6 while 35 != 0: --- True

7 rem = x % y

rem = 50 % 35

rem = 15

8 x = y

x = 35

9 y = rem

y = 15

6 while 15 != 0: --- True

7 rem = x % y

rem = 35 % 15

rem = 5

8 x = y

x = 15

9 y = rem

y = 5

6 while 5 != 0: --- True

7 rem = x % y

rem = 15 % 5

rem = 0

8 x = y

x = 5

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 5

2. gcd(0, 9) = 9

1 in function gcd(x=0, y=9)

2 if 0 == 0: --- True

3 return y

return 9

3. gcd(58, 0) = 58

1 in function gcd(x=58, y=0)

2 if 58 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 58

4. hex(174) = 'AE'

3 in function hex(number=174)

4 if 174 == 0: --- False

6 res = ''

res = ''

7 while 174 > 0: --- True

8 digit = number % 16

digit = 174 % 16

digit = 14

9 res = DIGITS[digit] + res

res = DIGITS[14] + ''

res = 'E'

10 number = number // 16

number = 174 // 16

number = 10

7 while 10 > 0: --- True

8 digit = number % 16

digit = 10 % 16

digit = 10

9 res = DIGITS[digit] + res

res = DIGITS[10] + 'E'

res = 'AE'

10 number = number // 16

number = 10 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return AE

5. square\_equal(1, -18, 17) = [1.0, 17.0]

3 in function square\_equal(a=1, b=-18, c=17)

4 if 1 != 0: --- True

5 D = b\*b - 4\*a\*c

D = -18\*-18 - 4\*1\*17

D = 256

6 if 256 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (--18 - sqrt(256)) / (2\*1)

x1 = 1.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (--18 + sqrt(256)) / (2\*1)

x2 = 17.0

9 return [x1, x2]

return [1.0, 17.0]

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

3 in function square\_equal(a=-16, b=24, c=-9)

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

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [0.75]

7. square\_equal(37, -11, 1) = []

3 in function square\_equal(a=37, b=-11, c=1)

4 if 37 != 0: --- True

5 D = b\*b - 4\*a\*c

D = -11\*-11 - 4\*37\*1

D = -27

6 if -27 > 0: --- False

10 elif -27 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, 15, 39) = [-2.6]

3 in function square\_equal(a=0, b=15, c=39)

4 if 0 != 0: --- False

14 else:

15 if 15 != 0: --- True

16 return [-c / b]

return [-2.6]

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

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

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-37

1. gcd(44, 60) = 4

1 in function gcd(x=44, y=60)

2 if 44 == 0: --- False

4 if 60 == 0: --- False

6 while 60 != 0: --- True

7 rem = x % y

rem = 44 % 60

rem = 44

8 x = y

x = 60

9 y = rem

y = 44

6 while 44 != 0: --- True

7 rem = x % y

rem = 60 % 44

rem = 16

8 x = y

x = 44

9 y = rem

y = 16

6 while 16 != 0: --- True

7 rem = x % y

rem = 44 % 16

rem = 12

8 x = y

x = 16

9 y = rem

y = 12

6 while 12 != 0: --- True

7 rem = x % y

rem = 16 % 12

rem = 4

8 x = y

x = 12

9 y = rem

y = 4

6 while 4 != 0: --- True

7 rem = x % y

rem = 12 % 4

rem = 0

8 x = y

x = 4

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 4

2. gcd(0, 74) = 74

1 in function gcd(x=0, y=74)

2 if 0 == 0: --- True

3 return y

return 74

3. gcd(55, 0) = 55

1 in function gcd(x=55, y=0)

2 if 55 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 55

4. hex(213) = 'D5'

3 in function hex(number=213)

4 if 213 == 0: --- False

6 res = ''

res = ''

7 while 213 > 0: --- True

8 digit = number % 16

digit = 213 % 16

digit = 5

9 res = DIGITS[digit] + res

res = DIGITS[5] + ''

res = '5'

10 number = number // 16

number = 213 // 16

number = 13

7 while 13 > 0: --- True

8 digit = number % 16

digit = 13 % 16

digit = 13

9 res = DIGITS[digit] + res

res = DIGITS[13] + '5'

res = 'D5'

10 number = number // 16

number = 13 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return D5

5. square\_equal(-4, -14, 30) = [1.5, -5.0]

3 in function square\_equal(a=-4, b=-14, c=30)

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

5 D = b\*b - 4\*a\*c

D = -14\*-14 - 4\*-4\*30

D = 676

6 if 676 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (--14 - sqrt(676)) / (2\*-4)

x1 = 1.5

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (--14 + sqrt(676)) / (2\*-4)

x2 = -5.0

9 return [x1, x2]

return [1.5, -5.0]

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

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

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

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [0.25]

7. square\_equal(3, -25, 61) = []

3 in function square\_equal(a=3, b=-25, c=61)

4 if 3 != 0: --- True

5 D = b\*b - 4\*a\*c

D = -25\*-25 - 4\*3\*61

D = -107

6 if -107 > 0: --- False

10 elif -107 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, -65, 26) = [0.4]

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

4 if 0 != 0: --- False

14 else:

15 if -65 != 0: --- True

16 return [-c / b]

return [0.4]

9. square\_equal(0, 0, 93) = []

3 in function square\_equal(a=0, b=0, c=93)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-38

1. gcd(28, 44) = 4

1 in function gcd(x=28, y=44)

2 if 28 == 0: --- False

4 if 44 == 0: --- False

6 while 44 != 0: --- True

7 rem = x % y

rem = 28 % 44

rem = 28

8 x = y

x = 44

9 y = rem

y = 28

6 while 28 != 0: --- True

7 rem = x % y

rem = 44 % 28

rem = 16

8 x = y

x = 28

9 y = rem

y = 16

6 while 16 != 0: --- True

7 rem = x % y

rem = 28 % 16

rem = 12

8 x = y

x = 16

9 y = rem

y = 12

6 while 12 != 0: --- True

7 rem = x % y

rem = 16 % 12

rem = 4

8 x = y

x = 12

9 y = rem

y = 4

6 while 4 != 0: --- True

7 rem = x % y

rem = 12 % 4

rem = 0

8 x = y

x = 4

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 4

2. gcd(0, 13) = 13

1 in function gcd(x=0, y=13)

2 if 0 == 0: --- True

3 return y

return 13

3. gcd(80, 0) = 80

1 in function gcd(x=80, y=0)

2 if 80 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 80

4. hex(232) = 'E8'

3 in function hex(number=232)

4 if 232 == 0: --- False

6 res = ''

res = ''

7 while 232 > 0: --- True

8 digit = number % 16

digit = 232 % 16

digit = 8

9 res = DIGITS[digit] + res

res = DIGITS[8] + ''

res = '8'

10 number = number // 16

number = 232 // 16

number = 14

7 while 14 > 0: --- True

8 digit = number % 16

digit = 14 % 16

digit = 14

9 res = DIGITS[digit] + res

res = DIGITS[14] + '8'

res = 'E8'

10 number = number // 16

number = 14 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return E8

5. square\_equal(-1, 1, 72) = [9.0, -8.0]

3 in function square\_equal(a=-1, b=1, c=72)

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

5 D = b\*b - 4\*a\*c

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

D = 289

6 if 289 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (-1 - sqrt(289)) / (2\*-1)

x1 = 9.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (-1 + sqrt(289)) / (2\*-1)

x2 = -8.0

9 return [x1, x2]

return [9.0, -8.0]

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

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

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

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [1.0]

7. square\_equal(-44, 0, -1) = []

3 in function square\_equal(a=-44, b=0, c=-1)

4 if -44 != 0: --- True

5 D = b\*b - 4\*a\*c

D = 0\*0 - 4\*-44\*-1

D = -176

6 if -176 > 0: --- False

10 elif -176 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, 62, 93) = [-1.5]

3 in function square\_equal(a=0, b=62, c=93)

4 if 0 != 0: --- False

14 else:

15 if 62 != 0: --- True

16 return [-c / b]

return [-1.5]

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

3 in function square\_equal(a=0, b=0, c=-86)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-39

1. gcd(76, 92) = 4

1 in function gcd(x=76, y=92)

2 if 76 == 0: --- False

4 if 92 == 0: --- False

6 while 92 != 0: --- True

7 rem = x % y

rem = 76 % 92

rem = 76

8 x = y

x = 92

9 y = rem

y = 76

6 while 76 != 0: --- True

7 rem = x % y

rem = 92 % 76

rem = 16

8 x = y

x = 76

9 y = rem

y = 16

6 while 16 != 0: --- True

7 rem = x % y

rem = 76 % 16

rem = 12

8 x = y

x = 16

9 y = rem

y = 12

6 while 12 != 0: --- True

7 rem = x % y

rem = 16 % 12

rem = 4

8 x = y

x = 12

9 y = rem

y = 4

6 while 4 != 0: --- True

7 rem = x % y

rem = 12 % 4

rem = 0

8 x = y

x = 4

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 4

2. gcd(0, 68) = 68

1 in function gcd(x=0, y=68)

2 if 0 == 0: --- True

3 return y

return 68

3. gcd(83, 0) = 83

1 in function gcd(x=83, y=0)

2 if 83 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 83

4. hex(250) = 'FA'

3 in function hex(number=250)

4 if 250 == 0: --- False

6 res = ''

res = ''

7 while 250 > 0: --- True

8 digit = number % 16

digit = 250 % 16

digit = 10

9 res = DIGITS[digit] + res

res = DIGITS[10] + ''

res = 'A'

10 number = number // 16

number = 250 // 16

number = 15

7 while 15 > 0: --- True

8 digit = number % 16

digit = 15 % 16

digit = 15

9 res = DIGITS[digit] + res

res = DIGITS[15] + 'A'

res = 'FA'

10 number = number // 16

number = 15 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return FA

5. square\_equal(25, 65, 36) = [-1.8, -0.8]

3 in function square\_equal(a=25, b=65, c=36)

4 if 25 != 0: --- True

5 D = b\*b - 4\*a\*c

D = 65\*65 - 4\*25\*36

D = 625

6 if 625 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (-65 - sqrt(625)) / (2\*25)

x1 = -1.8

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (-65 + sqrt(625)) / (2\*25)

x2 = -0.8

9 return [x1, x2]

return [-1.8, -0.8]

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

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

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

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [-8.0]

7. square\_equal(42, -19, 5) = []

3 in function square\_equal(a=42, b=-19, c=5)

4 if 42 != 0: --- True

5 D = b\*b - 4\*a\*c

D = -19\*-19 - 4\*42\*5

D = -479

6 if -479 > 0: --- False

10 elif -479 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, 2, 70) = [-35.0]

3 in function square\_equal(a=0, b=2, c=70)

4 if 0 != 0: --- False

14 else:

15 if 2 != 0: --- True

16 return [-c / b]

return [-35.0]

9. square\_equal(0, 0, -61) = []

3 in function square\_equal(a=0, b=0, c=-61)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-40

1. gcd(44, 68) = 4

1 in function gcd(x=44, y=68)

2 if 44 == 0: --- False

4 if 68 == 0: --- False

6 while 68 != 0: --- True

7 rem = x % y

rem = 44 % 68

rem = 44

8 x = y

x = 68

9 y = rem

y = 44

6 while 44 != 0: --- True

7 rem = x % y

rem = 68 % 44

rem = 24

8 x = y

x = 44

9 y = rem

y = 24

6 while 24 != 0: --- True

7 rem = x % y

rem = 44 % 24

rem = 20

8 x = y

x = 24

9 y = rem

y = 20

6 while 20 != 0: --- True

7 rem = x % y

rem = 24 % 20

rem = 4

8 x = y

x = 20

9 y = rem

y = 4

6 while 4 != 0: --- True

7 rem = x % y

rem = 20 % 4

rem = 0

8 x = y

x = 4

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 4

2. gcd(0, 92) = 92

1 in function gcd(x=0, y=92)

2 if 0 == 0: --- True

3 return y

return 92

3. gcd(33, 0) = 33

1 in function gcd(x=33, y=0)

2 if 33 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 33

4. hex(236) = 'EC'

3 in function hex(number=236)

4 if 236 == 0: --- False

6 res = ''

res = ''

7 while 236 > 0: --- True

8 digit = number % 16

digit = 236 % 16

digit = 12

9 res = DIGITS[digit] + res

res = DIGITS[12] + ''

res = 'C'

10 number = number // 16

number = 236 // 16

number = 14

7 while 14 > 0: --- True

8 digit = number % 16

digit = 14 % 16

digit = 14

9 res = DIGITS[digit] + res

res = DIGITS[14] + 'C'

res = 'EC'

10 number = number // 16

number = 14 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return EC

5. square\_equal(-1, 0, 49) = [7.0, -7.0]

3 in function square\_equal(a=-1, b=0, c=49)

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

5 D = b\*b - 4\*a\*c

D = 0\*0 - 4\*-1\*49

D = 196

6 if 196 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (-0 - sqrt(196)) / (2\*-1)

x1 = 7.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (-0 + sqrt(196)) / (2\*-1)

x2 = -7.0

9 return [x1, x2]

return [7.0, -7.0]

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

3 in function square\_equal(a=-9, b=-54, c=-81)

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

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [-3.0]

7. square\_equal(28, -4, 1) = []

3 in function square\_equal(a=28, b=-4, c=1)

4 if 28 != 0: --- True

5 D = b\*b - 4\*a\*c

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

D = -96

6 if -96 > 0: --- False

10 elif -96 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, 10, -38) = [3.8]

3 in function square\_equal(a=0, b=10, c=-38)

4 if 0 != 0: --- False

14 else:

15 if 10 != 0: --- True

16 return [-c / b]

return [3.8]

9. square\_equal(0, 0, 56) = []

3 in function square\_equal(a=0, b=0, c=56)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-41

1. gcd(68, 96) = 4

1 in function gcd(x=68, y=96)

2 if 68 == 0: --- False

4 if 96 == 0: --- False

6 while 96 != 0: --- True

7 rem = x % y

rem = 68 % 96

rem = 68

8 x = y

x = 96

9 y = rem

y = 68

6 while 68 != 0: --- True

7 rem = x % y

rem = 96 % 68

rem = 28

8 x = y

x = 68

9 y = rem

y = 28

6 while 28 != 0: --- True

7 rem = x % y

rem = 68 % 28

rem = 12

8 x = y

x = 28

9 y = rem

y = 12

6 while 12 != 0: --- True

7 rem = x % y

rem = 28 % 12

rem = 4

8 x = y

x = 12

9 y = rem

y = 4

6 while 4 != 0: --- True

7 rem = x % y

rem = 12 % 4

rem = 0

8 x = y

x = 4

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 4

2. gcd(0, 98) = 98

1 in function gcd(x=0, y=98)

2 if 0 == 0: --- True

3 return y

return 98

3. gcd(44, 0) = 44

1 in function gcd(x=44, y=0)

2 if 44 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 44

4. hex(231) = 'E7'

3 in function hex(number=231)

4 if 231 == 0: --- False

6 res = ''

res = ''

7 while 231 > 0: --- True

8 digit = number % 16

digit = 231 % 16

digit = 7

9 res = DIGITS[digit] + res

res = DIGITS[7] + ''

res = '7'

10 number = number // 16

number = 231 // 16

number = 14

7 while 14 > 0: --- True

8 digit = number % 16

digit = 14 % 16

digit = 14

9 res = DIGITS[digit] + res

res = DIGITS[14] + '7'

res = 'E7'

10 number = number // 16

number = 14 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return E7

5. square\_equal(-1, 62, 63) = [63.0, -1.0]

3 in function square\_equal(a=-1, b=62, c=63)

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

5 D = b\*b - 4\*a\*c

D = 62\*62 - 4\*-1\*63

D = 4096

6 if 4096 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (-62 - sqrt(4096)) / (2\*-1)

x1 = 63.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (-62 + sqrt(4096)) / (2\*-1)

x2 = -1.0

9 return [x1, x2]

return [63.0, -1.0]

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

3 in function square\_equal(a=25, b=30, c=9)

4 if 25 != 0: --- True

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [-0.6]

7. square\_equal(12, 13, 12) = []

3 in function square\_equal(a=12, b=13, c=12)

4 if 12 != 0: --- True

5 D = b\*b - 4\*a\*c

D = 13\*13 - 4\*12\*12

D = -407

6 if -407 > 0: --- False

10 elif -407 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, -25, -93) = [-3.72]

3 in function square\_equal(a=0, b=-25, c=-93)

4 if 0 != 0: --- False

14 else:

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

16 return [-c / b]

return [-3.72]

9. square\_equal(0, 0, 90) = []

3 in function square\_equal(a=0, b=0, c=90)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-42

1. gcd(20, 72) = 4

1 in function gcd(x=20, y=72)

2 if 20 == 0: --- False

4 if 72 == 0: --- False

6 while 72 != 0: --- True

7 rem = x % y

rem = 20 % 72

rem = 20

8 x = y

x = 72

9 y = rem

y = 20

6 while 20 != 0: --- True

7 rem = x % y

rem = 72 % 20

rem = 12

8 x = y

x = 20

9 y = rem

y = 12

6 while 12 != 0: --- True

7 rem = x % y

rem = 20 % 12

rem = 8

8 x = y

x = 12

9 y = rem

y = 8

6 while 8 != 0: --- True

7 rem = x % y

rem = 12 % 8

rem = 4

8 x = y

x = 8

9 y = rem

y = 4

6 while 4 != 0: --- True

7 rem = x % y

rem = 8 % 4

rem = 0

8 x = y

x = 4

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 4

2. gcd(0, 87) = 87

1 in function gcd(x=0, y=87)

2 if 0 == 0: --- True

3 return y

return 87

3. gcd(66, 0) = 66

1 in function gcd(x=66, y=0)

2 if 66 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 66

4. hex(172) = 'AC'

3 in function hex(number=172)

4 if 172 == 0: --- False

6 res = ''

res = ''

7 while 172 > 0: --- True

8 digit = number % 16

digit = 172 % 16

digit = 12

9 res = DIGITS[digit] + res

res = DIGITS[12] + ''

res = 'C'

10 number = number // 16

number = 172 // 16

number = 10

7 while 10 > 0: --- True

8 digit = number % 16

digit = 10 % 16

digit = 10

9 res = DIGITS[digit] + res

res = DIGITS[10] + 'C'

res = 'AC'

10 number = number // 16

number = 10 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return AC

5. square\_equal(4, 38, 84) = [-6.0, -3.5]

3 in function square\_equal(a=4, b=38, c=84)

4 if 4 != 0: --- True

5 D = b\*b - 4\*a\*c

D = 38\*38 - 4\*4\*84

D = 100

6 if 100 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (-38 - sqrt(100)) / (2\*4)

x1 = -6.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (-38 + sqrt(100)) / (2\*4)

x2 = -3.5

9 return [x1, x2]

return [-6.0, -3.5]

6. square\_equal(4, 4, 1) = [-0.5]

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

4 if 4 != 0: --- True

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [-0.5]

7. square\_equal(-2, -10, -87) = []

3 in function square\_equal(a=-2, b=-10, c=-87)

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

5 D = b\*b - 4\*a\*c

D = -10\*-10 - 4\*-2\*-87

D = -596

6 if -596 > 0: --- False

10 elif -596 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, 75, -96) = [1.28]

3 in function square\_equal(a=0, b=75, c=-96)

4 if 0 != 0: --- False

14 else:

15 if 75 != 0: --- True

16 return [-c / b]

return [1.28]

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

3 in function square\_equal(a=0, b=0, c=-74)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-43

1. gcd(60, 85) = 5

1 in function gcd(x=60, y=85)

2 if 60 == 0: --- False

4 if 85 == 0: --- False

6 while 85 != 0: --- True

7 rem = x % y

rem = 60 % 85

rem = 60

8 x = y

x = 85

9 y = rem

y = 60

6 while 60 != 0: --- True

7 rem = x % y

rem = 85 % 60

rem = 25

8 x = y

x = 60

9 y = rem

y = 25

6 while 25 != 0: --- True

7 rem = x % y

rem = 60 % 25

rem = 10

8 x = y

x = 25

9 y = rem

y = 10

6 while 10 != 0: --- True

7 rem = x % y

rem = 25 % 10

rem = 5

8 x = y

x = 10

9 y = rem

y = 5

6 while 5 != 0: --- True

7 rem = x % y

rem = 10 % 5

rem = 0

8 x = y

x = 5

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 5

2. gcd(0, 60) = 60

1 in function gcd(x=0, y=60)

2 if 0 == 0: --- True

3 return y

return 60

3. gcd(77, 0) = 77

1 in function gcd(x=77, y=0)

2 if 77 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 77

4. hex(206) = 'CE'

3 in function hex(number=206)

4 if 206 == 0: --- False

6 res = ''

res = ''

7 while 206 > 0: --- True

8 digit = number % 16

digit = 206 % 16

digit = 14

9 res = DIGITS[digit] + res

res = DIGITS[14] + ''

res = 'E'

10 number = number // 16

number = 206 // 16

number = 12

7 while 12 > 0: --- True

8 digit = number % 16

digit = 12 % 16

digit = 12

9 res = DIGITS[digit] + res

res = DIGITS[12] + 'E'

res = 'CE'

10 number = number // 16

number = 12 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return CE

5. square\_equal(4, -22, 30) = [2.5, 3.0]

3 in function square\_equal(a=4, b=-22, c=30)

4 if 4 != 0: --- True

5 D = b\*b - 4\*a\*c

D = -22\*-22 - 4\*4\*30

D = 4

6 if 4 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

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

x1 = 2.5

8 x2 = (-b + sqrt(D)) / (2\*a)

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

x2 = 3.0

9 return [x1, x2]

return [2.5, 3.0]

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

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

4 if -36 != 0: --- True

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [0.5]

7. square\_equal(-4, -11, -42) = []

3 in function square\_equal(a=-4, b=-11, c=-42)

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

5 D = b\*b - 4\*a\*c

D = -11\*-11 - 4\*-4\*-42

D = -551

6 if -551 > 0: --- False

10 elif -551 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, -75, 51) = [0.68]

3 in function square\_equal(a=0, b=-75, c=51)

4 if 0 != 0: --- False

14 else:

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

16 return [-c / b]

return [0.68]

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

3 in function square\_equal(a=0, b=0, c=97)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-44

1. gcd(90, 55) = 5

1 in function gcd(x=90, y=55)

2 if 90 == 0: --- False

4 if 55 == 0: --- False

6 while 55 != 0: --- True

7 rem = x % y

rem = 90 % 55

rem = 35

8 x = y

x = 55

9 y = rem

y = 35

6 while 35 != 0: --- True

7 rem = x % y

rem = 55 % 35

rem = 20

8 x = y

x = 35

9 y = rem

y = 20

6 while 20 != 0: --- True

7 rem = x % y

rem = 35 % 20

rem = 15

8 x = y

x = 20

9 y = rem

y = 15

6 while 15 != 0: --- True

7 rem = x % y

rem = 20 % 15

rem = 5

8 x = y

x = 15

9 y = rem

y = 5

6 while 5 != 0: --- True

7 rem = x % y

rem = 15 % 5

rem = 0

8 x = y

x = 5

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 5

2. gcd(0, 77) = 77

1 in function gcd(x=0, y=77)

2 if 0 == 0: --- True

3 return y

return 77

3. gcd(76, 0) = 76

1 in function gcd(x=76, y=0)

2 if 76 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 76

4. hex(238) = 'EE'

3 in function hex(number=238)

4 if 238 == 0: --- False

6 res = ''

res = ''

7 while 238 > 0: --- True

8 digit = number % 16

digit = 238 % 16

digit = 14

9 res = DIGITS[digit] + res

res = DIGITS[14] + ''

res = 'E'

10 number = number // 16

number = 238 // 16

number = 14

7 while 14 > 0: --- True

8 digit = number % 16

digit = 14 % 16

digit = 14

9 res = DIGITS[digit] + res

res = DIGITS[14] + 'E'

res = 'EE'

10 number = number // 16

number = 14 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return EE

5. square\_equal(-400, 0, 1) = [0.05, -0.05]

3 in function square\_equal(a=-400, b=0, c=1)

4 if -400 != 0: --- True

5 D = b\*b - 4\*a\*c

D = 0\*0 - 4\*-400\*1

D = 1600

6 if 1600 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (-0 - sqrt(1600)) / (2\*-400)

x1 = 0.05

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (-0 + sqrt(1600)) / (2\*-400)

x2 = -0.05

9 return [x1, x2]

return [0.05, -0.05]

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

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

4 if 4 != 0: --- True

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [3.5]

7. square\_equal(31, -26, 6) = []

3 in function square\_equal(a=31, b=-26, c=6)

4 if 31 != 0: --- True

5 D = b\*b - 4\*a\*c

D = -26\*-26 - 4\*31\*6

D = -68

6 if -68 > 0: --- False

10 elif -68 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, 3, -57) = [19.0]

3 in function square\_equal(a=0, b=3, c=-57)

4 if 0 != 0: --- False

14 else:

15 if 3 != 0: --- True

16 return [-c / b]

return [19.0]

9. square\_equal(0, 0, 55) = []

3 in function square\_equal(a=0, b=0, c=55)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-45

1. gcd(52, 96) = 4

1 in function gcd(x=52, y=96)

2 if 52 == 0: --- False

4 if 96 == 0: --- False

6 while 96 != 0: --- True

7 rem = x % y

rem = 52 % 96

rem = 52

8 x = y

x = 96

9 y = rem

y = 52

6 while 52 != 0: --- True

7 rem = x % y

rem = 96 % 52

rem = 44

8 x = y

x = 52

9 y = rem

y = 44

6 while 44 != 0: --- True

7 rem = x % y

rem = 52 % 44

rem = 8

8 x = y

x = 44

9 y = rem

y = 8

6 while 8 != 0: --- True

7 rem = x % y

rem = 44 % 8

rem = 4

8 x = y

x = 8

9 y = rem

y = 4

6 while 4 != 0: --- True

7 rem = x % y

rem = 8 % 4

rem = 0

8 x = y

x = 4

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 4

2. gcd(0, 61) = 61

1 in function gcd(x=0, y=61)

2 if 0 == 0: --- True

3 return y

return 61

3. gcd(71, 0) = 71

1 in function gcd(x=71, y=0)

2 if 71 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 71

4. hex(167) = 'A7'

3 in function hex(number=167)

4 if 167 == 0: --- False

6 res = ''

res = ''

7 while 167 > 0: --- True

8 digit = number % 16

digit = 167 % 16

digit = 7

9 res = DIGITS[digit] + res

res = DIGITS[7] + ''

res = '7'

10 number = number // 16

number = 167 // 16

number = 10

7 while 10 > 0: --- True

8 digit = number % 16

digit = 10 % 16

digit = 10

9 res = DIGITS[digit] + res

res = DIGITS[10] + '7'

res = 'A7'

10 number = number // 16

number = 10 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return A7

5. square\_equal(-1, 29, 62) = [31.0, -2.0]

3 in function square\_equal(a=-1, b=29, c=62)

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

5 D = b\*b - 4\*a\*c

D = 29\*29 - 4\*-1\*62

D = 1089

6 if 1089 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (-29 - sqrt(1089)) / (2\*-1)

x1 = 31.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (-29 + sqrt(1089)) / (2\*-1)

x2 = -2.0

9 return [x1, x2]

return [31.0, -2.0]

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

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

4 if 49 != 0: --- True

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [-1.0]

7. square\_equal(18, 3, 10) = []

3 in function square\_equal(a=18, b=3, c=10)

4 if 18 != 0: --- True

5 D = b\*b - 4\*a\*c

D = 3\*3 - 4\*18\*10

D = -711

6 if -711 > 0: --- False

10 elif -711 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, 5, -25) = [5.0]

3 in function square\_equal(a=0, b=5, c=-25)

4 if 0 != 0: --- False

14 else:

15 if 5 != 0: --- True

16 return [-c / b]

return [5.0]

9. square\_equal(0, 0, -33) = []

3 in function square\_equal(a=0, b=0, c=-33)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-46

1. gcd(91, 56) = 7

1 in function gcd(x=91, y=56)

2 if 91 == 0: --- False

4 if 56 == 0: --- False

6 while 56 != 0: --- True

7 rem = x % y

rem = 91 % 56

rem = 35

8 x = y

x = 56

9 y = rem

y = 35

6 while 35 != 0: --- True

7 rem = x % y

rem = 56 % 35

rem = 21

8 x = y

x = 35

9 y = rem

y = 21

6 while 21 != 0: --- True

7 rem = x % y

rem = 35 % 21

rem = 14

8 x = y

x = 21

9 y = rem

y = 14

6 while 14 != 0: --- True

7 rem = x % y

rem = 21 % 14

rem = 7

8 x = y

x = 14

9 y = rem

y = 7

6 while 7 != 0: --- True

7 rem = x % y

rem = 14 % 7

rem = 0

8 x = y

x = 7

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 7

2. gcd(0, 85) = 85

1 in function gcd(x=0, y=85)

2 if 0 == 0: --- True

3 return y

return 85

3. gcd(32, 0) = 32

1 in function gcd(x=32, y=0)

2 if 32 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 32

4. hex(227) = 'E3'

3 in function hex(number=227)

4 if 227 == 0: --- False

6 res = ''

res = ''

7 while 227 > 0: --- True

8 digit = number % 16

digit = 227 % 16

digit = 3

9 res = DIGITS[digit] + res

res = DIGITS[3] + ''

res = '3'

10 number = number // 16

number = 227 // 16

number = 14

7 while 14 > 0: --- True

8 digit = number % 16

digit = 14 % 16

digit = 14

9 res = DIGITS[digit] + res

res = DIGITS[14] + '3'

res = 'E3'

10 number = number // 16

number = 14 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return E3

5. square\_equal(4, 82, 78) = [-19.5, -1.0]

3 in function square\_equal(a=4, b=82, c=78)

4 if 4 != 0: --- True

5 D = b\*b - 4\*a\*c

D = 82\*82 - 4\*4\*78

D = 5476

6 if 5476 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (-82 - sqrt(5476)) / (2\*4)

x1 = -19.5

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (-82 + sqrt(5476)) / (2\*4)

x2 = -1.0

9 return [x1, x2]

return [-19.5, -1.0]

6. square\_equal(-1, -8, -16) = [-4.0]

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

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

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [-4.0]

7. square\_equal(-14, 19, -17) = []

3 in function square\_equal(a=-14, b=19, c=-17)

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

5 D = b\*b - 4\*a\*c

D = 19\*19 - 4\*-14\*-17

D = -591

6 if -591 > 0: --- False

10 elif -591 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, -50, 62) = [1.24]

3 in function square\_equal(a=0, b=-50, c=62)

4 if 0 != 0: --- False

14 else:

15 if -50 != 0: --- True

16 return [-c / b]

return [1.24]

9. square\_equal(0, 0, -59) = []

3 in function square\_equal(a=0, b=0, c=-59)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-47

1. gcd(30, 48) = 6

1 in function gcd(x=30, y=48)

2 if 30 == 0: --- False

4 if 48 == 0: --- False

6 while 48 != 0: --- True

7 rem = x % y

rem = 30 % 48

rem = 30

8 x = y

x = 48

9 y = rem

y = 30

6 while 30 != 0: --- True

7 rem = x % y

rem = 48 % 30

rem = 18

8 x = y

x = 30

9 y = rem

y = 18

6 while 18 != 0: --- True

7 rem = x % y

rem = 30 % 18

rem = 12

8 x = y

x = 18

9 y = rem

y = 12

6 while 12 != 0: --- True

7 rem = x % y

rem = 18 % 12

rem = 6

8 x = y

x = 12

9 y = rem

y = 6

6 while 6 != 0: --- True

7 rem = x % y

rem = 12 % 6

rem = 0

8 x = y

x = 6

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 6

2. gcd(0, 22) = 22

1 in function gcd(x=0, y=22)

2 if 0 == 0: --- True

3 return y

return 22

3. gcd(62, 0) = 62

1 in function gcd(x=62, y=0)

2 if 62 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 62

4. hex(165) = 'A5'

3 in function hex(number=165)

4 if 165 == 0: --- False

6 res = ''

res = ''

7 while 165 > 0: --- True

8 digit = number % 16

digit = 165 % 16

digit = 5

9 res = DIGITS[digit] + res

res = DIGITS[5] + ''

res = '5'

10 number = number // 16

number = 165 // 16

number = 10

7 while 10 > 0: --- True

8 digit = number % 16

digit = 10 % 16

digit = 10

9 res = DIGITS[digit] + res

res = DIGITS[10] + '5'

res = 'A5'

10 number = number // 16

number = 10 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return A5

5. square\_equal(-1, 6, 7) = [7.0, -1.0]

3 in function square\_equal(a=-1, b=6, c=7)

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

5 D = b\*b - 4\*a\*c

D = 6\*6 - 4\*-1\*7

D = 64

6 if 64 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (-6 - sqrt(64)) / (2\*-1)

x1 = 7.0

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (-6 + sqrt(64)) / (2\*-1)

x2 = -1.0

9 return [x1, x2]

return [7.0, -1.0]

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

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

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

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [1.0]

7. square\_equal(-28, 9, -4) = []

3 in function square\_equal(a=-28, b=9, c=-4)

4 if -28 != 0: --- True

5 D = b\*b - 4\*a\*c

D = 9\*9 - 4\*-28\*-4

D = -367

6 if -367 > 0: --- False

10 elif -367 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, -25, -10) = [-0.4]

3 in function square\_equal(a=0, b=-25, c=-10)

4 if 0 != 0: --- False

14 else:

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

16 return [-c / b]

return [-0.4]

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

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

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-48

1. gcd(25, 90) = 5

1 in function gcd(x=25, y=90)

2 if 25 == 0: --- False

4 if 90 == 0: --- False

6 while 90 != 0: --- True

7 rem = x % y

rem = 25 % 90

rem = 25

8 x = y

x = 90

9 y = rem

y = 25

6 while 25 != 0: --- True

7 rem = x % y

rem = 90 % 25

rem = 15

8 x = y

x = 25

9 y = rem

y = 15

6 while 15 != 0: --- True

7 rem = x % y

rem = 25 % 15

rem = 10

8 x = y

x = 15

9 y = rem

y = 10

6 while 10 != 0: --- True

7 rem = x % y

rem = 15 % 10

rem = 5

8 x = y

x = 10

9 y = rem

y = 5

6 while 5 != 0: --- True

7 rem = x % y

rem = 10 % 5

rem = 0

8 x = y

x = 5

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 5

2. gcd(0, 7) = 7

1 in function gcd(x=0, y=7)

2 if 0 == 0: --- True

3 return y

return 7

3. gcd(14, 0) = 14

1 in function gcd(x=14, y=0)

2 if 14 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 14

4. hex(243) = 'F3'

3 in function hex(number=243)

4 if 243 == 0: --- False

6 res = ''

res = ''

7 while 243 > 0: --- True

8 digit = number % 16

digit = 243 % 16

digit = 3

9 res = DIGITS[digit] + res

res = DIGITS[3] + ''

res = '3'

10 number = number // 16

number = 243 // 16

number = 15

7 while 15 > 0: --- True

8 digit = number % 16

digit = 15 % 16

digit = 15

9 res = DIGITS[digit] + res

res = DIGITS[15] + '3'

res = 'F3'

10 number = number // 16

number = 15 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return F3

5. square\_equal(-4, -30, 16) = [0.5, -8.0]

3 in function square\_equal(a=-4, b=-30, c=16)

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

5 D = b\*b - 4\*a\*c

D = -30\*-30 - 4\*-4\*16

D = 1156

6 if 1156 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (--30 - sqrt(1156)) / (2\*-4)

x1 = 0.5

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (--30 + sqrt(1156)) / (2\*-4)

x2 = -8.0

9 return [x1, x2]

return [0.5, -8.0]

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

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

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

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [1.5]

7. square\_equal(8, 25, 22) = []

3 in function square\_equal(a=8, b=25, c=22)

4 if 8 != 0: --- True

5 D = b\*b - 4\*a\*c

D = 25\*25 - 4\*8\*22

D = -79

6 if -79 > 0: --- False

10 elif -79 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, 11, -66) = [6.0]

3 in function square\_equal(a=0, b=11, c=-66)

4 if 0 != 0: --- False

14 else:

15 if 11 != 0: --- True

16 return [-c / b]

return [6.0]

9. square\_equal(0, 0, 8) = []

3 in function square\_equal(a=0, b=0, c=8)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []

Вариант: 1-1-49

1. gcd(64, 92) = 4

1 in function gcd(x=64, y=92)

2 if 64 == 0: --- False

4 if 92 == 0: --- False

6 while 92 != 0: --- True

7 rem = x % y

rem = 64 % 92

rem = 64

8 x = y

x = 92

9 y = rem

y = 64

6 while 64 != 0: --- True

7 rem = x % y

rem = 92 % 64

rem = 28

8 x = y

x = 64

9 y = rem

y = 28

6 while 28 != 0: --- True

7 rem = x % y

rem = 64 % 28

rem = 8

8 x = y

x = 28

9 y = rem

y = 8

6 while 8 != 0: --- True

7 rem = x % y

rem = 28 % 8

rem = 4

8 x = y

x = 8

9 y = rem

y = 4

6 while 4 != 0: --- True

7 rem = x % y

rem = 8 % 4

rem = 0

8 x = y

x = 4

9 y = rem

y = 0

6 while 0 != 0: --- False

10 return x

return 4

2. gcd(0, 62) = 62

1 in function gcd(x=0, y=62)

2 if 0 == 0: --- True

3 return y

return 62

3. gcd(90, 0) = 90

1 in function gcd(x=90, y=0)

2 if 90 == 0: --- False

4 if 0 == 0: --- True

5 return x

return 90

4. hex(173) = 'AD'

3 in function hex(number=173)

4 if 173 == 0: --- False

6 res = ''

res = ''

7 while 173 > 0: --- True

8 digit = number % 16

digit = 173 % 16

digit = 13

9 res = DIGITS[digit] + res

res = DIGITS[13] + ''

res = 'D'

10 number = number // 16

number = 173 // 16

number = 10

7 while 10 > 0: --- True

8 digit = number % 16

digit = 10 % 16

digit = 10

9 res = DIGITS[digit] + res

res = DIGITS[10] + 'D'

res = 'AD'

10 number = number // 16

number = 10 // 16

number = 0

7 while 0 > 0: --- False

11 return res

return AD

5. square\_equal(-100, 30, 10) = [0.5, -0.2]

3 in function square\_equal(a=-100, b=30, c=10)

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

5 D = b\*b - 4\*a\*c

D = 30\*30 - 4\*-100\*10

D = 4900

6 if 4900 > 0: --- True

7 x1 = (-b - sqrt(D)) / (2\*a)

x1 = (-30 - sqrt(4900)) / (2\*-100)

x1 = 0.5

8 x2 = (-b + sqrt(D)) / (2\*a)

x2 = (-30 + sqrt(4900)) / (2\*-100)

x2 = -0.2

9 return [x1, x2]

return [0.5, -0.2]

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

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

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

5 D = b\*b - 4\*a\*c

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

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return [-b / (2\*a)]

return [-2.25]

7. square\_equal(16, 19, 13) = []

3 in function square\_equal(a=16, b=19, c=13)

4 if 16 != 0: --- True

5 D = b\*b - 4\*a\*c

D = 19\*19 - 4\*16\*13

D = -471

6 if -471 > 0: --- False

10 elif -471 == 0: --- False

12 else:

13 return []

return []

8. square\_equal(0, 1, -33) = [33.0]

3 in function square\_equal(a=0, b=1, c=-33)

4 if 0 != 0: --- False

14 else:

15 if 1 != 0: --- True

16 return [-c / b]

return [33.0]

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

3 in function square\_equal(a=0, b=0, c=40)

4 if 0 != 0: --- False

14 else:

15 if 0 != 0: --- False

17 else:

18 return []

return []