Вариант: 1-2-1

1. gcd(20, 96) = 4

1 def gcd(x=20, y=96)

2 if 20 < 0: --- False

4 if 96 < 0: --- False

6 while 96 != 0: --- True

7 rem = 20 % 96

rem = 20

8 x = 96

9 y = 20

6 while 20 != 0: --- True

7 rem = 96 % 20

rem = 16

8 x = 20

9 y = 16

6 while 16 != 0: --- True

7 rem = 20 % 16

rem = 4

8 x = 16

9 y = 4

6 while 4 != 0: --- True

7 rem = 16 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, -67) = 67

1 def gcd(x=0, y=-67)

2 if 0 < 0: --- False

4 if -67 < 0: --- True

5 y = --67

y = 67

6 while 67 != 0: --- True

7 rem = 0 % 67

rem = 0

8 x = 67

9 y = 0

6 while 0 != 0: --- False

10 return 67

3. hex(248) = 'F8'

1 def hex(number=248)

2 if 248 == 0: --- False

4 res = ''

5 while 248 > 0: --- True

6 digit = 248 % 16

digit = 8

7 if 8 <= 9: --- True

8 digit = str(8)

digit = '8'

23 res = '8' + ''

res = '8'

24 number = 248 // 16

number = 15

5 while 15 > 0: --- True

6 digit = 15 % 16

digit = 15

7 if 15 <= 9: --- False

9 elif 15 <= 13: --- False

19 elif 15 == 14: --- False

21 else:

22 digit = 'F'

23 res = 'F' + '8'

res = 'F8'

24 number = 15 // 16

number = 0

5 while 0 > 0: --- False

25 return 'F8'

4. square\_equal(-4, -1, 0) = '-0.0 and -0.25'

3 def square\_equal(a=-4, b=-1, c=0)

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

5 D = -1\*-1 - 4\*-4\*0

D = 1

6 if 1 > 0: --- True

7 x1 = (--1 - sqrt(1)) / (2\*-4)

x1 = -0.0

8 x2 = (--1 + sqrt(1)) / (2\*-4)

x2 = -0.25

9 return str(-0.0) + ' and ' + str(-0.25)

return '-0.0 and -0.25'

5. square\_equal(-24, 15, -73) = 'no roots'

3 def square\_equal(a=-24, b=15, c=-73)

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

5 D = 15\*15 - 4\*-24\*-73

D = -6783

6 if -6783 > 0: --- False

10 elif -6783 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(135) = '3\*3\*3\*5'

1 def factorize(n=135)

2 res = ''

3 while 135 > 2 and 135 % 2 == 0: --- False

6 d = 3

7 while 135 > 3: --- True

8 if 135 % 3 == 0: --- True

9 res = '' + str(3) + '\*'

res = '3\*'

10 n = 135 // 3

n = 45

7 while 45 > 3: --- True

8 if 45 % 3 == 0: --- True

9 res = '3\*' + str(3) + '\*'

res = '3\*3\*'

10 n = 45 // 3

n = 15

7 while 15 > 3: --- True

8 if 15 % 3 == 0: --- True

9 res = '3\*3\*' + str(3) + '\*'

res = '3\*3\*3\*'

10 n = 15 // 3

n = 5

7 while 5 > 3: --- True

8 if 5 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 5 > 5: --- False

13 return '3\*3\*3\*' + str(5)

return '3\*3\*3\*5'

7. remove\_digit(806, 6) = 80

1 def remove\_digit(number=806, digit=6)

2 res = 0

3 power = 1

4 while 806 > 0: --- True

5 cur\_digit = 806 % 10

cur\_digit = 6

6 if 6 != 6: --- False

9 number = 806 // 10

number = 80

4 while 80 > 0: --- True

5 cur\_digit = 80 % 10

cur\_digit = 0

6 if 0 != 6: --- True

7 res = 0 + 0 \* 1

res = 0

8 power = 1 \* 10

power = 10

9 number = 80 // 10

number = 8

4 while 8 > 0: --- True

5 cur\_digit = 8 % 10

cur\_digit = 8

6 if 8 != 6: --- True

7 res = 0 + 8 \* 10

res = 80

8 power = 10 \* 10

power = 100

9 number = 8 // 10

number = 0

4 while 0 > 0: --- False

10 return 80

Вариант: 1-2-2

1. gcd(48, 57) = 3

1 def gcd(x=48, y=57)

2 if 48 < 0: --- False

4 if 57 < 0: --- False

6 while 57 != 0: --- True

7 rem = 48 % 57

rem = 48

8 x = 57

9 y = 48

6 while 48 != 0: --- True

7 rem = 57 % 48

rem = 9

8 x = 48

9 y = 9

6 while 9 != 0: --- True

7 rem = 48 % 9

rem = 3

8 x = 9

9 y = 3

6 while 3 != 0: --- True

7 rem = 9 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(36, 0) = 36

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

2 if 36 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 36

3. hex(223) = 'DF'

1 def hex(number=223)

2 if 223 == 0: --- False

4 res = ''

5 while 223 > 0: --- True

6 digit = 223 % 16

digit = 15

7 if 15 <= 9: --- False

9 elif 15 <= 13: --- False

19 elif 15 == 14: --- False

21 else:

22 digit = 'F'

23 res = 'F' + ''

res = 'F'

24 number = 223 // 16

number = 13

5 while 13 > 0: --- True

6 digit = 13 % 16

digit = 13

7 if 13 <= 9: --- False

9 elif 13 <= 13: --- True

10 if 13 <= 11: --- False

15 elif 13 == 12: --- False

17 else:

18 digit = 'D'

23 res = 'D' + 'F'

res = 'DF'

24 number = 13 // 16

number = 0

5 while 0 > 0: --- False

25 return 'DF'

4. square\_equal(0, 16, -76) = '4.75'

3 def square\_equal(a=0, b=16, c=-76)

4 if 0 != 0: --- False

14 else:

15 if 16 != 0: --- True

16 return str(--76 / 16)

return '4.75'

5. square\_equal(39, 10, 11) = 'no roots'

3 def square\_equal(a=39, b=10, c=11)

4 if 39 != 0: --- True

5 D = 10\*10 - 4\*39\*11

D = -1616

6 if -1616 > 0: --- False

10 elif -1616 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(28) = '2\*2\*7'

1 def factorize(n=28)

2 res = ''

3 while 28 > 2 and 28 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 28 // 2

n = 14

3 while 14 > 2 and 14 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 14 // 2

n = 7

3 while 7 > 2 and 7 % 2 == 0: --- False

6 d = 3

7 while 7 > 3: --- True

8 if 7 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 7 > 5: --- True

8 if 7 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 7 > 7: --- False

13 return '2\*2\*' + str(7)

return '2\*2\*7'

7. remove\_digit(531, 3) = 51

1 def remove\_digit(number=531, digit=3)

2 res = 0

3 power = 1

4 while 531 > 0: --- True

5 cur\_digit = 531 % 10

cur\_digit = 1

6 if 1 != 3: --- True

7 res = 0 + 1 \* 1

res = 1

8 power = 1 \* 10

power = 10

9 number = 531 // 10

number = 53

4 while 53 > 0: --- True

5 cur\_digit = 53 % 10

cur\_digit = 3

6 if 3 != 3: --- False

9 number = 53 // 10

number = 5

4 while 5 > 0: --- True

5 cur\_digit = 5 % 10

cur\_digit = 5

6 if 5 != 3: --- True

7 res = 1 + 5 \* 10

res = 51

8 power = 10 \* 10

power = 100

9 number = 5 // 10

number = 0

4 while 0 > 0: --- False

10 return 51

Вариант: 1-2-3

1. gcd(40, 68) = 4

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

2 if 40 < 0: --- False

4 if 68 < 0: --- False

6 while 68 != 0: --- True

7 rem = 40 % 68

rem = 40

8 x = 68

9 y = 40

6 while 40 != 0: --- True

7 rem = 68 % 40

rem = 28

8 x = 40

9 y = 28

6 while 28 != 0: --- True

7 rem = 40 % 28

rem = 12

8 x = 28

9 y = 12

6 while 12 != 0: --- True

7 rem = 28 % 12

rem = 4

8 x = 12

9 y = 4

6 while 4 != 0: --- True

7 rem = 12 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, -8) = 8

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

2 if 0 < 0: --- False

4 if -8 < 0: --- True

5 y = --8

y = 8

6 while 8 != 0: --- True

7 rem = 0 % 8

rem = 0

8 x = 8

9 y = 0

6 while 0 != 0: --- False

10 return 8

3. hex(219) = 'DB'

1 def hex(number=219)

2 if 219 == 0: --- False

4 res = ''

5 while 219 > 0: --- True

6 digit = 219 % 16

digit = 11

7 if 11 <= 9: --- False

9 elif 11 <= 13: --- True

10 if 11 <= 11: --- True

11 if 11 == 10: --- False

13 else:

14 digit = 'B'

23 res = 'B' + ''

res = 'B'

24 number = 219 // 16

number = 13

5 while 13 > 0: --- True

6 digit = 13 % 16

digit = 13

7 if 13 <= 9: --- False

9 elif 13 <= 13: --- True

10 if 13 <= 11: --- False

15 elif 13 == 12: --- False

17 else:

18 digit = 'D'

23 res = 'D' + 'B'

res = 'DB'

24 number = 13 // 16

number = 0

5 while 0 > 0: --- False

25 return 'DB'

4. square\_equal(30, 93, 45) = '-2.5 and -0.6'

3 def square\_equal(a=30, b=93, c=45)

4 if 30 != 0: --- True

5 D = 93\*93 - 4\*30\*45

D = 3249

6 if 3249 > 0: --- True

7 x1 = (-93 - sqrt(3249)) / (2\*30)

x1 = -2.5

8 x2 = (-93 + sqrt(3249)) / (2\*30)

x2 = -0.6

9 return str(-2.5) + ' and ' + str(-0.6)

return '-2.5 and -0.6'

5. square\_equal(-2, 8, -16) = 'no roots'

3 def square\_equal(a=-2, b=8, c=-16)

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

5 D = 8\*8 - 4\*-2\*-16

D = -64

6 if -64 > 0: --- False

10 elif -64 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(5) = '5'

1 def factorize(n=5)

2 res = ''

3 while 5 > 2 and 5 % 2 == 0: --- False

6 d = 3

7 while 5 > 3: --- True

8 if 5 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 5 > 5: --- False

13 return '' + str(5)

return '5'

7. remove\_digit(795, 5) = 79

1 def remove\_digit(number=795, digit=5)

2 res = 0

3 power = 1

4 while 795 > 0: --- True

5 cur\_digit = 795 % 10

cur\_digit = 5

6 if 5 != 5: --- False

9 number = 795 // 10

number = 79

4 while 79 > 0: --- True

5 cur\_digit = 79 % 10

cur\_digit = 9

6 if 9 != 5: --- True

7 res = 0 + 9 \* 1

res = 9

8 power = 1 \* 10

power = 10

9 number = 79 // 10

number = 7

4 while 7 > 0: --- True

5 cur\_digit = 7 % 10

cur\_digit = 7

6 if 7 != 5: --- True

7 res = 9 + 7 \* 10

res = 79

8 power = 10 \* 10

power = 100

9 number = 7 // 10

number = 0

4 while 0 > 0: --- False

10 return 79

Вариант: 1-2-4

1. gcd(18, 30) = 6

1 def gcd(x=18, y=30)

2 if 18 < 0: --- False

4 if 30 < 0: --- False

6 while 30 != 0: --- True

7 rem = 18 % 30

rem = 18

8 x = 30

9 y = 18

6 while 18 != 0: --- True

7 rem = 30 % 18

rem = 12

8 x = 18

9 y = 12

6 while 12 != 0: --- True

7 rem = 18 % 12

rem = 6

8 x = 12

9 y = 6

6 while 6 != 0: --- True

7 rem = 12 % 6

rem = 0

8 x = 6

9 y = 0

6 while 0 != 0: --- False

10 return 6

2. gcd(8, 0) = 8

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

2 if 8 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 8

3. hex(206) = 'CE'

1 def hex(number=206)

2 if 206 == 0: --- False

4 res = ''

5 while 206 > 0: --- True

6 digit = 206 % 16

digit = 14

7 if 14 <= 9: --- False

9 elif 14 <= 13: --- False

19 elif 14 == 14: --- True

20 digit = 'E'

23 res = 'E' + ''

res = 'E'

24 number = 206 // 16

number = 12

5 while 12 > 0: --- True

6 digit = 12 % 16

digit = 12

7 if 12 <= 9: --- False

9 elif 12 <= 13: --- True

10 if 12 <= 11: --- False

15 elif 12 == 12: --- True

16 digit = 'C'

23 res = 'C' + 'E'

res = 'CE'

24 number = 12 // 16

number = 0

5 while 0 > 0: --- False

25 return 'CE'

4. square\_equal(100, -98, 0) = '0.0 and 0.98'

3 def square\_equal(a=100, b=-98, c=0)

4 if 100 != 0: --- True

5 D = -98\*-98 - 4\*100\*0

D = 9604

6 if 9604 > 0: --- True

7 x1 = (--98 - sqrt(9604)) / (2\*100)

x1 = 0.0

8 x2 = (--98 + sqrt(9604)) / (2\*100)

x2 = 0.98

9 return str(0.0) + ' and ' + str(0.98)

return '0.0 and 0.98'

5. square\_equal(-19, -12, -64) = 'no roots'

3 def square\_equal(a=-19, b=-12, c=-64)

4 if -19 != 0: --- True

5 D = -12\*-12 - 4\*-19\*-64

D = -4720

6 if -4720 > 0: --- False

10 elif -4720 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(55) = '5\*11'

1 def factorize(n=55)

2 res = ''

3 while 55 > 2 and 55 % 2 == 0: --- False

6 d = 3

7 while 55 > 3: --- True

8 if 55 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 55 > 5: --- True

8 if 55 % 5 == 0: --- True

9 res = '' + str(5) + '\*'

res = '5\*'

10 n = 55 // 5

n = 11

7 while 11 > 5: --- True

8 if 11 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 11 > 7: --- True

8 if 11 % 7 == 0: --- False

11 else:

12 d = 7 + 2

d = 9

7 while 11 > 9: --- True

8 if 11 % 9 == 0: --- False

11 else:

12 d = 9 + 2

d = 11

7 while 11 > 11: --- False

13 return '5\*' + str(11)

return '5\*11'

7. remove\_digit(293, 3) = 29

1 def remove\_digit(number=293, digit=3)

2 res = 0

3 power = 1

4 while 293 > 0: --- True

5 cur\_digit = 293 % 10

cur\_digit = 3

6 if 3 != 3: --- False

9 number = 293 // 10

number = 29

4 while 29 > 0: --- True

5 cur\_digit = 29 % 10

cur\_digit = 9

6 if 9 != 3: --- True

7 res = 0 + 9 \* 1

res = 9

8 power = 1 \* 10

power = 10

9 number = 29 // 10

number = 2

4 while 2 > 0: --- True

5 cur\_digit = 2 % 10

cur\_digit = 2

6 if 2 != 3: --- True

7 res = 9 + 2 \* 10

res = 29

8 power = 10 \* 10

power = 100

9 number = 2 // 10

number = 0

4 while 0 > 0: --- False

10 return 29

Вариант: 1-2-5

1. gcd(-96, -27) = 3

1 def gcd(x=-96, y=-27)

2 if -96 < 0: --- True

3 x = --96

x = 96

4 if -27 < 0: --- True

5 y = --27

y = 27

6 while 27 != 0: --- True

7 rem = 96 % 27

rem = 15

8 x = 27

9 y = 15

6 while 15 != 0: --- True

7 rem = 27 % 15

rem = 12

8 x = 15

9 y = 12

6 while 12 != 0: --- True

7 rem = 15 % 12

rem = 3

8 x = 12

9 y = 3

6 while 3 != 0: --- True

7 rem = 12 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(0, -79) = 79

1 def gcd(x=0, y=-79)

2 if 0 < 0: --- False

4 if -79 < 0: --- True

5 y = --79

y = 79

6 while 79 != 0: --- True

7 rem = 0 % 79

rem = 0

8 x = 79

9 y = 0

6 while 0 != 0: --- False

10 return 79

3. hex(186) = 'BA'

1 def hex(number=186)

2 if 186 == 0: --- False

4 res = ''

5 while 186 > 0: --- True

6 digit = 186 % 16

digit = 10

7 if 10 <= 9: --- False

9 elif 10 <= 13: --- True

10 if 10 <= 11: --- True

11 if 10 == 10: --- True

12 digit = 'A'

23 res = 'A' + ''

res = 'A'

24 number = 186 // 16

number = 11

5 while 11 > 0: --- True

6 digit = 11 % 16

digit = 11

7 if 11 <= 9: --- False

9 elif 11 <= 13: --- True

10 if 11 <= 11: --- True

11 if 11 == 10: --- False

13 else:

14 digit = 'B'

23 res = 'B' + 'A'

res = 'BA'

24 number = 11 // 16

number = 0

5 while 0 > 0: --- False

25 return 'BA'

4. square\_equal(4, 93, -72) = '-24.0 and 0.75'

3 def square\_equal(a=4, b=93, c=-72)

4 if 4 != 0: --- True

5 D = 93\*93 - 4\*4\*-72

D = 9801

6 if 9801 > 0: --- True

7 x1 = (-93 - sqrt(9801)) / (2\*4)

x1 = -24.0

8 x2 = (-93 + sqrt(9801)) / (2\*4)

x2 = 0.75

9 return str(-24.0) + ' and ' + str(0.75)

return '-24.0 and 0.75'

5. square\_equal(79, -1, 14) = 'no roots'

3 def square\_equal(a=79, b=-1, c=14)

4 if 79 != 0: --- True

5 D = -1\*-1 - 4\*79\*14

D = -4423

6 if -4423 > 0: --- False

10 elif -4423 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(125) = '5\*5\*5'

1 def factorize(n=125)

2 res = ''

3 while 125 > 2 and 125 % 2 == 0: --- False

6 d = 3

7 while 125 > 3: --- True

8 if 125 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 125 > 5: --- True

8 if 125 % 5 == 0: --- True

9 res = '' + str(5) + '\*'

res = '5\*'

10 n = 125 // 5

n = 25

7 while 25 > 5: --- True

8 if 25 % 5 == 0: --- True

9 res = '5\*' + str(5) + '\*'

res = '5\*5\*'

10 n = 25 // 5

n = 5

7 while 5 > 5: --- False

13 return '5\*5\*' + str(5)

return '5\*5\*5'

7. remove\_digit(210, 1) = 20

1 def remove\_digit(number=210, digit=1)

2 res = 0

3 power = 1

4 while 210 > 0: --- True

5 cur\_digit = 210 % 10

cur\_digit = 0

6 if 0 != 1: --- True

7 res = 0 + 0 \* 1

res = 0

8 power = 1 \* 10

power = 10

9 number = 210 // 10

number = 21

4 while 21 > 0: --- True

5 cur\_digit = 21 % 10

cur\_digit = 1

6 if 1 != 1: --- False

9 number = 21 // 10

number = 2

4 while 2 > 0: --- True

5 cur\_digit = 2 % 10

cur\_digit = 2

6 if 2 != 1: --- True

7 res = 0 + 2 \* 10

res = 20

8 power = 10 \* 10

power = 100

9 number = 2 // 10

number = 0

4 while 0 > 0: --- False

10 return 20

Вариант: 1-2-6

1. gcd(16, 92) = 4

1 def gcd(x=16, y=92)

2 if 16 < 0: --- False

4 if 92 < 0: --- False

6 while 92 != 0: --- True

7 rem = 16 % 92

rem = 16

8 x = 92

9 y = 16

6 while 16 != 0: --- True

7 rem = 92 % 16

rem = 12

8 x = 16

9 y = 12

6 while 12 != 0: --- True

7 rem = 16 % 12

rem = 4

8 x = 12

9 y = 4

6 while 4 != 0: --- True

7 rem = 12 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(-50, 0) = 50

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

2 if -50 < 0: --- True

3 x = --50

x = 50

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 50

3. hex(252) = 'FC'

1 def hex(number=252)

2 if 252 == 0: --- False

4 res = ''

5 while 252 > 0: --- True

6 digit = 252 % 16

digit = 12

7 if 12 <= 9: --- False

9 elif 12 <= 13: --- True

10 if 12 <= 11: --- False

15 elif 12 == 12: --- True

16 digit = 'C'

23 res = 'C' + ''

res = 'C'

24 number = 252 // 16

number = 15

5 while 15 > 0: --- True

6 digit = 15 % 16

digit = 15

7 if 15 <= 9: --- False

9 elif 15 <= 13: --- False

19 elif 15 == 14: --- False

21 else:

22 digit = 'F'

23 res = 'F' + 'C'

res = 'FC'

24 number = 15 // 16

number = 0

5 while 0 > 0: --- False

25 return 'FC'

4. square\_equal(-4, -37, 0) = '-0.0 and -9.25'

3 def square\_equal(a=-4, b=-37, c=0)

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

5 D = -37\*-37 - 4\*-4\*0

D = 1369

6 if 1369 > 0: --- True

7 x1 = (--37 - sqrt(1369)) / (2\*-4)

x1 = -0.0

8 x2 = (--37 + sqrt(1369)) / (2\*-4)

x2 = -9.25

9 return str(-0.0) + ' and ' + str(-9.25)

return '-0.0 and -9.25'

5. square\_equal(-34, -40, -53) = 'no roots'

3 def square\_equal(a=-34, b=-40, c=-53)

4 if -34 != 0: --- True

5 D = -40\*-40 - 4\*-34\*-53

D = -5608

6 if -5608 > 0: --- False

10 elif -5608 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(100) = '2\*2\*5\*5'

1 def factorize(n=100)

2 res = ''

3 while 100 > 2 and 100 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 100 // 2

n = 50

3 while 50 > 2 and 50 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 50 // 2

n = 25

3 while 25 > 2 and 25 % 2 == 0: --- False

6 d = 3

7 while 25 > 3: --- True

8 if 25 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 25 > 5: --- True

8 if 25 % 5 == 0: --- True

9 res = '2\*2\*' + str(5) + '\*'

res = '2\*2\*5\*'

10 n = 25 // 5

n = 5

7 while 5 > 5: --- False

13 return '2\*2\*5\*' + str(5)

return '2\*2\*5\*5'

7. remove\_digit(3666, 6) = 3

1 def remove\_digit(number=3666, digit=6)

2 res = 0

3 power = 1

4 while 3666 > 0: --- True

5 cur\_digit = 3666 % 10

cur\_digit = 6

6 if 6 != 6: --- False

9 number = 3666 // 10

number = 366

4 while 366 > 0: --- True

5 cur\_digit = 366 % 10

cur\_digit = 6

6 if 6 != 6: --- False

9 number = 366 // 10

number = 36

4 while 36 > 0: --- True

5 cur\_digit = 36 % 10

cur\_digit = 6

6 if 6 != 6: --- False

9 number = 36 // 10

number = 3

4 while 3 > 0: --- True

5 cur\_digit = 3 % 10

cur\_digit = 3

6 if 3 != 6: --- True

7 res = 0 + 3 \* 1

res = 3

8 power = 1 \* 10

power = 10

9 number = 3 // 10

number = 0

4 while 0 > 0: --- False

10 return 3

Вариант: 1-2-7

1. gcd(52, 20) = 4

1 def gcd(x=52, y=20)

2 if 52 < 0: --- False

4 if 20 < 0: --- False

6 while 20 != 0: --- True

7 rem = 52 % 20

rem = 12

8 x = 20

9 y = 12

6 while 12 != 0: --- True

7 rem = 20 % 12

rem = 8

8 x = 12

9 y = 8

6 while 8 != 0: --- True

7 rem = 12 % 8

rem = 4

8 x = 8

9 y = 4

6 while 4 != 0: --- True

7 rem = 8 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(65, 0) = 65

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

2 if 65 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 65

3. hex(216) = 'D8'

1 def hex(number=216)

2 if 216 == 0: --- False

4 res = ''

5 while 216 > 0: --- True

6 digit = 216 % 16

digit = 8

7 if 8 <= 9: --- True

8 digit = str(8)

digit = '8'

23 res = '8' + ''

res = '8'

24 number = 216 // 16

number = 13

5 while 13 > 0: --- True

6 digit = 13 % 16

digit = 13

7 if 13 <= 9: --- False

9 elif 13 <= 13: --- True

10 if 13 <= 11: --- False

15 elif 13 == 12: --- False

17 else:

18 digit = 'D'

23 res = 'D' + '8'

res = 'D8'

24 number = 13 // 16

number = 0

5 while 0 > 0: --- False

25 return 'D8'

4. square\_equal(25, -2, 0) = '0.0 and 0.08'

3 def square\_equal(a=25, b=-2, c=0)

4 if 25 != 0: --- True

5 D = -2\*-2 - 4\*25\*0

D = 4

6 if 4 > 0: --- True

7 x1 = (--2 - sqrt(4)) / (2\*25)

x1 = 0.0

8 x2 = (--2 + sqrt(4)) / (2\*25)

x2 = 0.08

9 return str(0.0) + ' and ' + str(0.08)

return '0.0 and 0.08'

5. square\_equal(-81, -47, -15) = 'no roots'

3 def square\_equal(a=-81, b=-47, c=-15)

4 if -81 != 0: --- True

5 D = -47\*-47 - 4\*-81\*-15

D = -2651

6 if -2651 > 0: --- False

10 elif -2651 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(3) = '3'

1 def factorize(n=3)

2 res = ''

3 while 3 > 2 and 3 % 2 == 0: --- False

6 d = 3

7 while 3 > 3: --- False

13 return '' + str(3)

return '3'

7. remove\_digit(866, 6) = 8

1 def remove\_digit(number=866, digit=6)

2 res = 0

3 power = 1

4 while 866 > 0: --- True

5 cur\_digit = 866 % 10

cur\_digit = 6

6 if 6 != 6: --- False

9 number = 866 // 10

number = 86

4 while 86 > 0: --- True

5 cur\_digit = 86 % 10

cur\_digit = 6

6 if 6 != 6: --- False

9 number = 86 // 10

number = 8

4 while 8 > 0: --- True

5 cur\_digit = 8 % 10

cur\_digit = 8

6 if 8 != 6: --- True

7 res = 0 + 8 \* 1

res = 8

8 power = 1 \* 10

power = 10

9 number = 8 // 10

number = 0

4 while 0 > 0: --- False

10 return 8

Вариант: 1-2-8

1. gcd(-25, -95) = 5

1 def gcd(x=-25, y=-95)

2 if -25 < 0: --- True

3 x = --25

x = 25

4 if -95 < 0: --- True

5 y = --95

y = 95

6 while 95 != 0: --- True

7 rem = 25 % 95

rem = 25

8 x = 95

9 y = 25

6 while 25 != 0: --- True

7 rem = 95 % 25

rem = 20

8 x = 25

9 y = 20

6 while 20 != 0: --- True

7 rem = 25 % 20

rem = 5

8 x = 20

9 y = 5

6 while 5 != 0: --- True

7 rem = 20 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

2. gcd(-68, 0) = 68

1 def gcd(x=-68, y=0)

2 if -68 < 0: --- True

3 x = --68

x = 68

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 68

3. hex(217) = 'D9'

1 def hex(number=217)

2 if 217 == 0: --- False

4 res = ''

5 while 217 > 0: --- True

6 digit = 217 % 16

digit = 9

7 if 9 <= 9: --- True

8 digit = str(9)

digit = '9'

23 res = '9' + ''

res = '9'

24 number = 217 // 16

number = 13

5 while 13 > 0: --- True

6 digit = 13 % 16

digit = 13

7 if 13 <= 9: --- False

9 elif 13 <= 13: --- True

10 if 13 <= 11: --- False

15 elif 13 == 12: --- False

17 else:

18 digit = 'D'

23 res = 'D' + '9'

res = 'D9'

24 number = 13 // 16

number = 0

5 while 0 > 0: --- False

25 return 'D9'

4. square\_equal(8, -34, -100) = '-2.0 and 6.25'

3 def square\_equal(a=8, b=-34, c=-100)

4 if 8 != 0: --- True

5 D = -34\*-34 - 4\*8\*-100

D = 4356

6 if 4356 > 0: --- True

7 x1 = (--34 - sqrt(4356)) / (2\*8)

x1 = -2.0

8 x2 = (--34 + sqrt(4356)) / (2\*8)

x2 = 6.25

9 return str(-2.0) + ' and ' + str(6.25)

return '-2.0 and 6.25'

5. square\_equal(9, -19, 12) = 'no roots'

3 def square\_equal(a=9, b=-19, c=12)

4 if 9 != 0: --- True

5 D = -19\*-19 - 4\*9\*12

D = -71

6 if -71 > 0: --- False

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

12 else:

13 return 'no roots'

6. factorize(180) = '2\*2\*3\*3\*5'

1 def factorize(n=180)

2 res = ''

3 while 180 > 2 and 180 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 180 // 2

n = 90

3 while 90 > 2 and 90 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 90 // 2

n = 45

3 while 45 > 2 and 45 % 2 == 0: --- False

6 d = 3

7 while 45 > 3: --- True

8 if 45 % 3 == 0: --- True

9 res = '2\*2\*' + str(3) + '\*'

res = '2\*2\*3\*'

10 n = 45 // 3

n = 15

7 while 15 > 3: --- True

8 if 15 % 3 == 0: --- True

9 res = '2\*2\*3\*' + str(3) + '\*'

res = '2\*2\*3\*3\*'

10 n = 15 // 3

n = 5

7 while 5 > 3: --- True

8 if 5 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 5 > 5: --- False

13 return '2\*2\*3\*3\*' + str(5)

return '2\*2\*3\*3\*5'

7. remove\_digit(473, 3) = 47

1 def remove\_digit(number=473, digit=3)

2 res = 0

3 power = 1

4 while 473 > 0: --- True

5 cur\_digit = 473 % 10

cur\_digit = 3

6 if 3 != 3: --- False

9 number = 473 // 10

number = 47

4 while 47 > 0: --- True

5 cur\_digit = 47 % 10

cur\_digit = 7

6 if 7 != 3: --- True

7 res = 0 + 7 \* 1

res = 7

8 power = 1 \* 10

power = 10

9 number = 47 // 10

number = 4

4 while 4 > 0: --- True

5 cur\_digit = 4 % 10

cur\_digit = 4

6 if 4 != 3: --- True

7 res = 7 + 4 \* 10

res = 47

8 power = 10 \* 10

power = 100

9 number = 4 // 10

number = 0

4 while 0 > 0: --- False

10 return 47

Вариант: 1-2-9

1. gcd(-85, -90) = 5

1 def gcd(x=-85, y=-90)

2 if -85 < 0: --- True

3 x = --85

x = 85

4 if -90 < 0: --- True

5 y = --90

y = 90

6 while 90 != 0: --- True

7 rem = 85 % 90

rem = 85

8 x = 90

9 y = 85

6 while 85 != 0: --- True

7 rem = 90 % 85

rem = 5

8 x = 85

9 y = 5

6 while 5 != 0: --- True

7 rem = 85 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

2. gcd(0, -25) = 25

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

2 if 0 < 0: --- False

4 if -25 < 0: --- True

5 y = --25

y = 25

6 while 25 != 0: --- True

7 rem = 0 % 25

rem = 0

8 x = 25

9 y = 0

6 while 0 != 0: --- False

10 return 25

3. hex(194) = 'C2'

1 def hex(number=194)

2 if 194 == 0: --- False

4 res = ''

5 while 194 > 0: --- True

6 digit = 194 % 16

digit = 2

7 if 2 <= 9: --- True

8 digit = str(2)

digit = '2'

23 res = '2' + ''

res = '2'

24 number = 194 // 16

number = 12

5 while 12 > 0: --- True

6 digit = 12 % 16

digit = 12

7 if 12 <= 9: --- False

9 elif 12 <= 13: --- True

10 if 12 <= 11: --- False

15 elif 12 == 12: --- True

16 digit = 'C'

23 res = 'C' + '2'

res = 'C2'

24 number = 12 // 16

number = 0

5 while 0 > 0: --- False

25 return 'C2'

4. square\_equal(-3, 21, 24) = '8.0 and -1.0'

3 def square\_equal(a=-3, b=21, c=24)

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

5 D = 21\*21 - 4\*-3\*24

D = 729

6 if 729 > 0: --- True

7 x1 = (-21 - sqrt(729)) / (2\*-3)

x1 = 8.0

8 x2 = (-21 + sqrt(729)) / (2\*-3)

x2 = -1.0

9 return str(8.0) + ' and ' + str(-1.0)

return '8.0 and -1.0'

5. square\_equal(83, 30, 13) = 'no roots'

3 def square\_equal(a=83, b=30, c=13)

4 if 83 != 0: --- True

5 D = 30\*30 - 4\*83\*13

D = -3416

6 if -3416 > 0: --- False

10 elif -3416 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(32) = '2\*2\*2\*2\*2'

1 def factorize(n=32)

2 res = ''

3 while 32 > 2 and 32 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 32 // 2

n = 16

3 while 16 > 2 and 16 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 16 // 2

n = 8

3 while 8 > 2 and 8 % 2 == 0: --- True

4 res = '2\*2\*' + '2\*'

res = '2\*2\*2\*'

5 n = 8 // 2

n = 4

3 while 4 > 2 and 4 % 2 == 0: --- True

4 res = '2\*2\*2\*' + '2\*'

res = '2\*2\*2\*2\*'

5 n = 4 // 2

n = 2

3 while 2 > 2 and 2 % 2 == 0: --- False

6 d = 3

7 while 2 > 3: --- False

13 return '2\*2\*2\*2\*' + str(2)

return '2\*2\*2\*2\*2'

7. remove\_digit(402, 0) = 42

1 def remove\_digit(number=402, digit=0)

2 res = 0

3 power = 1

4 while 402 > 0: --- True

5 cur\_digit = 402 % 10

cur\_digit = 2

6 if 2 != 0: --- True

7 res = 0 + 2 \* 1

res = 2

8 power = 1 \* 10

power = 10

9 number = 402 // 10

number = 40

4 while 40 > 0: --- True

5 cur\_digit = 40 % 10

cur\_digit = 0

6 if 0 != 0: --- False

9 number = 40 // 10

number = 4

4 while 4 > 0: --- True

5 cur\_digit = 4 % 10

cur\_digit = 4

6 if 4 != 0: --- True

7 res = 2 + 4 \* 10

res = 42

8 power = 10 \* 10

power = 100

9 number = 4 // 10

number = 0

4 while 0 > 0: --- False

10 return 42

Вариант: 1-2-10

1. gcd(-21, 35) = 7

1 def gcd(x=-21, y=35)

2 if -21 < 0: --- True

3 x = --21

x = 21

4 if 35 < 0: --- False

6 while 35 != 0: --- True

7 rem = 21 % 35

rem = 21

8 x = 35

9 y = 21

6 while 21 != 0: --- True

7 rem = 35 % 21

rem = 14

8 x = 21

9 y = 14

6 while 14 != 0: --- True

7 rem = 21 % 14

rem = 7

8 x = 14

9 y = 7

6 while 7 != 0: --- True

7 rem = 14 % 7

rem = 0

8 x = 7

9 y = 0

6 while 0 != 0: --- False

10 return 7

2. gcd(56, 0) = 56

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

2 if 56 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 56

3. hex(165) = 'A5'

1 def hex(number=165)

2 if 165 == 0: --- False

4 res = ''

5 while 165 > 0: --- True

6 digit = 165 % 16

digit = 5

7 if 5 <= 9: --- True

8 digit = str(5)

digit = '5'

23 res = '5' + ''

res = '5'

24 number = 165 // 16

number = 10

5 while 10 > 0: --- True

6 digit = 10 % 16

digit = 10

7 if 10 <= 9: --- False

9 elif 10 <= 13: --- True

10 if 10 <= 11: --- True

11 if 10 == 10: --- True

12 digit = 'A'

23 res = 'A' + '5'

res = 'A5'

24 number = 10 // 16

number = 0

5 while 0 > 0: --- False

25 return 'A5'

4. square\_equal(-20, -19, 55) = '1.25 and -2.2'

3 def square\_equal(a=-20, b=-19, c=55)

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

5 D = -19\*-19 - 4\*-20\*55

D = 4761

6 if 4761 > 0: --- True

7 x1 = (--19 - sqrt(4761)) / (2\*-20)

x1 = 1.25

8 x2 = (--19 + sqrt(4761)) / (2\*-20)

x2 = -2.2

9 return str(1.25) + ' and ' + str(-2.2)

return '1.25 and -2.2'

5. square\_equal(56, -21, 40) = 'no roots'

3 def square\_equal(a=56, b=-21, c=40)

4 if 56 != 0: --- True

5 D = -21\*-21 - 4\*56\*40

D = -8519

6 if -8519 > 0: --- False

10 elif -8519 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(30) = '2\*3\*5'

1 def factorize(n=30)

2 res = ''

3 while 30 > 2 and 30 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 30 // 2

n = 15

3 while 15 > 2 and 15 % 2 == 0: --- False

6 d = 3

7 while 15 > 3: --- True

8 if 15 % 3 == 0: --- True

9 res = '2\*' + str(3) + '\*'

res = '2\*3\*'

10 n = 15 // 3

n = 5

7 while 5 > 3: --- True

8 if 5 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 5 > 5: --- False

13 return '2\*3\*' + str(5)

return '2\*3\*5'

7. remove\_digit(73, 3) = 7

1 def remove\_digit(number=73, digit=3)

2 res = 0

3 power = 1

4 while 73 > 0: --- True

5 cur\_digit = 73 % 10

cur\_digit = 3

6 if 3 != 3: --- False

9 number = 73 // 10

number = 7

4 while 7 > 0: --- True

5 cur\_digit = 7 % 10

cur\_digit = 7

6 if 7 != 3: --- True

7 res = 0 + 7 \* 1

res = 7

8 power = 1 \* 10

power = 10

9 number = 7 // 10

number = 0

4 while 0 > 0: --- False

10 return 7

Вариант: 1-2-11

1. gcd(32, 68) = 4

1 def gcd(x=32, y=68)

2 if 32 < 0: --- False

4 if 68 < 0: --- False

6 while 68 != 0: --- True

7 rem = 32 % 68

rem = 32

8 x = 68

9 y = 32

6 while 32 != 0: --- True

7 rem = 68 % 32

rem = 4

8 x = 32

9 y = 4

6 while 4 != 0: --- True

7 rem = 32 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, 84) = 84

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

2 if 0 < 0: --- False

4 if 84 < 0: --- False

6 while 84 != 0: --- True

7 rem = 0 % 84

rem = 0

8 x = 84

9 y = 0

6 while 0 != 0: --- False

10 return 84

3. hex(238) = 'EE'

1 def hex(number=238)

2 if 238 == 0: --- False

4 res = ''

5 while 238 > 0: --- True

6 digit = 238 % 16

digit = 14

7 if 14 <= 9: --- False

9 elif 14 <= 13: --- False

19 elif 14 == 14: --- True

20 digit = 'E'

23 res = 'E' + ''

res = 'E'

24 number = 238 // 16

number = 14

5 while 14 > 0: --- True

6 digit = 14 % 16

digit = 14

7 if 14 <= 9: --- False

9 elif 14 <= 13: --- False

19 elif 14 == 14: --- True

20 digit = 'E'

23 res = 'E' + 'E'

res = 'EE'

24 number = 14 // 16

number = 0

5 while 0 > 0: --- False

25 return 'EE'

4. square\_equal(5, 61, -52) = '-13.0 and 0.8'

3 def square\_equal(a=5, b=61, c=-52)

4 if 5 != 0: --- True

5 D = 61\*61 - 4\*5\*-52

D = 4761

6 if 4761 > 0: --- True

7 x1 = (-61 - sqrt(4761)) / (2\*5)

x1 = -13.0

8 x2 = (-61 + sqrt(4761)) / (2\*5)

x2 = 0.8

9 return str(-13.0) + ' and ' + str(0.8)

return '-13.0 and 0.8'

5. square\_equal(63, -71, 44) = 'no roots'

3 def square\_equal(a=63, b=-71, c=44)

4 if 63 != 0: --- True

5 D = -71\*-71 - 4\*63\*44

D = -6047

6 if -6047 > 0: --- False

10 elif -6047 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(35) = '5\*7'

1 def factorize(n=35)

2 res = ''

3 while 35 > 2 and 35 % 2 == 0: --- False

6 d = 3

7 while 35 > 3: --- True

8 if 35 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 35 > 5: --- True

8 if 35 % 5 == 0: --- True

9 res = '' + str(5) + '\*'

res = '5\*'

10 n = 35 // 5

n = 7

7 while 7 > 5: --- True

8 if 7 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 7 > 7: --- False

13 return '5\*' + str(7)

return '5\*7'

7. remove\_digit(135, 5) = 13

1 def remove\_digit(number=135, digit=5)

2 res = 0

3 power = 1

4 while 135 > 0: --- True

5 cur\_digit = 135 % 10

cur\_digit = 5

6 if 5 != 5: --- False

9 number = 135 // 10

number = 13

4 while 13 > 0: --- True

5 cur\_digit = 13 % 10

cur\_digit = 3

6 if 3 != 5: --- True

7 res = 0 + 3 \* 1

res = 3

8 power = 1 \* 10

power = 10

9 number = 13 // 10

number = 1

4 while 1 > 0: --- True

5 cur\_digit = 1 % 10

cur\_digit = 1

6 if 1 != 5: --- True

7 res = 3 + 1 \* 10

res = 13

8 power = 10 \* 10

power = 100

9 number = 1 // 10

number = 0

4 while 0 > 0: --- False

10 return 13

Вариант: 1-2-12

1. gcd(-44, 66) = 22

1 def gcd(x=-44, y=66)

2 if -44 < 0: --- True

3 x = --44

x = 44

4 if 66 < 0: --- False

6 while 66 != 0: --- True

7 rem = 44 % 66

rem = 44

8 x = 66

9 y = 44

6 while 44 != 0: --- True

7 rem = 66 % 44

rem = 22

8 x = 44

9 y = 22

6 while 22 != 0: --- True

7 rem = 44 % 22

rem = 0

8 x = 22

9 y = 0

6 while 0 != 0: --- False

10 return 22

2. gcd(-21, 0) = 21

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

2 if -21 < 0: --- True

3 x = --21

x = 21

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 21

3. hex(173) = 'AD'

1 def hex(number=173)

2 if 173 == 0: --- False

4 res = ''

5 while 173 > 0: --- True

6 digit = 173 % 16

digit = 13

7 if 13 <= 9: --- False

9 elif 13 <= 13: --- True

10 if 13 <= 11: --- False

15 elif 13 == 12: --- False

17 else:

18 digit = 'D'

23 res = 'D' + ''

res = 'D'

24 number = 173 // 16

number = 10

5 while 10 > 0: --- True

6 digit = 10 % 16

digit = 10

7 if 10 <= 9: --- False

9 elif 10 <= 13: --- True

10 if 10 <= 11: --- True

11 if 10 == 10: --- True

12 digit = 'A'

23 res = 'A' + 'D'

res = 'AD'

24 number = 10 // 16

number = 0

5 while 0 > 0: --- False

25 return 'AD'

4. square\_equal(-14, 21, 14) = '2.0 and -0.5'

3 def square\_equal(a=-14, b=21, c=14)

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

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

D = 1225

6 if 1225 > 0: --- True

7 x1 = (-21 - sqrt(1225)) / (2\*-14)

x1 = 2.0

8 x2 = (-21 + sqrt(1225)) / (2\*-14)

x2 = -0.5

9 return str(2.0) + ' and ' + str(-0.5)

return '2.0 and -0.5'

5. square\_equal(11, -28, 76) = 'no roots'

3 def square\_equal(a=11, b=-28, c=76)

4 if 11 != 0: --- True

5 D = -28\*-28 - 4\*11\*76

D = -2560

6 if -2560 > 0: --- False

10 elif -2560 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(0) = '0'

1 def factorize(n=0)

2 res = ''

3 while 0 > 2 and 0 % 2 == 0: --- False

6 d = 3

7 while 0 > 3: --- False

13 return '' + str(0)

return '0'

7. remove\_digit(660, 0) = 66

1 def remove\_digit(number=660, digit=0)

2 res = 0

3 power = 1

4 while 660 > 0: --- True

5 cur\_digit = 660 % 10

cur\_digit = 0

6 if 0 != 0: --- False

9 number = 660 // 10

number = 66

4 while 66 > 0: --- True

5 cur\_digit = 66 % 10

cur\_digit = 6

6 if 6 != 0: --- True

7 res = 0 + 6 \* 1

res = 6

8 power = 1 \* 10

power = 10

9 number = 66 // 10

number = 6

4 while 6 > 0: --- True

5 cur\_digit = 6 % 10

cur\_digit = 6

6 if 6 != 0: --- True

7 res = 6 + 6 \* 10

res = 66

8 power = 10 \* 10

power = 100

9 number = 6 // 10

number = 0

4 while 0 > 0: --- False

10 return 66

Вариант: 1-2-13

1. gcd(-39, -42) = 3

1 def gcd(x=-39, y=-42)

2 if -39 < 0: --- True

3 x = --39

x = 39

4 if -42 < 0: --- True

5 y = --42

y = 42

6 while 42 != 0: --- True

7 rem = 39 % 42

rem = 39

8 x = 42

9 y = 39

6 while 39 != 0: --- True

7 rem = 42 % 39

rem = 3

8 x = 39

9 y = 3

6 while 3 != 0: --- True

7 rem = 39 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(0, 36) = 36

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

2 if 0 < 0: --- False

4 if 36 < 0: --- False

6 while 36 != 0: --- True

7 rem = 0 % 36

rem = 0

8 x = 36

9 y = 0

6 while 0 != 0: --- False

10 return 36

3. hex(187) = 'BB'

1 def hex(number=187)

2 if 187 == 0: --- False

4 res = ''

5 while 187 > 0: --- True

6 digit = 187 % 16

digit = 11

7 if 11 <= 9: --- False

9 elif 11 <= 13: --- True

10 if 11 <= 11: --- True

11 if 11 == 10: --- False

13 else:

14 digit = 'B'

23 res = 'B' + ''

res = 'B'

24 number = 187 // 16

number = 11

5 while 11 > 0: --- True

6 digit = 11 % 16

digit = 11

7 if 11 <= 9: --- False

9 elif 11 <= 13: --- True

10 if 11 <= 11: --- True

11 if 11 == 10: --- False

13 else:

14 digit = 'B'

23 res = 'B' + 'B'

res = 'BB'

24 number = 11 // 16

number = 0

5 while 0 > 0: --- False

25 return 'BB'

4. square\_equal(0, -1, -11) = '-11.0'

3 def square\_equal(a=0, b=-1, c=-11)

4 if 0 != 0: --- False

14 else:

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

16 return str(--11 / -1)

return '-11.0'

5. square\_equal(71, 7, 27) = 'no roots'

3 def square\_equal(a=71, b=7, c=27)

4 if 71 != 0: --- True

5 D = 7\*7 - 4\*71\*27

D = -7619

6 if -7619 > 0: --- False

10 elif -7619 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(343) = '7\*7\*7'

1 def factorize(n=343)

2 res = ''

3 while 343 > 2 and 343 % 2 == 0: --- False

6 d = 3

7 while 343 > 3: --- True

8 if 343 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 343 > 5: --- True

8 if 343 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 343 > 7: --- True

8 if 343 % 7 == 0: --- True

9 res = '' + str(7) + '\*'

res = '7\*'

10 n = 343 // 7

n = 49

7 while 49 > 7: --- True

8 if 49 % 7 == 0: --- True

9 res = '7\*' + str(7) + '\*'

res = '7\*7\*'

10 n = 49 // 7

n = 7

7 while 7 > 7: --- False

13 return '7\*7\*' + str(7)

return '7\*7\*7'

7. remove\_digit(644, 4) = 6

1 def remove\_digit(number=644, digit=4)

2 res = 0

3 power = 1

4 while 644 > 0: --- True

5 cur\_digit = 644 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 644 // 10

number = 64

4 while 64 > 0: --- True

5 cur\_digit = 64 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 64 // 10

number = 6

4 while 6 > 0: --- True

5 cur\_digit = 6 % 10

cur\_digit = 6

6 if 6 != 4: --- True

7 res = 0 + 6 \* 1

res = 6

8 power = 1 \* 10

power = 10

9 number = 6 // 10

number = 0

4 while 0 > 0: --- False

10 return 6

Вариант: 1-2-14

1. gcd(87, 60) = 3

1 def gcd(x=87, y=60)

2 if 87 < 0: --- False

4 if 60 < 0: --- False

6 while 60 != 0: --- True

7 rem = 87 % 60

rem = 27

8 x = 60

9 y = 27

6 while 27 != 0: --- True

7 rem = 60 % 27

rem = 6

8 x = 27

9 y = 6

6 while 6 != 0: --- True

7 rem = 27 % 6

rem = 3

8 x = 6

9 y = 3

6 while 3 != 0: --- True

7 rem = 6 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(35, 0) = 35

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

2 if 35 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 35

3. hex(224) = 'E0'

1 def hex(number=224)

2 if 224 == 0: --- False

4 res = ''

5 while 224 > 0: --- True

6 digit = 224 % 16

digit = 0

7 if 0 <= 9: --- True

8 digit = str(0)

digit = '0'

23 res = '0' + ''

res = '0'

24 number = 224 // 16

number = 14

5 while 14 > 0: --- True

6 digit = 14 % 16

digit = 14

7 if 14 <= 9: --- False

9 elif 14 <= 13: --- False

19 elif 14 == 14: --- True

20 digit = 'E'

23 res = 'E' + '0'

res = 'E0'

24 number = 14 // 16

number = 0

5 while 0 > 0: --- False

25 return 'E0'

4. square\_equal(0, -35, 21) = '0.6'

3 def square\_equal(a=0, b=-35, c=21)

4 if 0 != 0: --- False

14 else:

15 if -35 != 0: --- True

16 return str(-21 / -35)

return '0.6'

5. square\_equal(52, -68, 35) = 'no roots'

3 def square\_equal(a=52, b=-68, c=35)

4 if 52 != 0: --- True

5 D = -68\*-68 - 4\*52\*35

D = -2656

6 if -2656 > 0: --- False

10 elif -2656 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(294) = '2\*3\*7\*7'

1 def factorize(n=294)

2 res = ''

3 while 294 > 2 and 294 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 294 // 2

n = 147

3 while 147 > 2 and 147 % 2 == 0: --- False

6 d = 3

7 while 147 > 3: --- True

8 if 147 % 3 == 0: --- True

9 res = '2\*' + str(3) + '\*'

res = '2\*3\*'

10 n = 147 // 3

n = 49

7 while 49 > 3: --- True

8 if 49 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 49 > 5: --- True

8 if 49 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 49 > 7: --- True

8 if 49 % 7 == 0: --- True

9 res = '2\*3\*' + str(7) + '\*'

res = '2\*3\*7\*'

10 n = 49 // 7

n = 7

7 while 7 > 7: --- False

13 return '2\*3\*7\*' + str(7)

return '2\*3\*7\*7'

7. remove\_digit(153, 3) = 15

1 def remove\_digit(number=153, digit=3)

2 res = 0

3 power = 1

4 while 153 > 0: --- True

5 cur\_digit = 153 % 10

cur\_digit = 3

6 if 3 != 3: --- False

9 number = 153 // 10

number = 15

4 while 15 > 0: --- True

5 cur\_digit = 15 % 10

cur\_digit = 5

6 if 5 != 3: --- True

7 res = 0 + 5 \* 1

res = 5

8 power = 1 \* 10

power = 10

9 number = 15 // 10

number = 1

4 while 1 > 0: --- True

5 cur\_digit = 1 % 10

cur\_digit = 1

6 if 1 != 3: --- True

7 res = 5 + 1 \* 10

res = 15

8 power = 10 \* 10

power = 100

9 number = 1 // 10

number = 0

4 while 0 > 0: --- False

10 return 15

Вариант: 1-2-15

1. gcd(-72, -57) = 3

1 def gcd(x=-72, y=-57)

2 if -72 < 0: --- True

3 x = --72

x = 72

4 if -57 < 0: --- True

5 y = --57

y = 57

6 while 57 != 0: --- True

7 rem = 72 % 57

rem = 15

8 x = 57

9 y = 15

6 while 15 != 0: --- True

7 rem = 57 % 15

rem = 12

8 x = 15

9 y = 12

6 while 12 != 0: --- True

7 rem = 15 % 12

rem = 3

8 x = 12

9 y = 3

6 while 3 != 0: --- True

7 rem = 12 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(77, 0) = 77

1 def gcd(x=77, y=0)

2 if 77 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 77

3. hex(192) = 'C0'

1 def hex(number=192)

2 if 192 == 0: --- False

4 res = ''

5 while 192 > 0: --- True

6 digit = 192 % 16

digit = 0

7 if 0 <= 9: --- True

8 digit = str(0)

digit = '0'

23 res = '0' + ''

res = '0'

24 number = 192 // 16

number = 12

5 while 12 > 0: --- True

6 digit = 12 % 16

digit = 12

7 if 12 <= 9: --- False

9 elif 12 <= 13: --- True

10 if 12 <= 11: --- False

15 elif 12 == 12: --- True

16 digit = 'C'

23 res = 'C' + '0'

res = 'C0'

24 number = 12 // 16

number = 0

5 while 0 > 0: --- False

25 return 'C0'

4. square\_equal(8, 22, 5) = '-2.5 and -0.25'

3 def square\_equal(a=8, b=22, c=5)

4 if 8 != 0: --- True

5 D = 22\*22 - 4\*8\*5

D = 324

6 if 324 > 0: --- True

7 x1 = (-22 - sqrt(324)) / (2\*8)

x1 = -2.5

8 x2 = (-22 + sqrt(324)) / (2\*8)

x2 = -0.25

9 return str(-2.5) + ' and ' + str(-0.25)

return '-2.5 and -0.25'

5. square\_equal(97, -49, 13) = 'no roots'

3 def square\_equal(a=97, b=-49, c=13)

4 if 97 != 0: --- True

5 D = -49\*-49 - 4\*97\*13

D = -2643

6 if -2643 > 0: --- False

10 elif -2643 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(21) = '3\*7'

1 def factorize(n=21)

2 res = ''

3 while 21 > 2 and 21 % 2 == 0: --- False

6 d = 3

7 while 21 > 3: --- True

8 if 21 % 3 == 0: --- True

9 res = '' + str(3) + '\*'

res = '3\*'

10 n = 21 // 3

n = 7

7 while 7 > 3: --- True

8 if 7 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 7 > 5: --- True

8 if 7 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 7 > 7: --- False

13 return '3\*' + str(7)

return '3\*7'

7. remove\_digit(532, 3) = 52

1 def remove\_digit(number=532, digit=3)

2 res = 0

3 power = 1

4 while 532 > 0: --- True

5 cur\_digit = 532 % 10

cur\_digit = 2

6 if 2 != 3: --- True

7 res = 0 + 2 \* 1

res = 2

8 power = 1 \* 10

power = 10

9 number = 532 // 10

number = 53

4 while 53 > 0: --- True

5 cur\_digit = 53 % 10

cur\_digit = 3

6 if 3 != 3: --- False

9 number = 53 // 10

number = 5

4 while 5 > 0: --- True

5 cur\_digit = 5 % 10

cur\_digit = 5

6 if 5 != 3: --- True

7 res = 2 + 5 \* 10

res = 52

8 power = 10 \* 10

power = 100

9 number = 5 // 10

number = 0

4 while 0 > 0: --- False

10 return 52

Вариант: 1-2-16

1. gcd(75, -40) = 5

1 def gcd(x=75, y=-40)

2 if 75 < 0: --- False

4 if -40 < 0: --- True

5 y = --40

y = 40

6 while 40 != 0: --- True

7 rem = 75 % 40

rem = 35

8 x = 40

9 y = 35

6 while 35 != 0: --- True

7 rem = 40 % 35

rem = 5

8 x = 35

9 y = 5

6 while 5 != 0: --- True

7 rem = 35 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

2. gcd(0, -40) = 40

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

2 if 0 < 0: --- False

4 if -40 < 0: --- True

5 y = --40

y = 40

6 while 40 != 0: --- True

7 rem = 0 % 40

rem = 0

8 x = 40

9 y = 0

6 while 0 != 0: --- False

10 return 40

3. hex(200) = 'C8'

1 def hex(number=200)

2 if 200 == 0: --- False

4 res = ''

5 while 200 > 0: --- True

6 digit = 200 % 16

digit = 8

7 if 8 <= 9: --- True

8 digit = str(8)

digit = '8'

23 res = '8' + ''

res = '8'

24 number = 200 // 16

number = 12

5 while 12 > 0: --- True

6 digit = 12 % 16

digit = 12

7 if 12 <= 9: --- False

9 elif 12 <= 13: --- True

10 if 12 <= 11: --- False

15 elif 12 == 12: --- True

16 digit = 'C'

23 res = 'C' + '8'

res = 'C8'

24 number = 12 // 16

number = 0

5 while 0 > 0: --- False

25 return 'C8'

4. square\_equal(-30, -9, 0) = '-0.0 and -0.3'

3 def square\_equal(a=-30, b=-9, c=0)

4 if -30 != 0: --- True

5 D = -9\*-9 - 4\*-30\*0

D = 81

6 if 81 > 0: --- True

7 x1 = (--9 - sqrt(81)) / (2\*-30)

x1 = -0.0

8 x2 = (--9 + sqrt(81)) / (2\*-30)

x2 = -0.3

9 return str(-0.0) + ' and ' + str(-0.3)

return '-0.0 and -0.3'

5. square\_equal(-8, 42, -67) = 'no roots'

3 def square\_equal(a=-8, b=42, c=-67)

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

5 D = 42\*42 - 4\*-8\*-67

D = -380

6 if -380 > 0: --- False

10 elif -380 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(50) = '2\*5\*5'

1 def factorize(n=50)

2 res = ''

3 while 50 > 2 and 50 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 50 // 2

n = 25

3 while 25 > 2 and 25 % 2 == 0: --- False

6 d = 3

7 while 25 > 3: --- True

8 if 25 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 25 > 5: --- True

8 if 25 % 5 == 0: --- True

9 res = '2\*' + str(5) + '\*'

res = '2\*5\*'

10 n = 25 // 5

n = 5

7 while 5 > 5: --- False

13 return '2\*5\*' + str(5)

return '2\*5\*5'

7. remove\_digit(8222, 2) = 8

1 def remove\_digit(number=8222, digit=2)

2 res = 0

3 power = 1

4 while 8222 > 0: --- True

5 cur\_digit = 8222 % 10

cur\_digit = 2

6 if 2 != 2: --- False

9 number = 8222 // 10

number = 822

4 while 822 > 0: --- True

5 cur\_digit = 822 % 10

cur\_digit = 2

6 if 2 != 2: --- False

9 number = 822 // 10

number = 82

4 while 82 > 0: --- True

5 cur\_digit = 82 % 10

cur\_digit = 2

6 if 2 != 2: --- False

9 number = 82 // 10

number = 8

4 while 8 > 0: --- True

5 cur\_digit = 8 % 10

cur\_digit = 8

6 if 8 != 2: --- True

7 res = 0 + 8 \* 1

res = 8

8 power = 1 \* 10

power = 10

9 number = 8 // 10

number = 0

4 while 0 > 0: --- False

10 return 8

Вариант: 1-2-17

1. gcd(-42, -24) = 6

1 def gcd(x=-42, y=-24)

2 if -42 < 0: --- True

3 x = --42

x = 42

4 if -24 < 0: --- True

5 y = --24

y = 24

6 while 24 != 0: --- True

7 rem = 42 % 24

rem = 18

8 x = 24

9 y = 18

6 while 18 != 0: --- True

7 rem = 24 % 18

rem = 6

8 x = 18

9 y = 6

6 while 6 != 0: --- True

7 rem = 18 % 6

rem = 0

8 x = 6

9 y = 0

6 while 0 != 0: --- False

10 return 6

2. gcd(0, 97) = 97

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

2 if 0 < 0: --- False

4 if 97 < 0: --- False

6 while 97 != 0: --- True

7 rem = 0 % 97

rem = 0

8 x = 97

9 y = 0

6 while 0 != 0: --- False

10 return 97

3. hex(221) = 'DD'

1 def hex(number=221)

2 if 221 == 0: --- False

4 res = ''

5 while 221 > 0: --- True

6 digit = 221 % 16

digit = 13

7 if 13 <= 9: --- False

9 elif 13 <= 13: --- True

10 if 13 <= 11: --- False

15 elif 13 == 12: --- False

17 else:

18 digit = 'D'

23 res = 'D' + ''

res = 'D'

24 number = 221 // 16

number = 13

5 while 13 > 0: --- True

6 digit = 13 % 16

digit = 13

7 if 13 <= 9: --- False

9 elif 13 <= 13: --- True

10 if 13 <= 11: --- False

15 elif 13 == 12: --- False

17 else:

18 digit = 'D'

23 res = 'D' + 'D'

res = 'DD'

24 number = 13 // 16

number = 0

5 while 0 > 0: --- False

25 return 'DD'

4. square\_equal(21, 0, 0) = '0.0'

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

4 if 21 != 0: --- True

5 D = 0\*0 - 4\*21\*0

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return str(-0 / (2\*21))

return '0.0'

5. square\_equal(45, 29, 7) = 'no roots'

3 def square\_equal(a=45, b=29, c=7)

4 if 45 != 0: --- True

5 D = 29\*29 - 4\*45\*7

D = -419

6 if -419 > 0: --- False

10 elif -419 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(13) = '13'

1 def factorize(n=13)

2 res = ''

3 while 13 > 2 and 13 % 2 == 0: --- False

6 d = 3

7 while 13 > 3: --- True

8 if 13 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 13 > 5: --- True

8 if 13 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 13 > 7: --- True

8 if 13 % 7 == 0: --- False

11 else:

12 d = 7 + 2

d = 9

7 while 13 > 9: --- True

8 if 13 % 9 == 0: --- False

11 else:

12 d = 9 + 2

d = 11

7 while 13 > 11: --- True

8 if 13 % 11 == 0: --- False

11 else:

12 d = 11 + 2

d = 13

7 while 13 > 13: --- False

13 return '' + str(13)

return '13'

7. remove\_digit(323, 2) = 33

1 def remove\_digit(number=323, digit=2)

2 res = 0

3 power = 1

4 while 323 > 0: --- True

5 cur\_digit = 323 % 10

cur\_digit = 3

6 if 3 != 2: --- True

7 res = 0 + 3 \* 1

res = 3

8 power = 1 \* 10

power = 10

9 number = 323 // 10

number = 32

4 while 32 > 0: --- True

5 cur\_digit = 32 % 10

cur\_digit = 2

6 if 2 != 2: --- False

9 number = 32 // 10

number = 3

4 while 3 > 0: --- True

5 cur\_digit = 3 % 10

cur\_digit = 3

6 if 3 != 2: --- True

7 res = 3 + 3 \* 10

res = 33

8 power = 10 \* 10

power = 100

9 number = 3 // 10

number = 0

4 while 0 > 0: --- False

10 return 33

Вариант: 1-2-18

1. gcd(-66, 24) = 6

1 def gcd(x=-66, y=24)

2 if -66 < 0: --- True

3 x = --66

x = 66

4 if 24 < 0: --- False

6 while 24 != 0: --- True

7 rem = 66 % 24

rem = 18

8 x = 24

9 y = 18

6 while 18 != 0: --- True

7 rem = 24 % 18

rem = 6

8 x = 18

9 y = 6

6 while 6 != 0: --- True

7 rem = 18 % 6

rem = 0

8 x = 6

9 y = 0

6 while 0 != 0: --- False

10 return 6

2. gcd(0, 61) = 61

1 def gcd(x=0, y=61)

2 if 0 < 0: --- False

4 if 61 < 0: --- False

6 while 61 != 0: --- True

7 rem = 0 % 61

rem = 0

8 x = 61

9 y = 0

6 while 0 != 0: --- False

10 return 61

3. hex(207) = 'CF'

1 def hex(number=207)

2 if 207 == 0: --- False

4 res = ''

5 while 207 > 0: --- True

6 digit = 207 % 16

digit = 15

7 if 15 <= 9: --- False

9 elif 15 <= 13: --- False

19 elif 15 == 14: --- False

21 else:

22 digit = 'F'

23 res = 'F' + ''

res = 'F'

24 number = 207 // 16

number = 12

5 while 12 > 0: --- True

6 digit = 12 % 16

digit = 12

7 if 12 <= 9: --- False

9 elif 12 <= 13: --- True

10 if 12 <= 11: --- False

15 elif 12 == 12: --- True

16 digit = 'C'

23 res = 'C' + 'F'

res = 'CF'

24 number = 12 // 16

number = 0

5 while 0 > 0: --- False

25 return 'CF'

4. square\_equal(-4, 45, 49) = '12.25 and -1.0'

3 def square\_equal(a=-4, b=45, c=49)

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

5 D = 45\*45 - 4\*-4\*49

D = 2809

6 if 2809 > 0: --- True

7 x1 = (-45 - sqrt(2809)) / (2\*-4)

x1 = 12.25

8 x2 = (-45 + sqrt(2809)) / (2\*-4)

x2 = -1.0

9 return str(12.25) + ' and ' + str(-1.0)

return '12.25 and -1.0'

5. square\_equal(-11, 37, -84) = 'no roots'

3 def square\_equal(a=-11, b=37, c=-84)

4 if -11 != 0: --- True

5 D = 37\*37 - 4\*-11\*-84

D = -2327

6 if -2327 > 0: --- False

10 elif -2327 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(75) = '3\*5\*5'

1 def factorize(n=75)

2 res = ''

3 while 75 > 2 and 75 % 2 == 0: --- False

6 d = 3

7 while 75 > 3: --- True

8 if 75 % 3 == 0: --- True

9 res = '' + str(3) + '\*'

res = '3\*'

10 n = 75 // 3

n = 25

7 while 25 > 3: --- True

8 if 25 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 25 > 5: --- True

8 if 25 % 5 == 0: --- True

9 res = '3\*' + str(5) + '\*'

res = '3\*5\*'

10 n = 25 // 5

n = 5

7 while 5 > 5: --- False

13 return '3\*5\*' + str(5)

return '3\*5\*5'

7. remove\_digit(6333, 3) = 6

1 def remove\_digit(number=6333, digit=3)

2 res = 0

3 power = 1

4 while 6333 > 0: --- True

5 cur\_digit = 6333 % 10

cur\_digit = 3

6 if 3 != 3: --- False

9 number = 6333 // 10

number = 633

4 while 633 > 0: --- True

5 cur\_digit = 633 % 10

cur\_digit = 3

6 if 3 != 3: --- False

9 number = 633 // 10

number = 63

4 while 63 > 0: --- True

5 cur\_digit = 63 % 10

cur\_digit = 3

6 if 3 != 3: --- False

9 number = 63 // 10

number = 6

4 while 6 > 0: --- True

5 cur\_digit = 6 % 10

cur\_digit = 6

6 if 6 != 3: --- True

7 res = 0 + 6 \* 1

res = 6

8 power = 1 \* 10

power = 10

9 number = 6 // 10

number = 0

4 while 0 > 0: --- False

10 return 6

Вариант: 1-2-19

1. gcd(-52, -78) = 26

1 def gcd(x=-52, y=-78)

2 if -52 < 0: --- True

3 x = --52

x = 52

4 if -78 < 0: --- True

5 y = --78

y = 78

6 while 78 != 0: --- True

7 rem = 52 % 78

rem = 52

8 x = 78

9 y = 52

6 while 52 != 0: --- True

7 rem = 78 % 52

rem = 26

8 x = 52

9 y = 26

6 while 26 != 0: --- True

7 rem = 52 % 26

rem = 0

8 x = 26

9 y = 0

6 while 0 != 0: --- False

10 return 26

2. gcd(0, -72) = 72

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

2 if 0 < 0: --- False

4 if -72 < 0: --- True

5 y = --72

y = 72

6 while 72 != 0: --- True

7 rem = 0 % 72

rem = 0

8 x = 72

9 y = 0

6 while 0 != 0: --- False

10 return 72

3. hex(169) = 'A9'

1 def hex(number=169)

2 if 169 == 0: --- False

4 res = ''

5 while 169 > 0: --- True

6 digit = 169 % 16

digit = 9

7 if 9 <= 9: --- True

8 digit = str(9)

digit = '9'

23 res = '9' + ''

res = '9'

24 number = 169 // 16

number = 10

5 while 10 > 0: --- True

6 digit = 10 % 16

digit = 10

7 if 10 <= 9: --- False

9 elif 10 <= 13: --- True

10 if 10 <= 11: --- True

11 if 10 == 10: --- True

12 digit = 'A'

23 res = 'A' + '9'

res = 'A9'

24 number = 10 // 16

number = 0

5 while 0 > 0: --- False

25 return 'A9'

4. square\_equal(-1, -52, -51) = '-1.0 and -51.0'

3 def square\_equal(a=-1, b=-52, c=-51)

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

5 D = -52\*-52 - 4\*-1\*-51

D = 2500

6 if 2500 > 0: --- True

7 x1 = (--52 - sqrt(2500)) / (2\*-1)

x1 = -1.0

8 x2 = (--52 + sqrt(2500)) / (2\*-1)

x2 = -51.0

9 return str(-1.0) + ' and ' + str(-51.0)

return '-1.0 and -51.0'

5. square\_equal(-100, -81, -40) = 'no roots'

3 def square\_equal(a=-100, b=-81, c=-40)

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

5 D = -81\*-81 - 4\*-100\*-40

D = -9439

6 if -9439 > 0: --- False

10 elif -9439 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(108) = '2\*2\*3\*3\*3'

1 def factorize(n=108)

2 res = ''

3 while 108 > 2 and 108 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 108 // 2

n = 54

3 while 54 > 2 and 54 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 54 // 2

n = 27

3 while 27 > 2 and 27 % 2 == 0: --- False

6 d = 3

7 while 27 > 3: --- True

8 if 27 % 3 == 0: --- True

9 res = '2\*2\*' + str(3) + '\*'

res = '2\*2\*3\*'

10 n = 27 // 3

n = 9

7 while 9 > 3: --- True

8 if 9 % 3 == 0: --- True

9 res = '2\*2\*3\*' + str(3) + '\*'

res = '2\*2\*3\*3\*'

10 n = 9 // 3

n = 3

7 while 3 > 3: --- False

13 return '2\*2\*3\*3\*' + str(3)

return '2\*2\*3\*3\*3'

7. remove\_digit(784, 4) = 78

1 def remove\_digit(number=784, digit=4)

2 res = 0

3 power = 1

4 while 784 > 0: --- True

5 cur\_digit = 784 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 784 // 10

number = 78

4 while 78 > 0: --- True

5 cur\_digit = 78 % 10

cur\_digit = 8

6 if 8 != 4: --- True

7 res = 0 + 8 \* 1

res = 8

8 power = 1 \* 10

power = 10

9 number = 78 // 10

number = 7

4 while 7 > 0: --- True

5 cur\_digit = 7 % 10

cur\_digit = 7

6 if 7 != 4: --- True

7 res = 8 + 7 \* 10

res = 78

8 power = 10 \* 10

power = 100

9 number = 7 // 10

number = 0

4 while 0 > 0: --- False

10 return 78

Вариант: 1-2-20

1. gcd(-12, 76) = 4

1 def gcd(x=-12, y=76)

2 if -12 < 0: --- True

3 x = --12

x = 12

4 if 76 < 0: --- False

6 while 76 != 0: --- True

7 rem = 12 % 76

rem = 12

8 x = 76

9 y = 12

6 while 12 != 0: --- True

7 rem = 76 % 12

rem = 4

8 x = 12

9 y = 4

6 while 4 != 0: --- True

7 rem = 12 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, -97) = 97

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

2 if 0 < 0: --- False

4 if -97 < 0: --- True

5 y = --97

y = 97

6 while 97 != 0: --- True

7 rem = 0 % 97

rem = 0

8 x = 97

9 y = 0

6 while 0 != 0: --- False

10 return 97

3. hex(211) = 'D3'

1 def hex(number=211)

2 if 211 == 0: --- False

4 res = ''

5 while 211 > 0: --- True

6 digit = 211 % 16

digit = 3

7 if 3 <= 9: --- True

8 digit = str(3)

digit = '3'

23 res = '3' + ''

res = '3'

24 number = 211 // 16

number = 13

5 while 13 > 0: --- True

6 digit = 13 % 16

digit = 13

7 if 13 <= 9: --- False

9 elif 13 <= 13: --- True

10 if 13 <= 11: --- False

15 elif 13 == 12: --- False

17 else:

18 digit = 'D'

23 res = 'D' + '3'

res = 'D3'

24 number = 13 // 16

number = 0

5 while 0 > 0: --- False

25 return 'D3'

4. square\_equal(3, 12, -96) = '-8.0 and 4.0'

3 def square\_equal(a=3, b=12, c=-96)

4 if 3 != 0: --- True

5 D = 12\*12 - 4\*3\*-96

D = 1296

6 if 1296 > 0: --- True

7 x1 = (-12 - sqrt(1296)) / (2\*3)

x1 = -8.0

8 x2 = (-12 + sqrt(1296)) / (2\*3)

x2 = 4.0

9 return str(-8.0) + ' and ' + str(4.0)

return '-8.0 and 4.0'

5. square\_equal(-25, 15, -16) = 'no roots'

3 def square\_equal(a=-25, b=15, c=-16)

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

5 D = 15\*15 - 4\*-25\*-16

D = -1375

6 if -1375 > 0: --- False

10 elif -1375 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(490) = '2\*5\*7\*7'

1 def factorize(n=490)

2 res = ''

3 while 490 > 2 and 490 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 490 // 2

n = 245

3 while 245 > 2 and 245 % 2 == 0: --- False

6 d = 3

7 while 245 > 3: --- True

8 if 245 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 245 > 5: --- True

8 if 245 % 5 == 0: --- True

9 res = '2\*' + str(5) + '\*'

res = '2\*5\*'

10 n = 245 // 5

n = 49

7 while 49 > 5: --- True

8 if 49 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 49 > 7: --- True

8 if 49 % 7 == 0: --- True

9 res = '2\*5\*' + str(7) + '\*'

res = '2\*5\*7\*'

10 n = 49 // 7

n = 7

7 while 7 > 7: --- False

13 return '2\*5\*7\*' + str(7)

return '2\*5\*7\*7'

7. remove\_digit(262, 6) = 22

1 def remove\_digit(number=262, digit=6)

2 res = 0

3 power = 1

4 while 262 > 0: --- True

5 cur\_digit = 262 % 10

cur\_digit = 2

6 if 2 != 6: --- True

7 res = 0 + 2 \* 1

res = 2

8 power = 1 \* 10

power = 10

9 number = 262 // 10

number = 26

4 while 26 > 0: --- True

5 cur\_digit = 26 % 10

cur\_digit = 6

6 if 6 != 6: --- False

9 number = 26 // 10

number = 2

4 while 2 > 0: --- True

5 cur\_digit = 2 % 10

cur\_digit = 2

6 if 2 != 6: --- True

7 res = 2 + 2 \* 10

res = 22

8 power = 10 \* 10

power = 100

9 number = 2 // 10

number = 0

4 while 0 > 0: --- False

10 return 22

Вариант: 1-2-21

1. gcd(72, 90) = 18

1 def gcd(x=72, y=90)

2 if 72 < 0: --- False

4 if 90 < 0: --- False

6 while 90 != 0: --- True

7 rem = 72 % 90

rem = 72

8 x = 90

9 y = 72

6 while 72 != 0: --- True

7 rem = 90 % 72

rem = 18

8 x = 72

9 y = 18

6 while 18 != 0: --- True

7 rem = 72 % 18

rem = 0

8 x = 18

9 y = 0

6 while 0 != 0: --- False

10 return 18

2. gcd(0, -15) = 15

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

2 if 0 < 0: --- False

4 if -15 < 0: --- True

5 y = --15

y = 15

6 while 15 != 0: --- True

7 rem = 0 % 15

rem = 0

8 x = 15

9 y = 0

6 while 0 != 0: --- False

10 return 15

3. hex(160) = 'A0'

1 def hex(number=160)

2 if 160 == 0: --- False

4 res = ''

5 while 160 > 0: --- True

6 digit = 160 % 16

digit = 0

7 if 0 <= 9: --- True

8 digit = str(0)

digit = '0'

23 res = '0' + ''

res = '0'

24 number = 160 // 16

number = 10

5 while 10 > 0: --- True

6 digit = 10 % 16

digit = 10

7 if 10 <= 9: --- False

9 elif 10 <= 13: --- True

10 if 10 <= 11: --- True

11 if 10 == 10: --- True

12 digit = 'A'

23 res = 'A' + '0'

res = 'A0'

24 number = 10 // 16

number = 0

5 while 0 > 0: --- False

25 return 'A0'

4. square\_equal(-15, 33, 0) = '2.2 and -0.0'

3 def square\_equal(a=-15, b=33, c=0)

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

5 D = 33\*33 - 4\*-15\*0

D = 1089

6 if 1089 > 0: --- True

7 x1 = (-33 - sqrt(1089)) / (2\*-15)

x1 = 2.2

8 x2 = (-33 + sqrt(1089)) / (2\*-15)

x2 = -0.0

9 return str(2.2) + ' and ' + str(-0.0)

return '2.2 and -0.0'

5. square\_equal(-19, 18, -43) = 'no roots'

3 def square\_equal(a=-19, b=18, c=-43)

4 if -19 != 0: --- True

5 D = 18\*18 - 4\*-19\*-43

D = -2944

6 if -2944 > 0: --- False

10 elif -2944 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(126) = '2\*3\*3\*7'

1 def factorize(n=126)

2 res = ''

3 while 126 > 2 and 126 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 126 // 2

n = 63

3 while 63 > 2 and 63 % 2 == 0: --- False

6 d = 3

7 while 63 > 3: --- True

8 if 63 % 3 == 0: --- True

9 res = '2\*' + str(3) + '\*'

res = '2\*3\*'

10 n = 63 // 3

n = 21

7 while 21 > 3: --- True

8 if 21 % 3 == 0: --- True

9 res = '2\*3\*' + str(3) + '\*'

res = '2\*3\*3\*'

10 n = 21 // 3

n = 7

7 while 7 > 3: --- True

8 if 7 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 7 > 5: --- True

8 if 7 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 7 > 7: --- False

13 return '2\*3\*3\*' + str(7)

return '2\*3\*3\*7'

7. remove\_digit(8999, 9) = 8

1 def remove\_digit(number=8999, digit=9)

2 res = 0

3 power = 1

4 while 8999 > 0: --- True

5 cur\_digit = 8999 % 10

cur\_digit = 9

6 if 9 != 9: --- False

9 number = 8999 // 10

number = 899

4 while 899 > 0: --- True

5 cur\_digit = 899 % 10

cur\_digit = 9

6 if 9 != 9: --- False

9 number = 899 // 10

number = 89

4 while 89 > 0: --- True

5 cur\_digit = 89 % 10

cur\_digit = 9

6 if 9 != 9: --- False

9 number = 89 // 10

number = 8

4 while 8 > 0: --- True

5 cur\_digit = 8 % 10

cur\_digit = 8

6 if 8 != 9: --- True

7 res = 0 + 8 \* 1

res = 8

8 power = 1 \* 10

power = 10

9 number = 8 // 10

number = 0

4 while 0 > 0: --- False

10 return 8

Вариант: 1-2-22

1. gcd(-92, -48) = 4

1 def gcd(x=-92, y=-48)

2 if -92 < 0: --- True

3 x = --92

x = 92

4 if -48 < 0: --- True

5 y = --48

y = 48

6 while 48 != 0: --- True

7 rem = 92 % 48

rem = 44

8 x = 48

9 y = 44

6 while 44 != 0: --- True

7 rem = 48 % 44

rem = 4

8 x = 44

9 y = 4

6 while 4 != 0: --- True

7 rem = 44 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, -91) = 91

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

2 if 0 < 0: --- False

4 if -91 < 0: --- True

5 y = --91

y = 91

6 while 91 != 0: --- True

7 rem = 0 % 91

rem = 0

8 x = 91

9 y = 0

6 while 0 != 0: --- False

10 return 91

3. hex(230) = 'E6'

1 def hex(number=230)

2 if 230 == 0: --- False

4 res = ''

5 while 230 > 0: --- True

6 digit = 230 % 16

digit = 6

7 if 6 <= 9: --- True

8 digit = str(6)

digit = '6'

23 res = '6' + ''

res = '6'

24 number = 230 // 16

number = 14

5 while 14 > 0: --- True

6 digit = 14 % 16

digit = 14

7 if 14 <= 9: --- False

9 elif 14 <= 13: --- False

19 elif 14 == 14: --- True

20 digit = 'E'

23 res = 'E' + '6'

res = 'E6'

24 number = 14 // 16

number = 0

5 while 0 > 0: --- False

25 return 'E6'

4. square\_equal(-6, -9, 6) = '0.5 and -2.0'

3 def square\_equal(a=-6, b=-9, c=6)

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

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

D = 225

6 if 225 > 0: --- True

7 x1 = (--9 - sqrt(225)) / (2\*-6)

x1 = 0.5

8 x2 = (--9 + sqrt(225)) / (2\*-6)

x2 = -2.0

9 return str(0.5) + ' and ' + str(-2.0)

return '0.5 and -2.0'

5. square\_equal(65, -4, 13) = 'no roots'

3 def square\_equal(a=65, b=-4, c=13)

4 if 65 != 0: --- True

5 D = -4\*-4 - 4\*65\*13

D = -3364

6 if -3364 > 0: --- False

10 elif -3364 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(11) = '11'

1 def factorize(n=11)

2 res = ''

3 while 11 > 2 and 11 % 2 == 0: --- False

6 d = 3

7 while 11 > 3: --- True

8 if 11 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 11 > 5: --- True

8 if 11 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 11 > 7: --- True

8 if 11 % 7 == 0: --- False

11 else:

12 d = 7 + 2

d = 9

7 while 11 > 9: --- True

8 if 11 % 9 == 0: --- False

11 else:

12 d = 9 + 2

d = 11

7 while 11 > 11: --- False

13 return '' + str(11)

return '11'

7. remove\_digit(8333, 3) = 8

1 def remove\_digit(number=8333, digit=3)

2 res = 0

3 power = 1

4 while 8333 > 0: --- True

5 cur\_digit = 8333 % 10

cur\_digit = 3

6 if 3 != 3: --- False

9 number = 8333 // 10

number = 833

4 while 833 > 0: --- True

5 cur\_digit = 833 % 10

cur\_digit = 3

6 if 3 != 3: --- False

9 number = 833 // 10

number = 83

4 while 83 > 0: --- True

5 cur\_digit = 83 % 10

cur\_digit = 3

6 if 3 != 3: --- False

9 number = 83 // 10

number = 8

4 while 8 > 0: --- True

5 cur\_digit = 8 % 10

cur\_digit = 8

6 if 8 != 3: --- True

7 res = 0 + 8 \* 1

res = 8

8 power = 1 \* 10

power = 10

9 number = 8 // 10

number = 0

4 while 0 > 0: --- False

10 return 8

Вариант: 1-2-23

1. gcd(-9, -15) = 3

1 def gcd(x=-9, y=-15)

2 if -9 < 0: --- True

3 x = --9

x = 9

4 if -15 < 0: --- True

5 y = --15

y = 15

6 while 15 != 0: --- True

7 rem = 9 % 15

rem = 9

8 x = 15

9 y = 9

6 while 9 != 0: --- True

7 rem = 15 % 9

rem = 6

8 x = 9

9 y = 6

6 while 6 != 0: --- True

7 rem = 9 % 6

rem = 3

8 x = 6

9 y = 3

6 while 3 != 0: --- True

7 rem = 6 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(4, 0) = 4

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

2 if 4 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 4

3. hex(233) = 'E9'

1 def hex(number=233)

2 if 233 == 0: --- False

4 res = ''

5 while 233 > 0: --- True

6 digit = 233 % 16

digit = 9

7 if 9 <= 9: --- True

8 digit = str(9)

digit = '9'

23 res = '9' + ''

res = '9'

24 number = 233 // 16

number = 14

5 while 14 > 0: --- True

6 digit = 14 % 16

digit = 14

7 if 14 <= 9: --- False

9 elif 14 <= 13: --- False

19 elif 14 == 14: --- True

20 digit = 'E'

23 res = 'E' + '9'

res = 'E9'

24 number = 14 // 16

number = 0

5 while 0 > 0: --- False

25 return 'E9'

4. square\_equal(-15, -69, -54) = '-1.0 and -3.6'

3 def square\_equal(a=-15, b=-69, c=-54)

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

5 D = -69\*-69 - 4\*-15\*-54

D = 1521

6 if 1521 > 0: --- True

7 x1 = (--69 - sqrt(1521)) / (2\*-15)

x1 = -1.0

8 x2 = (--69 + sqrt(1521)) / (2\*-15)

x2 = -3.6

9 return str(-1.0) + ' and ' + str(-3.6)

return '-1.0 and -3.6'

5. square\_equal(27, 5, 74) = 'no roots'

3 def square\_equal(a=27, b=5, c=74)

4 if 27 != 0: --- True

5 D = 5\*5 - 4\*27\*74

D = -7967

6 if -7967 > 0: --- False

10 elif -7967 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(7) = '7'

1 def factorize(n=7)

2 res = ''

3 while 7 > 2 and 7 % 2 == 0: --- False

6 d = 3

7 while 7 > 3: --- True

8 if 7 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 7 > 5: --- True

8 if 7 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 7 > 7: --- False

13 return '' + str(7)

return '7'

7. remove\_digit(326, 6) = 32

1 def remove\_digit(number=326, digit=6)

2 res = 0

3 power = 1

4 while 326 > 0: --- True

5 cur\_digit = 326 % 10

cur\_digit = 6

6 if 6 != 6: --- False

9 number = 326 // 10

number = 32

4 while 32 > 0: --- True

5 cur\_digit = 32 % 10

cur\_digit = 2

6 if 2 != 6: --- True

7 res = 0 + 2 \* 1

res = 2

8 power = 1 \* 10

power = 10

9 number = 32 // 10

number = 3

4 while 3 > 0: --- True

5 cur\_digit = 3 % 10

cur\_digit = 3

6 if 3 != 6: --- True

7 res = 2 + 3 \* 10

res = 32

8 power = 10 \* 10

power = 100

9 number = 3 // 10

number = 0

4 while 0 > 0: --- False

10 return 32

Вариант: 1-2-24

1. gcd(-42, 48) = 6

1 def gcd(x=-42, y=48)

2 if -42 < 0: --- True

3 x = --42

x = 42

4 if 48 < 0: --- False

6 while 48 != 0: --- True

7 rem = 42 % 48

rem = 42

8 x = 48

9 y = 42

6 while 42 != 0: --- True

7 rem = 48 % 42

rem = 6

8 x = 42

9 y = 6

6 while 6 != 0: --- True

7 rem = 42 % 6

rem = 0

8 x = 6

9 y = 0

6 while 0 != 0: --- False

10 return 6

2. gcd(0, 47) = 47

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

2 if 0 < 0: --- False

4 if 47 < 0: --- False

6 while 47 != 0: --- True

7 rem = 0 % 47

rem = 0

8 x = 47

9 y = 0

6 while 0 != 0: --- False

10 return 47

3. hex(225) = 'E1'

1 def hex(number=225)

2 if 225 == 0: --- False

4 res = ''

5 while 225 > 0: --- True

6 digit = 225 % 16

digit = 1

7 if 1 <= 9: --- True

8 digit = str(1)

digit = '1'

23 res = '1' + ''

res = '1'

24 number = 225 // 16

number = 14

5 while 14 > 0: --- True

6 digit = 14 % 16

digit = 14

7 if 14 <= 9: --- False

9 elif 14 <= 13: --- False

19 elif 14 == 14: --- True

20 digit = 'E'

23 res = 'E' + '1'

res = 'E1'

24 number = 14 // 16

number = 0

5 while 0 > 0: --- False

25 return 'E1'

4. square\_equal(-16, -36, -14) = '-0.5 and -1.75'

3 def square\_equal(a=-16, b=-36, c=-14)

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

5 D = -36\*-36 - 4\*-16\*-14

D = 400

6 if 400 > 0: --- True

7 x1 = (--36 - sqrt(400)) / (2\*-16)

x1 = -0.5

8 x2 = (--36 + sqrt(400)) / (2\*-16)

x2 = -1.75

9 return str(-0.5) + ' and ' + str(-1.75)

return '-0.5 and -1.75'

5. square\_equal(-25, 28, -67) = 'no roots'

3 def square\_equal(a=-25, b=28, c=-67)

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

5 D = 28\*28 - 4\*-25\*-67

D = -5916

6 if -5916 > 0: --- False

10 elif -5916 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(15) = '3\*5'

1 def factorize(n=15)

2 res = ''

3 while 15 > 2 and 15 % 2 == 0: --- False

6 d = 3

7 while 15 > 3: --- True

8 if 15 % 3 == 0: --- True

9 res = '' + str(3) + '\*'

res = '3\*'

10 n = 15 // 3

n = 5

7 while 5 > 3: --- True

8 if 5 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 5 > 5: --- False

13 return '3\*' + str(5)

return '3\*5'

7. remove\_digit(744, 4) = 7

1 def remove\_digit(number=744, digit=4)

2 res = 0

3 power = 1

4 while 744 > 0: --- True

5 cur\_digit = 744 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 744 // 10

number = 74

4 while 74 > 0: --- True

5 cur\_digit = 74 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 74 // 10

number = 7

4 while 7 > 0: --- True

5 cur\_digit = 7 % 10

cur\_digit = 7

6 if 7 != 4: --- True

7 res = 0 + 7 \* 1

res = 7

8 power = 1 \* 10

power = 10

9 number = 7 // 10

number = 0

4 while 0 > 0: --- False

10 return 7

Вариант: 1-2-25

1. gcd(-22, 77) = 11

1 def gcd(x=-22, y=77)

2 if -22 < 0: --- True

3 x = --22

x = 22

4 if 77 < 0: --- False

6 while 77 != 0: --- True

7 rem = 22 % 77

rem = 22

8 x = 77

9 y = 22

6 while 22 != 0: --- True

7 rem = 77 % 22

rem = 11

8 x = 22

9 y = 11

6 while 11 != 0: --- True

7 rem = 22 % 11

rem = 0

8 x = 11

9 y = 0

6 while 0 != 0: --- False

10 return 11

2. gcd(84, 0) = 84

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

2 if 84 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 84

3. hex(247) = 'F7'

1 def hex(number=247)

2 if 247 == 0: --- False

4 res = ''

5 while 247 > 0: --- True

6 digit = 247 % 16

digit = 7

7 if 7 <= 9: --- True

8 digit = str(7)

digit = '7'

23 res = '7' + ''

res = '7'

24 number = 247 // 16

number = 15

5 while 15 > 0: --- True

6 digit = 15 % 16

digit = 15

7 if 15 <= 9: --- False

9 elif 15 <= 13: --- False

19 elif 15 == 14: --- False

21 else:

22 digit = 'F'

23 res = 'F' + '7'

res = 'F7'

24 number = 15 // 16

number = 0

5 while 0 > 0: --- False

25 return 'F7'

4. square\_equal(16, -56, 33) = '0.75 and 2.75'

3 def square\_equal(a=16, b=-56, c=33)

4 if 16 != 0: --- True

5 D = -56\*-56 - 4\*16\*33

D = 1024

6 if 1024 > 0: --- True

7 x1 = (--56 - sqrt(1024)) / (2\*16)

x1 = 0.75

8 x2 = (--56 + sqrt(1024)) / (2\*16)

x2 = 2.75

9 return str(0.75) + ' and ' + str(2.75)

return '0.75 and 2.75'

5. square\_equal(55, 66, 41) = 'no roots'

3 def square\_equal(a=55, b=66, c=41)

4 if 55 != 0: --- True

5 D = 66\*66 - 4\*55\*41

D = -4664

6 if -4664 > 0: --- False

10 elif -4664 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(96) = '2\*2\*2\*2\*2\*3'

1 def factorize(n=96)

2 res = ''

3 while 96 > 2 and 96 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 96 // 2

n = 48

3 while 48 > 2 and 48 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 48 // 2

n = 24

3 while 24 > 2 and 24 % 2 == 0: --- True

4 res = '2\*2\*' + '2\*'

res = '2\*2\*2\*'

5 n = 24 // 2

n = 12

3 while 12 > 2 and 12 % 2 == 0: --- True

4 res = '2\*2\*2\*' + '2\*'

res = '2\*2\*2\*2\*'

5 n = 12 // 2

n = 6

3 while 6 > 2 and 6 % 2 == 0: --- True

4 res = '2\*2\*2\*2\*' + '2\*'

res = '2\*2\*2\*2\*2\*'

5 n = 6 // 2

n = 3

3 while 3 > 2 and 3 % 2 == 0: --- False

6 d = 3

7 while 3 > 3: --- False

13 return '2\*2\*2\*2\*2\*' + str(3)

return '2\*2\*2\*2\*2\*3'

7. remove\_digit(670, 0) = 67

1 def remove\_digit(number=670, digit=0)

2 res = 0

3 power = 1

4 while 670 > 0: --- True

5 cur\_digit = 670 % 10

cur\_digit = 0

6 if 0 != 0: --- False

9 number = 670 // 10

number = 67

4 while 67 > 0: --- True

5 cur\_digit = 67 % 10

cur\_digit = 7

6 if 7 != 0: --- True

7 res = 0 + 7 \* 1

res = 7

8 power = 1 \* 10

power = 10

9 number = 67 // 10

number = 6

4 while 6 > 0: --- True

5 cur\_digit = 6 % 10

cur\_digit = 6

6 if 6 != 0: --- True

7 res = 7 + 6 \* 10

res = 67

8 power = 10 \* 10

power = 100

9 number = 6 // 10

number = 0

4 while 0 > 0: --- False

10 return 67

Вариант: 1-2-26

1. gcd(-27, 87) = 3

1 def gcd(x=-27, y=87)

2 if -27 < 0: --- True

3 x = --27

x = 27

4 if 87 < 0: --- False

6 while 87 != 0: --- True

7 rem = 27 % 87

rem = 27

8 x = 87

9 y = 27

6 while 27 != 0: --- True

7 rem = 87 % 27

rem = 6

8 x = 27

9 y = 6

6 while 6 != 0: --- True

7 rem = 27 % 6

rem = 3

8 x = 6

9 y = 3

6 while 3 != 0: --- True

7 rem = 6 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(0, 96) = 96

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

2 if 0 < 0: --- False

4 if 96 < 0: --- False

6 while 96 != 0: --- True

7 rem = 0 % 96

rem = 0

8 x = 96

9 y = 0

6 while 0 != 0: --- False

10 return 96

3. hex(218) = 'DA'

1 def hex(number=218)

2 if 218 == 0: --- False

4 res = ''

5 while 218 > 0: --- True

6 digit = 218 % 16

digit = 10

7 if 10 <= 9: --- False

9 elif 10 <= 13: --- True

10 if 10 <= 11: --- True

11 if 10 == 10: --- True

12 digit = 'A'

23 res = 'A' + ''

res = 'A'

24 number = 218 // 16

number = 13

5 while 13 > 0: --- True

6 digit = 13 % 16

digit = 13

7 if 13 <= 9: --- False

9 elif 13 <= 13: --- True

10 if 13 <= 11: --- False

15 elif 13 == 12: --- False

17 else:

18 digit = 'D'

23 res = 'D' + 'A'

res = 'DA'

24 number = 13 // 16

number = 0

5 while 0 > 0: --- False

25 return 'DA'

4. square\_equal(100, -35, 3) = '0.15 and 0.2'

3 def square\_equal(a=100, b=-35, c=3)

4 if 100 != 0: --- True

5 D = -35\*-35 - 4\*100\*3

D = 25

6 if 25 > 0: --- True

7 x1 = (--35 - sqrt(25)) / (2\*100)

x1 = 0.15

8 x2 = (--35 + sqrt(25)) / (2\*100)

x2 = 0.2

9 return str(0.15) + ' and ' + str(0.2)

return '0.15 and 0.2'

5. square\_equal(-53, 32, -37) = 'no roots'

3 def square\_equal(a=-53, b=32, c=-37)

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

5 D = 32\*32 - 4\*-53\*-37

D = -6820

6 if -6820 > 0: --- False

10 elif -6820 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(300) = '2\*2\*3\*5\*5'

1 def factorize(n=300)

2 res = ''

3 while 300 > 2 and 300 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 300 // 2

n = 150

3 while 150 > 2 and 150 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 150 // 2

n = 75

3 while 75 > 2 and 75 % 2 == 0: --- False

6 d = 3

7 while 75 > 3: --- True

8 if 75 % 3 == 0: --- True

9 res = '2\*2\*' + str(3) + '\*'

res = '2\*2\*3\*'

10 n = 75 // 3

n = 25

7 while 25 > 3: --- True

8 if 25 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 25 > 5: --- True

8 if 25 % 5 == 0: --- True

9 res = '2\*2\*3\*' + str(5) + '\*'

res = '2\*2\*3\*5\*'

10 n = 25 // 5

n = 5

7 while 5 > 5: --- False

13 return '2\*2\*3\*5\*' + str(5)

return '2\*2\*3\*5\*5'

7. remove\_digit(502, 0) = 52

1 def remove\_digit(number=502, digit=0)

2 res = 0

3 power = 1

4 while 502 > 0: --- True

5 cur\_digit = 502 % 10

cur\_digit = 2

6 if 2 != 0: --- True

7 res = 0 + 2 \* 1

res = 2

8 power = 1 \* 10

power = 10

9 number = 502 // 10

number = 50

4 while 50 > 0: --- True

5 cur\_digit = 50 % 10

cur\_digit = 0

6 if 0 != 0: --- False

9 number = 50 // 10

number = 5

4 while 5 > 0: --- True

5 cur\_digit = 5 % 10

cur\_digit = 5

6 if 5 != 0: --- True

7 res = 2 + 5 \* 10

res = 52

8 power = 10 \* 10

power = 100

9 number = 5 // 10

number = 0

4 while 0 > 0: --- False

10 return 52

Вариант: 1-2-27

1. gcd(16, -88) = 8

1 def gcd(x=16, y=-88)

2 if 16 < 0: --- False

4 if -88 < 0: --- True

5 y = --88

y = 88

6 while 88 != 0: --- True

7 rem = 16 % 88

rem = 16

8 x = 88

9 y = 16

6 while 16 != 0: --- True

7 rem = 88 % 16

rem = 8

8 x = 16

9 y = 8

6 while 8 != 0: --- True

7 rem = 16 % 8

rem = 0

8 x = 8

9 y = 0

6 while 0 != 0: --- False

10 return 8

2. gcd(100, 0) = 100

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

2 if 100 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 100

3. hex(244) = 'F4'

1 def hex(number=244)

2 if 244 == 0: --- False

4 res = ''

5 while 244 > 0: --- True

6 digit = 244 % 16

digit = 4

7 if 4 <= 9: --- True

8 digit = str(4)

digit = '4'

23 res = '4' + ''

res = '4'

24 number = 244 // 16

number = 15

5 while 15 > 0: --- True

6 digit = 15 % 16

digit = 15

7 if 15 <= 9: --- False

9 elif 15 <= 13: --- False

19 elif 15 == 14: --- False

21 else:

22 digit = 'F'

23 res = 'F' + '4'

res = 'F4'

24 number = 15 // 16

number = 0

5 while 0 > 0: --- False

25 return 'F4'

4. square\_equal(1, -45, -46) = '-1.0 and 46.0'

3 def square\_equal(a=1, b=-45, c=-46)

4 if 1 != 0: --- True

5 D = -45\*-45 - 4\*1\*-46

D = 2209

6 if 2209 > 0: --- True

7 x1 = (--45 - sqrt(2209)) / (2\*1)

x1 = -1.0

8 x2 = (--45 + sqrt(2209)) / (2\*1)

x2 = 46.0

9 return str(-1.0) + ' and ' + str(46.0)

return '-1.0 and 46.0'

5. square\_equal(-93, -8, -3) = 'no roots'

3 def square\_equal(a=-93, b=-8, c=-3)

4 if -93 != 0: --- True

5 D = -8\*-8 - 4\*-93\*-3

D = -1052

6 if -1052 > 0: --- False

10 elif -1052 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(24) = '2\*2\*2\*3'

1 def factorize(n=24)

2 res = ''

3 while 24 > 2 and 24 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 24 // 2

n = 12

3 while 12 > 2 and 12 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 12 // 2

n = 6

3 while 6 > 2 and 6 % 2 == 0: --- True

4 res = '2\*2\*' + '2\*'

res = '2\*2\*2\*'

5 n = 6 // 2

n = 3

3 while 3 > 2 and 3 % 2 == 0: --- False

6 d = 3

7 while 3 > 3: --- False

13 return '2\*2\*2\*' + str(3)

return '2\*2\*2\*3'

7. remove\_digit(737, 3) = 77

1 def remove\_digit(number=737, digit=3)

2 res = 0

3 power = 1

4 while 737 > 0: --- True

5 cur\_digit = 737 % 10

cur\_digit = 7

6 if 7 != 3: --- True

7 res = 0 + 7 \* 1

res = 7

8 power = 1 \* 10

power = 10

9 number = 737 // 10

number = 73

4 while 73 > 0: --- True

5 cur\_digit = 73 % 10

cur\_digit = 3

6 if 3 != 3: --- False

9 number = 73 // 10

number = 7

4 while 7 > 0: --- True

5 cur\_digit = 7 % 10

cur\_digit = 7

6 if 7 != 3: --- True

7 res = 7 + 7 \* 10

res = 77

8 power = 10 \* 10

power = 100

9 number = 7 // 10

number = 0

4 while 0 > 0: --- False

10 return 77

Вариант: 1-2-28

1. gcd(-57, -66) = 3

1 def gcd(x=-57, y=-66)

2 if -57 < 0: --- True

3 x = --57

x = 57

4 if -66 < 0: --- True

5 y = --66

y = 66

6 while 66 != 0: --- True

7 rem = 57 % 66

rem = 57

8 x = 66

9 y = 57

6 while 57 != 0: --- True

7 rem = 66 % 57

rem = 9

8 x = 57

9 y = 9

6 while 9 != 0: --- True

7 rem = 57 % 9

rem = 3

8 x = 9

9 y = 3

6 while 3 != 0: --- True

7 rem = 9 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(0, 75) = 75

1 def gcd(x=0, y=75)

2 if 0 < 0: --- False

4 if 75 < 0: --- False

6 while 75 != 0: --- True

7 rem = 0 % 75

rem = 0

8 x = 75

9 y = 0

6 while 0 != 0: --- False

10 return 75

3. hex(183) = 'B7'

1 def hex(number=183)

2 if 183 == 0: --- False

4 res = ''

5 while 183 > 0: --- True

6 digit = 183 % 16

digit = 7

7 if 7 <= 9: --- True

8 digit = str(7)

digit = '7'

23 res = '7' + ''

res = '7'

24 number = 183 // 16

number = 11

5 while 11 > 0: --- True

6 digit = 11 % 16

digit = 11

7 if 11 <= 9: --- False

9 elif 11 <= 13: --- True

10 if 11 <= 11: --- True

11 if 11 == 10: --- False

13 else:

14 digit = 'B'

23 res = 'B' + '7'

res = 'B7'

24 number = 11 // 16

number = 0

5 while 0 > 0: --- False

25 return 'B7'

4. square\_equal(32, 48, 16) = '-1.0 and -0.5'

3 def square\_equal(a=32, b=48, c=16)

4 if 32 != 0: --- True

5 D = 48\*48 - 4\*32\*16

D = 256

6 if 256 > 0: --- True

7 x1 = (-48 - sqrt(256)) / (2\*32)

x1 = -1.0

8 x2 = (-48 + sqrt(256)) / (2\*32)

x2 = -0.5

9 return str(-1.0) + ' and ' + str(-0.5)

return '-1.0 and -0.5'

5. square\_equal(59, 80, 65) = 'no roots'

3 def square\_equal(a=59, b=80, c=65)

4 if 59 != 0: --- True

5 D = 80\*80 - 4\*59\*65

D = -8940

6 if -8940 > 0: --- False

10 elif -8940 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(36) = '2\*2\*3\*3'

1 def factorize(n=36)

2 res = ''

3 while 36 > 2 and 36 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 36 // 2

n = 18

3 while 18 > 2 and 18 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 18 // 2

n = 9

3 while 9 > 2 and 9 % 2 == 0: --- False

6 d = 3

7 while 9 > 3: --- True

8 if 9 % 3 == 0: --- True

9 res = '2\*2\*' + str(3) + '\*'

res = '2\*2\*3\*'

10 n = 9 // 3

n = 3

7 while 3 > 3: --- False

13 return '2\*2\*3\*' + str(3)

return '2\*2\*3\*3'

7. remove\_digit(598, 9) = 58

1 def remove\_digit(number=598, digit=9)

2 res = 0

3 power = 1

4 while 598 > 0: --- True

5 cur\_digit = 598 % 10

cur\_digit = 8

6 if 8 != 9: --- True

7 res = 0 + 8 \* 1

res = 8

8 power = 1 \* 10

power = 10

9 number = 598 // 10

number = 59

4 while 59 > 0: --- True

5 cur\_digit = 59 % 10

cur\_digit = 9

6 if 9 != 9: --- False

9 number = 59 // 10

number = 5

4 while 5 > 0: --- True

5 cur\_digit = 5 % 10

cur\_digit = 5

6 if 5 != 9: --- True

7 res = 8 + 5 \* 10

res = 58

8 power = 10 \* 10

power = 100

9 number = 5 // 10

number = 0

4 while 0 > 0: --- False

10 return 58

Вариант: 1-2-29

1. gcd(-68, 76) = 4

1 def gcd(x=-68, y=76)

2 if -68 < 0: --- True

3 x = --68

x = 68

4 if 76 < 0: --- False

6 while 76 != 0: --- True

7 rem = 68 % 76

rem = 68

8 x = 76

9 y = 68

6 while 68 != 0: --- True

7 rem = 76 % 68

rem = 8

8 x = 68

9 y = 8

6 while 8 != 0: --- True

7 rem = 68 % 8

rem = 4

8 x = 8

9 y = 4

6 while 4 != 0: --- True

7 rem = 8 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, -71) = 71

1 def gcd(x=0, y=-71)

2 if 0 < 0: --- False

4 if -71 < 0: --- True

5 y = --71

y = 71

6 while 71 != 0: --- True

7 rem = 0 % 71

rem = 0

8 x = 71

9 y = 0

6 while 0 != 0: --- False

10 return 71

3. hex(245) = 'F5'

1 def hex(number=245)

2 if 245 == 0: --- False

4 res = ''

5 while 245 > 0: --- True

6 digit = 245 % 16

digit = 5

7 if 5 <= 9: --- True

8 digit = str(5)

digit = '5'

23 res = '5' + ''

res = '5'

24 number = 245 // 16

number = 15

5 while 15 > 0: --- True

6 digit = 15 % 16

digit = 15

7 if 15 <= 9: --- False

9 elif 15 <= 13: --- False

19 elif 15 == 14: --- False

21 else:

22 digit = 'F'

23 res = 'F' + '5'

res = 'F5'

24 number = 15 // 16

number = 0

5 while 0 > 0: --- False

25 return 'F5'

4. square\_equal(-5, -58, 63) = '1.0 and -12.6'

3 def square\_equal(a=-5, b=-58, c=63)

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

5 D = -58\*-58 - 4\*-5\*63

D = 4624

6 if 4624 > 0: --- True

7 x1 = (--58 - sqrt(4624)) / (2\*-5)

x1 = 1.0

8 x2 = (--58 + sqrt(4624)) / (2\*-5)

x2 = -12.6

9 return str(1.0) + ' and ' + str(-12.6)

return '1.0 and -12.6'

5. square\_equal(29, -88, 83) = 'no roots'

3 def square\_equal(a=29, b=-88, c=83)

4 if 29 != 0: --- True

5 D = -88\*-88 - 4\*29\*83

D = -1884

6 if -1884 > 0: --- False

10 elif -1884 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(625) = '5\*5\*5\*5'

1 def factorize(n=625)

2 res = ''

3 while 625 > 2 and 625 % 2 == 0: --- False

6 d = 3

7 while 625 > 3: --- True

8 if 625 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 625 > 5: --- True

8 if 625 % 5 == 0: --- True

9 res = '' + str(5) + '\*'

res = '5\*'

10 n = 625 // 5

n = 125

7 while 125 > 5: --- True

8 if 125 % 5 == 0: --- True

9 res = '5\*' + str(5) + '\*'

res = '5\*5\*'

10 n = 125 // 5

n = 25

7 while 25 > 5: --- True

8 if 25 % 5 == 0: --- True

9 res = '5\*5\*' + str(5) + '\*'

res = '5\*5\*5\*'

10 n = 25 // 5

n = 5

7 while 5 > 5: --- False

13 return '5\*5\*5\*' + str(5)

return '5\*5\*5\*5'

7. remove\_digit(654, 5) = 64

1 def remove\_digit(number=654, digit=5)

2 res = 0

3 power = 1

4 while 654 > 0: --- True

5 cur\_digit = 654 % 10

cur\_digit = 4

6 if 4 != 5: --- True

7 res = 0 + 4 \* 1

res = 4

8 power = 1 \* 10

power = 10

9 number = 654 // 10

number = 65

4 while 65 > 0: --- True

5 cur\_digit = 65 % 10

cur\_digit = 5

6 if 5 != 5: --- False

9 number = 65 // 10

number = 6

4 while 6 > 0: --- True

5 cur\_digit = 6 % 10

cur\_digit = 6

6 if 6 != 5: --- True

7 res = 4 + 6 \* 10

res = 64

8 power = 10 \* 10

power = 100

9 number = 6 // 10

number = 0

4 while 0 > 0: --- False

10 return 64

Вариант: 1-2-30

1. gcd(15, 25) = 5

1 def gcd(x=15, y=25)

2 if 15 < 0: --- False

4 if 25 < 0: --- False

6 while 25 != 0: --- True

7 rem = 15 % 25

rem = 15

8 x = 25

9 y = 15

6 while 15 != 0: --- True

7 rem = 25 % 15

rem = 10

8 x = 15

9 y = 10

6 while 10 != 0: --- True

7 rem = 15 % 10

rem = 5

8 x = 10

9 y = 5

6 while 5 != 0: --- True

7 rem = 10 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

2. gcd(0, 10) = 10

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

2 if 0 < 0: --- False

4 if 10 < 0: --- False

6 while 10 != 0: --- True

7 rem = 0 % 10

rem = 0

8 x = 10

9 y = 0

6 while 0 != 0: --- False

10 return 10

3. hex(212) = 'D4'

1 def hex(number=212)

2 if 212 == 0: --- False

4 res = ''

5 while 212 > 0: --- True

6 digit = 212 % 16

digit = 4

7 if 4 <= 9: --- True

8 digit = str(4)

digit = '4'

23 res = '4' + ''

res = '4'

24 number = 212 // 16

number = 13

5 while 13 > 0: --- True

6 digit = 13 % 16

digit = 13

7 if 13 <= 9: --- False

9 elif 13 <= 13: --- True

10 if 13 <= 11: --- False

15 elif 13 == 12: --- False

17 else:

18 digit = 'D'

23 res = 'D' + '4'

res = 'D4'

24 number = 13 // 16

number = 0

5 while 0 > 0: --- False

25 return 'D4'

4. square\_equal(4, -31, -35) = '-1.0 and 8.75'

3 def square\_equal(a=4, b=-31, c=-35)

4 if 4 != 0: --- True

5 D = -31\*-31 - 4\*4\*-35

D = 1521

6 if 1521 > 0: --- True

7 x1 = (--31 - sqrt(1521)) / (2\*4)

x1 = -1.0

8 x2 = (--31 + sqrt(1521)) / (2\*4)

x2 = 8.75

9 return str(-1.0) + ' and ' + str(8.75)

return '-1.0 and 8.75'

5. square\_equal(54, 35, 48) = 'no roots'

3 def square\_equal(a=54, b=35, c=48)

4 if 54 != 0: --- True

5 D = 35\*35 - 4\*54\*48

D = -9143

6 if -9143 > 0: --- False

10 elif -9143 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(90) = '2\*3\*3\*5'

1 def factorize(n=90)

2 res = ''

3 while 90 > 2 and 90 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 90 // 2

n = 45

3 while 45 > 2 and 45 % 2 == 0: --- False

6 d = 3

7 while 45 > 3: --- True

8 if 45 % 3 == 0: --- True

9 res = '2\*' + str(3) + '\*'

res = '2\*3\*'

10 n = 45 // 3

n = 15

7 while 15 > 3: --- True

8 if 15 % 3 == 0: --- True

9 res = '2\*3\*' + str(3) + '\*'

res = '2\*3\*3\*'

10 n = 15 // 3

n = 5

7 while 5 > 3: --- True

8 if 5 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 5 > 5: --- False

13 return '2\*3\*3\*' + str(5)

return '2\*3\*3\*5'

7. remove\_digit(758, 5) = 78

1 def remove\_digit(number=758, digit=5)

2 res = 0

3 power = 1

4 while 758 > 0: --- True

5 cur\_digit = 758 % 10

cur\_digit = 8

6 if 8 != 5: --- True

7 res = 0 + 8 \* 1

res = 8

8 power = 1 \* 10

power = 10

9 number = 758 // 10

number = 75

4 while 75 > 0: --- True

5 cur\_digit = 75 % 10

cur\_digit = 5

6 if 5 != 5: --- False

9 number = 75 // 10

number = 7

4 while 7 > 0: --- True

5 cur\_digit = 7 % 10

cur\_digit = 7

6 if 7 != 5: --- True

7 res = 8 + 7 \* 10

res = 78

8 power = 10 \* 10

power = 100

9 number = 7 // 10

number = 0

4 while 0 > 0: --- False

10 return 78

Вариант: 1-2-31

1. gcd(-52, -44) = 4

1 def gcd(x=-52, y=-44)

2 if -52 < 0: --- True

3 x = --52

x = 52

4 if -44 < 0: --- True

5 y = --44

y = 44

6 while 44 != 0: --- True

7 rem = 52 % 44

rem = 8

8 x = 44

9 y = 8

6 while 8 != 0: --- True

7 rem = 44 % 8

rem = 4

8 x = 8

9 y = 4

6 while 4 != 0: --- True

7 rem = 8 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(69, 0) = 69

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

2 if 69 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 69

3. hex(164) = 'A4'

1 def hex(number=164)

2 if 164 == 0: --- False

4 res = ''

5 while 164 > 0: --- True

6 digit = 164 % 16

digit = 4

7 if 4 <= 9: --- True

8 digit = str(4)

digit = '4'

23 res = '4' + ''

res = '4'

24 number = 164 // 16

number = 10

5 while 10 > 0: --- True

6 digit = 10 % 16

digit = 10

7 if 10 <= 9: --- False

9 elif 10 <= 13: --- True

10 if 10 <= 11: --- True

11 if 10 == 10: --- True

12 digit = 'A'

23 res = 'A' + '4'

res = 'A4'

24 number = 10 // 16

number = 0

5 while 0 > 0: --- False

25 return 'A4'

4. square\_equal(-50, -59, -17) = '-0.5 and -0.68'

3 def square\_equal(a=-50, b=-59, c=-17)

4 if -50 != 0: --- True

5 D = -59\*-59 - 4\*-50\*-17

D = 81

6 if 81 > 0: --- True

7 x1 = (--59 - sqrt(81)) / (2\*-50)

x1 = -0.5

8 x2 = (--59 + sqrt(81)) / (2\*-50)

x2 = -0.68

9 return str(-0.5) + ' and ' + str(-0.68)

return '-0.5 and -0.68'

5. square\_equal(-95, -16, -6) = 'no roots'

3 def square\_equal(a=-95, b=-16, c=-6)

4 if -95 != 0: --- True

5 D = -16\*-16 - 4\*-95\*-6

D = -2024

6 if -2024 > 0: --- False

10 elif -2024 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(63) = '3\*3\*7'

1 def factorize(n=63)

2 res = ''

3 while 63 > 2 and 63 % 2 == 0: --- False

6 d = 3

7 while 63 > 3: --- True

8 if 63 % 3 == 0: --- True

9 res = '' + str(3) + '\*'

res = '3\*'

10 n = 63 // 3

n = 21

7 while 21 > 3: --- True

8 if 21 % 3 == 0: --- True

9 res = '3\*' + str(3) + '\*'

res = '3\*3\*'

10 n = 21 // 3

n = 7

7 while 7 > 3: --- True

8 if 7 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 7 > 5: --- True

8 if 7 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 7 > 7: --- False

13 return '3\*3\*' + str(7)

return '3\*3\*7'

7. remove\_digit(628, 8) = 62

1 def remove\_digit(number=628, digit=8)

2 res = 0

3 power = 1

4 while 628 > 0: --- True

5 cur\_digit = 628 % 10

cur\_digit = 8

6 if 8 != 8: --- False

9 number = 628 // 10

number = 62

4 while 62 > 0: --- True

5 cur\_digit = 62 % 10

cur\_digit = 2

6 if 2 != 8: --- True

7 res = 0 + 2 \* 1

res = 2

8 power = 1 \* 10

power = 10

9 number = 62 // 10

number = 6

4 while 6 > 0: --- True

5 cur\_digit = 6 % 10

cur\_digit = 6

6 if 6 != 8: --- True

7 res = 2 + 6 \* 10

res = 62

8 power = 10 \* 10

power = 100

9 number = 6 // 10

number = 0

4 while 0 > 0: --- False

10 return 62

Вариант: 1-2-32

1. gcd(-12, -52) = 4

1 def gcd(x=-12, y=-52)

2 if -12 < 0: --- True

3 x = --12

x = 12

4 if -52 < 0: --- True

5 y = --52

y = 52

6 while 52 != 0: --- True

7 rem = 12 % 52

rem = 12

8 x = 52

9 y = 12

6 while 12 != 0: --- True

7 rem = 52 % 12

rem = 4

8 x = 12

9 y = 4

6 while 4 != 0: --- True

7 rem = 12 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, 83) = 83

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

2 if 0 < 0: --- False

4 if 83 < 0: --- False

6 while 83 != 0: --- True

7 rem = 0 % 83

rem = 0

8 x = 83

9 y = 0

6 while 0 != 0: --- False

10 return 83

3. hex(188) = 'BC'

1 def hex(number=188)

2 if 188 == 0: --- False

4 res = ''

5 while 188 > 0: --- True

6 digit = 188 % 16

digit = 12

7 if 12 <= 9: --- False

9 elif 12 <= 13: --- True

10 if 12 <= 11: --- False

15 elif 12 == 12: --- True

16 digit = 'C'

23 res = 'C' + ''

res = 'C'

24 number = 188 // 16

number = 11

5 while 11 > 0: --- True

6 digit = 11 % 16

digit = 11

7 if 11 <= 9: --- False

9 elif 11 <= 13: --- True

10 if 11 <= 11: --- True

11 if 11 == 10: --- False

13 else:

14 digit = 'B'

23 res = 'B' + 'C'

res = 'BC'

24 number = 11 // 16

number = 0

5 while 0 > 0: --- False

25 return 'BC'

4. square\_equal(0, 40, 34) = '-0.85'

3 def square\_equal(a=0, b=40, c=34)

4 if 0 != 0: --- False

14 else:

15 if 40 != 0: --- True

16 return str(-34 / 40)

return '-0.85'

5. square\_equal(18, -59, 62) = 'no roots'

3 def square\_equal(a=18, b=-59, c=62)

4 if 18 != 0: --- True

5 D = -59\*-59 - 4\*18\*62

D = -983

6 if -983 > 0: --- False

10 elif -983 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(48) = '2\*2\*2\*2\*3'

1 def factorize(n=48)

2 res = ''

3 while 48 > 2 and 48 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 48 // 2

n = 24

3 while 24 > 2 and 24 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 24 // 2

n = 12

3 while 12 > 2 and 12 % 2 == 0: --- True

4 res = '2\*2\*' + '2\*'

res = '2\*2\*2\*'

5 n = 12 // 2

n = 6

3 while 6 > 2 and 6 % 2 == 0: --- True

4 res = '2\*2\*2\*' + '2\*'

res = '2\*2\*2\*2\*'

5 n = 6 // 2

n = 3

3 while 3 > 2 and 3 % 2 == 0: --- False

6 d = 3

7 while 3 > 3: --- False

13 return '2\*2\*2\*2\*' + str(3)

return '2\*2\*2\*2\*3'

7. remove\_digit(684, 4) = 68

1 def remove\_digit(number=684, digit=4)

2 res = 0

3 power = 1

4 while 684 > 0: --- True

5 cur\_digit = 684 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 684 // 10

number = 68

4 while 68 > 0: --- True

5 cur\_digit = 68 % 10

cur\_digit = 8

6 if 8 != 4: --- True

7 res = 0 + 8 \* 1

res = 8

8 power = 1 \* 10

power = 10

9 number = 68 // 10

number = 6

4 while 6 > 0: --- True

5 cur\_digit = 6 % 10

cur\_digit = 6

6 if 6 != 4: --- True

7 res = 8 + 6 \* 10

res = 68

8 power = 10 \* 10

power = 100

9 number = 6 // 10

number = 0

4 while 0 > 0: --- False

10 return 68

Вариант: 1-2-33

1. gcd(-28, 68) = 4

1 def gcd(x=-28, y=68)

2 if -28 < 0: --- True

3 x = --28

x = 28

4 if 68 < 0: --- False

6 while 68 != 0: --- True

7 rem = 28 % 68

rem = 28

8 x = 68

9 y = 28

6 while 28 != 0: --- True

7 rem = 68 % 28

rem = 12

8 x = 28

9 y = 12

6 while 12 != 0: --- True

7 rem = 28 % 12

rem = 4

8 x = 12

9 y = 4

6 while 4 != 0: --- True

7 rem = 12 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, 59) = 59

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

2 if 0 < 0: --- False

4 if 59 < 0: --- False

6 while 59 != 0: --- True

7 rem = 0 % 59

rem = 0

8 x = 59

9 y = 0

6 while 0 != 0: --- False

10 return 59

3. hex(205) = 'CD'

1 def hex(number=205)

2 if 205 == 0: --- False

4 res = ''

5 while 205 > 0: --- True

6 digit = 205 % 16

digit = 13

7 if 13 <= 9: --- False

9 elif 13 <= 13: --- True

10 if 13 <= 11: --- False

15 elif 13 == 12: --- False

17 else:

18 digit = 'D'

23 res = 'D' + ''

res = 'D'

24 number = 205 // 16

number = 12

5 while 12 > 0: --- True

6 digit = 12 % 16

digit = 12

7 if 12 <= 9: --- False

9 elif 12 <= 13: --- True

10 if 12 <= 11: --- False

15 elif 12 == 12: --- True

16 digit = 'C'

23 res = 'C' + 'D'

res = 'CD'

24 number = 12 // 16

number = 0

5 while 0 > 0: --- False

25 return 'CD'

4. square\_equal(10, -51, -91) = '-1.4 and 6.5'

3 def square\_equal(a=10, b=-51, c=-91)

4 if 10 != 0: --- True

5 D = -51\*-51 - 4\*10\*-91

D = 6241

6 if 6241 > 0: --- True

7 x1 = (--51 - sqrt(6241)) / (2\*10)

x1 = -1.4

8 x2 = (--51 + sqrt(6241)) / (2\*10)

x2 = 6.5

9 return str(-1.4) + ' and ' + str(6.5)

return '-1.4 and 6.5'

5. square\_equal(4, 7, 80) = 'no roots'

3 def square\_equal(a=4, b=7, c=80)

4 if 4 != 0: --- True

5 D = 7\*7 - 4\*4\*80

D = -1231

6 if -1231 > 0: --- False

10 elif -1231 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(33) = '3\*11'

1 def factorize(n=33)

2 res = ''

3 while 33 > 2 and 33 % 2 == 0: --- False

6 d = 3

7 while 33 > 3: --- True

8 if 33 % 3 == 0: --- True

9 res = '' + str(3) + '\*'

res = '3\*'

10 n = 33 // 3

n = 11

7 while 11 > 3: --- True

8 if 11 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 11 > 5: --- True

8 if 11 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 11 > 7: --- True

8 if 11 % 7 == 0: --- False

11 else:

12 d = 7 + 2

d = 9

7 while 11 > 9: --- True

8 if 11 % 9 == 0: --- False

11 else:

12 d = 9 + 2

d = 11

7 while 11 > 11: --- False

13 return '3\*' + str(11)

return '3\*11'

7. remove\_digit(342, 2) = 34

1 def remove\_digit(number=342, digit=2)

2 res = 0

3 power = 1

4 while 342 > 0: --- True

5 cur\_digit = 342 % 10

cur\_digit = 2

6 if 2 != 2: --- False

9 number = 342 // 10

number = 34

4 while 34 > 0: --- True

5 cur\_digit = 34 % 10

cur\_digit = 4

6 if 4 != 2: --- True

7 res = 0 + 4 \* 1

res = 4

8 power = 1 \* 10

power = 10

9 number = 34 // 10

number = 3

4 while 3 > 0: --- True

5 cur\_digit = 3 % 10

cur\_digit = 3

6 if 3 != 2: --- True

7 res = 4 + 3 \* 10

res = 34

8 power = 10 \* 10

power = 100

9 number = 3 // 10

number = 0

4 while 0 > 0: --- False

10 return 34

Вариант: 1-2-34

1. gcd(-76, -40) = 4

1 def gcd(x=-76, y=-40)

2 if -76 < 0: --- True

3 x = --76

x = 76

4 if -40 < 0: --- True

5 y = --40

y = 40

6 while 40 != 0: --- True

7 rem = 76 % 40

rem = 36

8 x = 40

9 y = 36

6 while 36 != 0: --- True

7 rem = 40 % 36

rem = 4

8 x = 36

9 y = 4

6 while 4 != 0: --- True

7 rem = 36 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, -88) = 88

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

2 if 0 < 0: --- False

4 if -88 < 0: --- True

5 y = --88

y = 88

6 while 88 != 0: --- True

7 rem = 0 % 88

rem = 0

8 x = 88

9 y = 0

6 while 0 != 0: --- False

10 return 88

3. hex(240) = 'F0'

1 def hex(number=240)

2 if 240 == 0: --- False

4 res = ''

5 while 240 > 0: --- True

6 digit = 240 % 16

digit = 0

7 if 0 <= 9: --- True

8 digit = str(0)

digit = '0'

23 res = '0' + ''

res = '0'

24 number = 240 // 16

number = 15

5 while 15 > 0: --- True

6 digit = 15 % 16

digit = 15

7 if 15 <= 9: --- False

9 elif 15 <= 13: --- False

19 elif 15 == 14: --- False

21 else:

22 digit = 'F'

23 res = 'F' + '0'

res = 'F0'

24 number = 15 // 16

number = 0

5 while 0 > 0: --- False

25 return 'F0'

4. square\_equal(-40, -82, -39) = '-0.75 and -1.3'

3 def square\_equal(a=-40, b=-82, c=-39)

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

5 D = -82\*-82 - 4\*-40\*-39

D = 484

6 if 484 > 0: --- True

7 x1 = (--82 - sqrt(484)) / (2\*-40)

x1 = -0.75

8 x2 = (--82 + sqrt(484)) / (2\*-40)

x2 = -1.3

9 return str(-0.75) + ' and ' + str(-1.3)

return '-0.75 and -1.3'

5. square\_equal(-72, -28, -15) = 'no roots'

3 def square\_equal(a=-72, b=-28, c=-15)

4 if -72 != 0: --- True

5 D = -28\*-28 - 4\*-72\*-15

D = -3536

6 if -3536 > 0: --- False

10 elif -3536 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(60) = '2\*2\*3\*5'

1 def factorize(n=60)

2 res = ''

3 while 60 > 2 and 60 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 60 // 2

n = 30

3 while 30 > 2 and 30 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 30 // 2

n = 15

3 while 15 > 2 and 15 % 2 == 0: --- False

6 d = 3

7 while 15 > 3: --- True

8 if 15 % 3 == 0: --- True

9 res = '2\*2\*' + str(3) + '\*'

res = '2\*2\*3\*'

10 n = 15 // 3

n = 5

7 while 5 > 3: --- True

8 if 5 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 5 > 5: --- False

13 return '2\*2\*3\*' + str(5)

return '2\*2\*3\*5'

7. remove\_digit(217, 7) = 21

1 def remove\_digit(number=217, digit=7)

2 res = 0

3 power = 1

4 while 217 > 0: --- True

5 cur\_digit = 217 % 10

cur\_digit = 7

6 if 7 != 7: --- False

9 number = 217 // 10

number = 21

4 while 21 > 0: --- True

5 cur\_digit = 21 % 10

cur\_digit = 1

6 if 1 != 7: --- True

7 res = 0 + 1 \* 1

res = 1

8 power = 1 \* 10

power = 10

9 number = 21 // 10

number = 2

4 while 2 > 0: --- True

5 cur\_digit = 2 % 10

cur\_digit = 2

6 if 2 != 7: --- True

7 res = 1 + 2 \* 10

res = 21

8 power = 10 \* 10

power = 100

9 number = 2 // 10

number = 0

4 while 0 > 0: --- False

10 return 21

Вариант: 1-2-35

1. gcd(-21, -30) = 3

1 def gcd(x=-21, y=-30)

2 if -21 < 0: --- True

3 x = --21

x = 21

4 if -30 < 0: --- True

5 y = --30

y = 30

6 while 30 != 0: --- True

7 rem = 21 % 30

rem = 21

8 x = 30

9 y = 21

6 while 21 != 0: --- True

7 rem = 30 % 21

rem = 9

8 x = 21

9 y = 9

6 while 9 != 0: --- True

7 rem = 21 % 9

rem = 3

8 x = 9

9 y = 3

6 while 3 != 0: --- True

7 rem = 9 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(25, 0) = 25

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

2 if 25 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 25

3. hex(210) = 'D2'

1 def hex(number=210)

2 if 210 == 0: --- False

4 res = ''

5 while 210 > 0: --- True

6 digit = 210 % 16

digit = 2

7 if 2 <= 9: --- True

8 digit = str(2)

digit = '2'

23 res = '2' + ''

res = '2'

24 number = 210 // 16

number = 13

5 while 13 > 0: --- True

6 digit = 13 % 16

digit = 13

7 if 13 <= 9: --- False

9 elif 13 <= 13: --- True

10 if 13 <= 11: --- False

15 elif 13 == 12: --- False

17 else:

18 digit = 'D'

23 res = 'D' + '2'

res = 'D2'

24 number = 13 // 16

number = 0

5 while 0 > 0: --- False

25 return 'D2'

4. square\_equal(-5, 1, 4) = '1.0 and -0.8'

3 def square\_equal(a=-5, b=1, c=4)

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

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

D = 81

6 if 81 > 0: --- True

7 x1 = (-1 - sqrt(81)) / (2\*-5)

x1 = 1.0

8 x2 = (-1 + sqrt(81)) / (2\*-5)

x2 = -0.8

9 return str(1.0) + ' and ' + str(-0.8)

return '1.0 and -0.8'

5. square\_equal(-35, 6, -26) = 'no roots'

3 def square\_equal(a=-35, b=6, c=-26)

4 if -35 != 0: --- True

5 D = 6\*6 - 4\*-35\*-26

D = -3604

6 if -3604 > 0: --- False

10 elif -3604 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(77) = '7\*11'

1 def factorize(n=77)

2 res = ''

3 while 77 > 2 and 77 % 2 == 0: --- False

6 d = 3

7 while 77 > 3: --- True

8 if 77 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 77 > 5: --- True

8 if 77 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 77 > 7: --- True

8 if 77 % 7 == 0: --- True

9 res = '' + str(7) + '\*'

res = '7\*'

10 n = 77 // 7

n = 11

7 while 11 > 7: --- True

8 if 11 % 7 == 0: --- False

11 else:

12 d = 7 + 2

d = 9

7 while 11 > 9: --- True

8 if 11 % 9 == 0: --- False

11 else:

12 d = 9 + 2

d = 11

7 while 11 > 11: --- False

13 return '7\*' + str(11)

return '7\*11'

7. remove\_digit(7666, 6) = 7

1 def remove\_digit(number=7666, digit=6)

2 res = 0

3 power = 1

4 while 7666 > 0: --- True

5 cur\_digit = 7666 % 10

cur\_digit = 6

6 if 6 != 6: --- False

9 number = 7666 // 10

number = 766

4 while 766 > 0: --- True

5 cur\_digit = 766 % 10

cur\_digit = 6

6 if 6 != 6: --- False

9 number = 766 // 10

number = 76

4 while 76 > 0: --- True

5 cur\_digit = 76 % 10

cur\_digit = 6

6 if 6 != 6: --- False

9 number = 76 // 10

number = 7

4 while 7 > 0: --- True

5 cur\_digit = 7 % 10

cur\_digit = 7

6 if 7 != 6: --- True

7 res = 0 + 7 \* 1

res = 7

8 power = 1 \* 10

power = 10

9 number = 7 // 10

number = 0

4 while 0 > 0: --- False

10 return 7

Вариант: 1-2-36

1. gcd(-95, -75) = 5

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

2 if -95 < 0: --- True

3 x = --95

x = 95

4 if -75 < 0: --- True

5 y = --75

y = 75

6 while 75 != 0: --- True

7 rem = 95 % 75

rem = 20

8 x = 75

9 y = 20

6 while 20 != 0: --- True

7 rem = 75 % 20

rem = 15

8 x = 20

9 y = 15

6 while 15 != 0: --- True

7 rem = 20 % 15

rem = 5

8 x = 15

9 y = 5

6 while 5 != 0: --- True

7 rem = 15 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

2. gcd(82, 0) = 82

1 def gcd(x=82, y=0)

2 if 82 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 82

3. hex(255) = 'FF'

1 def hex(number=255)

2 if 255 == 0: --- False

4 res = ''

5 while 255 > 0: --- True

6 digit = 255 % 16

digit = 15

7 if 15 <= 9: --- False

9 elif 15 <= 13: --- False

19 elif 15 == 14: --- False

21 else:

22 digit = 'F'

23 res = 'F' + ''

res = 'F'

24 number = 255 // 16

number = 15

5 while 15 > 0: --- True

6 digit = 15 % 16

digit = 15

7 if 15 <= 9: --- False

9 elif 15 <= 13: --- False

19 elif 15 == 14: --- False

21 else:

22 digit = 'F'

23 res = 'F' + 'F'

res = 'FF'

24 number = 15 // 16

number = 0

5 while 0 > 0: --- False

25 return 'FF'

4. square\_equal(16, -64, 48) = '1.0 and 3.0'

3 def square\_equal(a=16, b=-64, c=48)

4 if 16 != 0: --- True

5 D = -64\*-64 - 4\*16\*48

D = 1024

6 if 1024 > 0: --- True

7 x1 = (--64 - sqrt(1024)) / (2\*16)

x1 = 1.0

8 x2 = (--64 + sqrt(1024)) / (2\*16)

x2 = 3.0

9 return str(1.0) + ' and ' + str(3.0)

return '1.0 and 3.0'

5. square\_equal(-50, -90, -82) = 'no roots'

3 def square\_equal(a=-50, b=-90, c=-82)

4 if -50 != 0: --- True

5 D = -90\*-90 - 4\*-50\*-82

D = -8300

6 if -8300 > 0: --- False

10 elif -8300 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(22) = '2\*11'

1 def factorize(n=22)

2 res = ''

3 while 22 > 2 and 22 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 22 // 2

n = 11

3 while 11 > 2 and 11 % 2 == 0: --- False

6 d = 3

7 while 11 > 3: --- True

8 if 11 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 11 > 5: --- True

8 if 11 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 11 > 7: --- True

8 if 11 % 7 == 0: --- False

11 else:

12 d = 7 + 2

d = 9

7 while 11 > 9: --- True

8 if 11 % 9 == 0: --- False

11 else:

12 d = 9 + 2

d = 11

7 while 11 > 11: --- False

13 return '2\*' + str(11)

return '2\*11'

7. remove\_digit(1666, 6) = 1

1 def remove\_digit(number=1666, digit=6)

2 res = 0

3 power = 1

4 while 1666 > 0: --- True

5 cur\_digit = 1666 % 10

cur\_digit = 6

6 if 6 != 6: --- False

9 number = 1666 // 10

number = 166

4 while 166 > 0: --- True

5 cur\_digit = 166 % 10

cur\_digit = 6

6 if 6 != 6: --- False

9 number = 166 // 10

number = 16

4 while 16 > 0: --- True

5 cur\_digit = 16 % 10

cur\_digit = 6

6 if 6 != 6: --- False

9 number = 16 // 10

number = 1

4 while 1 > 0: --- True

5 cur\_digit = 1 % 10

cur\_digit = 1

6 if 1 != 6: --- True

7 res = 0 + 1 \* 1

res = 1

8 power = 1 \* 10

power = 10

9 number = 1 // 10

number = 0

4 while 0 > 0: --- False

10 return 1

Вариант: 1-2-37

1. gcd(-66, 87) = 3

1 def gcd(x=-66, y=87)

2 if -66 < 0: --- True

3 x = --66

x = 66

4 if 87 < 0: --- False

6 while 87 != 0: --- True

7 rem = 66 % 87

rem = 66

8 x = 87

9 y = 66

6 while 66 != 0: --- True

7 rem = 87 % 66

rem = 21

8 x = 66

9 y = 21

6 while 21 != 0: --- True

7 rem = 66 % 21

rem = 3

8 x = 21

9 y = 3

6 while 3 != 0: --- True

7 rem = 21 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(44, 0) = 44

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

2 if 44 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 44

3. hex(229) = 'E5'

1 def hex(number=229)

2 if 229 == 0: --- False

4 res = ''

5 while 229 > 0: --- True

6 digit = 229 % 16

digit = 5

7 if 5 <= 9: --- True

8 digit = str(5)

digit = '5'

23 res = '5' + ''

res = '5'

24 number = 229 // 16

number = 14

5 while 14 > 0: --- True

6 digit = 14 % 16

digit = 14

7 if 14 <= 9: --- False

9 elif 14 <= 13: --- False

19 elif 14 == 14: --- True

20 digit = 'E'

23 res = 'E' + '5'

res = 'E5'

24 number = 14 // 16

number = 0

5 while 0 > 0: --- False

25 return 'E5'

4. square\_equal(-20, 38, -14) = '1.4 and 0.5'

3 def square\_equal(a=-20, b=38, c=-14)

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

5 D = 38\*38 - 4\*-20\*-14

D = 324

6 if 324 > 0: --- True

7 x1 = (-38 - sqrt(324)) / (2\*-20)

x1 = 1.4

8 x2 = (-38 + sqrt(324)) / (2\*-20)

x2 = 0.5

9 return str(1.4) + ' and ' + str(0.5)

return '1.4 and 0.5'

5. square\_equal(-83, -54, -9) = 'no roots'

3 def square\_equal(a=-83, b=-54, c=-9)

4 if -83 != 0: --- True

5 D = -54\*-54 - 4\*-83\*-9

D = -72

6 if -72 > 0: --- False

10 elif -72 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(40) = '2\*2\*2\*5'

1 def factorize(n=40)

2 res = ''

3 while 40 > 2 and 40 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 40 // 2

n = 20

3 while 20 > 2 and 20 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 20 // 2

n = 10

3 while 10 > 2 and 10 % 2 == 0: --- True

4 res = '2\*2\*' + '2\*'

res = '2\*2\*2\*'

5 n = 10 // 2

n = 5

3 while 5 > 2 and 5 % 2 == 0: --- False

6 d = 3

7 while 5 > 3: --- True

8 if 5 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 5 > 5: --- False

13 return '2\*2\*2\*' + str(5)

return '2\*2\*2\*5'

7. remove\_digit(7000, 0) = 7

1 def remove\_digit(number=7000, digit=0)

2 res = 0

3 power = 1

4 while 7000 > 0: --- True

5 cur\_digit = 7000 % 10

cur\_digit = 0

6 if 0 != 0: --- False

9 number = 7000 // 10

number = 700

4 while 700 > 0: --- True

5 cur\_digit = 700 % 10

cur\_digit = 0

6 if 0 != 0: --- False

9 number = 700 // 10

number = 70

4 while 70 > 0: --- True

5 cur\_digit = 70 % 10

cur\_digit = 0

6 if 0 != 0: --- False

9 number = 70 // 10

number = 7

4 while 7 > 0: --- True

5 cur\_digit = 7 % 10

cur\_digit = 7

6 if 7 != 0: --- True

7 res = 0 + 7 \* 1

res = 7

8 power = 1 \* 10

power = 10

9 number = 7 // 10

number = 0

4 while 0 > 0: --- False

10 return 7

Вариант: 1-2-38

1. gcd(66, 75) = 3

1 def gcd(x=66, y=75)

2 if 66 < 0: --- False

4 if 75 < 0: --- False

6 while 75 != 0: --- True

7 rem = 66 % 75

rem = 66

8 x = 75

9 y = 66

6 while 66 != 0: --- True

7 rem = 75 % 66

rem = 9

8 x = 66

9 y = 9

6 while 9 != 0: --- True

7 rem = 66 % 9

rem = 3

8 x = 9

9 y = 3

6 while 3 != 0: --- True

7 rem = 9 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(-45, 0) = 45

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

2 if -45 < 0: --- True

3 x = --45

x = 45

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 45

3. hex(177) = 'B1'

1 def hex(number=177)

2 if 177 == 0: --- False

4 res = ''

5 while 177 > 0: --- True

6 digit = 177 % 16

digit = 1

7 if 1 <= 9: --- True

8 digit = str(1)

digit = '1'

23 res = '1' + ''

res = '1'

24 number = 177 // 16

number = 11

5 while 11 > 0: --- True

6 digit = 11 % 16

digit = 11

7 if 11 <= 9: --- False

9 elif 11 <= 13: --- True

10 if 11 <= 11: --- True

11 if 11 == 10: --- False

13 else:

14 digit = 'B'

23 res = 'B' + '1'

res = 'B1'

24 number = 11 // 16

number = 0

5 while 0 > 0: --- False

25 return 'B1'

4. square\_equal(-25, -30, -8) = '-0.4 and -0.8'

3 def square\_equal(a=-25, b=-30, c=-8)

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

5 D = -30\*-30 - 4\*-25\*-8

D = 100

6 if 100 > 0: --- True

7 x1 = (--30 - sqrt(100)) / (2\*-25)

x1 = -0.4

8 x2 = (--30 + sqrt(100)) / (2\*-25)

x2 = -0.8

9 return str(-0.4) + ' and ' + str(-0.8)

return '-0.4 and -0.8'

5. square\_equal(27, 9, 88) = 'no roots'

3 def square\_equal(a=27, b=9, c=88)

4 if 27 != 0: --- True

5 D = 9\*9 - 4\*27\*88

D = -9423

6 if -9423 > 0: --- False

10 elif -9423 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(225) = '3\*3\*5\*5'

1 def factorize(n=225)

2 res = ''

3 while 225 > 2 and 225 % 2 == 0: --- False

6 d = 3

7 while 225 > 3: --- True

8 if 225 % 3 == 0: --- True

9 res = '' + str(3) + '\*'

res = '3\*'

10 n = 225 // 3

n = 75

7 while 75 > 3: --- True

8 if 75 % 3 == 0: --- True

9 res = '3\*' + str(3) + '\*'

res = '3\*3\*'

10 n = 75 // 3

n = 25

7 while 25 > 3: --- True

8 if 25 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 25 > 5: --- True

8 if 25 % 5 == 0: --- True

9 res = '3\*3\*' + str(5) + '\*'

res = '3\*3\*5\*'

10 n = 25 // 5

n = 5

7 while 5 > 5: --- False

13 return '3\*3\*5\*' + str(5)

return '3\*3\*5\*5'

7. remove\_digit(141, 4) = 11

1 def remove\_digit(number=141, digit=4)

2 res = 0

3 power = 1

4 while 141 > 0: --- True

5 cur\_digit = 141 % 10

cur\_digit = 1

6 if 1 != 4: --- True

7 res = 0 + 1 \* 1

res = 1

8 power = 1 \* 10

power = 10

9 number = 141 // 10

number = 14

4 while 14 > 0: --- True

5 cur\_digit = 14 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 14 // 10

number = 1

4 while 1 > 0: --- True

5 cur\_digit = 1 % 10

cur\_digit = 1

6 if 1 != 4: --- True

7 res = 1 + 1 \* 10

res = 11

8 power = 10 \* 10

power = 100

9 number = 1 // 10

number = 0

4 while 0 > 0: --- False

10 return 11

Вариант: 1-2-39

1. gcd(99, -69) = 3

1 def gcd(x=99, y=-69)

2 if 99 < 0: --- False

4 if -69 < 0: --- True

5 y = --69

y = 69

6 while 69 != 0: --- True

7 rem = 99 % 69

rem = 30

8 x = 69

9 y = 30

6 while 30 != 0: --- True

7 rem = 69 % 30

rem = 9

8 x = 30

9 y = 9

6 while 9 != 0: --- True

7 rem = 30 % 9

rem = 3

8 x = 9

9 y = 3

6 while 3 != 0: --- True

7 rem = 9 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(-73, 0) = 73

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

2 if -73 < 0: --- True

3 x = --73

x = 73

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 73

3. hex(184) = 'B8'

1 def hex(number=184)

2 if 184 == 0: --- False

4 res = ''

5 while 184 > 0: --- True

6 digit = 184 % 16

digit = 8

7 if 8 <= 9: --- True

8 digit = str(8)

digit = '8'

23 res = '8' + ''

res = '8'

24 number = 184 // 16

number = 11

5 while 11 > 0: --- True

6 digit = 11 % 16

digit = 11

7 if 11 <= 9: --- False

9 elif 11 <= 13: --- True

10 if 11 <= 11: --- True

11 if 11 == 10: --- False

13 else:

14 digit = 'B'

23 res = 'B' + '8'

res = 'B8'

24 number = 11 // 16

number = 0

5 while 0 > 0: --- False

25 return 'B8'

4. square\_equal(-72, 90, -18) = '1.0 and 0.25'

3 def square\_equal(a=-72, b=90, c=-18)

4 if -72 != 0: --- True

5 D = 90\*90 - 4\*-72\*-18

D = 2916

6 if 2916 > 0: --- True

7 x1 = (-90 - sqrt(2916)) / (2\*-72)

x1 = 1.0

8 x2 = (-90 + sqrt(2916)) / (2\*-72)

x2 = 0.25

9 return str(1.0) + ' and ' + str(0.25)

return '1.0 and 0.25'

5. square\_equal(-95, 99, -36) = 'no roots'

3 def square\_equal(a=-95, b=99, c=-36)

4 if -95 != 0: --- True

5 D = 99\*99 - 4\*-95\*-36

D = -3879

6 if -3879 > 0: --- False

10 elif -3879 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(98) = '2\*7\*7'

1 def factorize(n=98)

2 res = ''

3 while 98 > 2 and 98 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 98 // 2

n = 49

3 while 49 > 2 and 49 % 2 == 0: --- False

6 d = 3

7 while 49 > 3: --- True

8 if 49 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 49 > 5: --- True

8 if 49 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 49 > 7: --- True

8 if 49 % 7 == 0: --- True

9 res = '2\*' + str(7) + '\*'

res = '2\*7\*'

10 n = 49 // 7

n = 7

7 while 7 > 7: --- False

13 return '2\*7\*' + str(7)

return '2\*7\*7'

7. remove\_digit(748, 4) = 78

1 def remove\_digit(number=748, digit=4)

2 res = 0

3 power = 1

4 while 748 > 0: --- True

5 cur\_digit = 748 % 10

cur\_digit = 8

6 if 8 != 4: --- True

7 res = 0 + 8 \* 1

res = 8

8 power = 1 \* 10

power = 10

9 number = 748 // 10

number = 74

4 while 74 > 0: --- True

5 cur\_digit = 74 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 74 // 10

number = 7

4 while 7 > 0: --- True

5 cur\_digit = 7 % 10

cur\_digit = 7

6 if 7 != 4: --- True

7 res = 8 + 7 \* 10

res = 78

8 power = 10 \* 10

power = 100

9 number = 7 // 10

number = 0

4 while 0 > 0: --- False

10 return 78

Вариант: 1-2-40

1. gcd(48, 54) = 6

1 def gcd(x=48, y=54)

2 if 48 < 0: --- False

4 if 54 < 0: --- False

6 while 54 != 0: --- True

7 rem = 48 % 54

rem = 48

8 x = 54

9 y = 48

6 while 48 != 0: --- True

7 rem = 54 % 48

rem = 6

8 x = 48

9 y = 6

6 while 6 != 0: --- True

7 rem = 48 % 6

rem = 0

8 x = 6

9 y = 0

6 while 0 != 0: --- False

10 return 6

2. gcd(-85, 0) = 85

1 def gcd(x=-85, y=0)

2 if -85 < 0: --- True

3 x = --85

x = 85

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 85

3. hex(241) = 'F1'

1 def hex(number=241)

2 if 241 == 0: --- False

4 res = ''

5 while 241 > 0: --- True

6 digit = 241 % 16

digit = 1

7 if 1 <= 9: --- True

8 digit = str(1)

digit = '1'

23 res = '1' + ''

res = '1'

24 number = 241 // 16

number = 15

5 while 15 > 0: --- True

6 digit = 15 % 16

digit = 15

7 if 15 <= 9: --- False

9 elif 15 <= 13: --- False

19 elif 15 == 14: --- False

21 else:

22 digit = 'F'

23 res = 'F' + '1'

res = 'F1'

24 number = 15 // 16

number = 0

5 while 0 > 0: --- False

25 return 'F1'

4. square\_equal(-40, 94, -9) = '2.25 and 0.1'

3 def square\_equal(a=-40, b=94, c=-9)

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

5 D = 94\*94 - 4\*-40\*-9

D = 7396

6 if 7396 > 0: --- True

7 x1 = (-94 - sqrt(7396)) / (2\*-40)

x1 = 2.25

8 x2 = (-94 + sqrt(7396)) / (2\*-40)

x2 = 0.1

9 return str(2.25) + ' and ' + str(0.1)

return '2.25 and 0.1'

5. square\_equal(15, -47, 78) = 'no roots'

3 def square\_equal(a=15, b=-47, c=78)

4 if 15 != 0: --- True

5 D = -47\*-47 - 4\*15\*78

D = -2471

6 if -2471 > 0: --- False

10 elif -2471 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(80) = '2\*2\*2\*2\*5'

1 def factorize(n=80)

2 res = ''

3 while 80 > 2 and 80 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 80 // 2

n = 40

3 while 40 > 2 and 40 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 40 // 2

n = 20

3 while 20 > 2 and 20 % 2 == 0: --- True

4 res = '2\*2\*' + '2\*'

res = '2\*2\*2\*'

5 n = 20 // 2

n = 10

3 while 10 > 2 and 10 % 2 == 0: --- True

4 res = '2\*2\*2\*' + '2\*'

res = '2\*2\*2\*2\*'

5 n = 10 // 2

n = 5

3 while 5 > 2 and 5 % 2 == 0: --- False

6 d = 3

7 while 5 > 3: --- True

8 if 5 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 5 > 5: --- False

13 return '2\*2\*2\*2\*' + str(5)

return '2\*2\*2\*2\*5'

7. remove\_digit(754, 4) = 75

1 def remove\_digit(number=754, digit=4)

2 res = 0

3 power = 1

4 while 754 > 0: --- True

5 cur\_digit = 754 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 754 // 10

number = 75

4 while 75 > 0: --- True

5 cur\_digit = 75 % 10

cur\_digit = 5

6 if 5 != 4: --- True

7 res = 0 + 5 \* 1

res = 5

8 power = 1 \* 10

power = 10

9 number = 75 // 10

number = 7

4 while 7 > 0: --- True

5 cur\_digit = 7 % 10

cur\_digit = 7

6 if 7 != 4: --- True

7 res = 5 + 7 \* 10

res = 75

8 power = 10 \* 10

power = 100

9 number = 7 // 10

number = 0

4 while 0 > 0: --- False

10 return 75

Вариант: 1-2-41

1. gcd(-78, -33) = 3

1 def gcd(x=-78, y=-33)

2 if -78 < 0: --- True

3 x = --78

x = 78

4 if -33 < 0: --- True

5 y = --33

y = 33

6 while 33 != 0: --- True

7 rem = 78 % 33

rem = 12

8 x = 33

9 y = 12

6 while 12 != 0: --- True

7 rem = 33 % 12

rem = 9

8 x = 12

9 y = 9

6 while 9 != 0: --- True

7 rem = 12 % 9

rem = 3

8 x = 9

9 y = 3

6 while 3 != 0: --- True

7 rem = 9 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(-53, 0) = 53

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

2 if -53 < 0: --- True

3 x = --53

x = 53

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 53

3. hex(180) = 'B4'

1 def hex(number=180)

2 if 180 == 0: --- False

4 res = ''

5 while 180 > 0: --- True

6 digit = 180 % 16

digit = 4

7 if 4 <= 9: --- True

8 digit = str(4)

digit = '4'

23 res = '4' + ''

res = '4'

24 number = 180 // 16

number = 11

5 while 11 > 0: --- True

6 digit = 11 % 16

digit = 11

7 if 11 <= 9: --- False

9 elif 11 <= 13: --- True

10 if 11 <= 11: --- True

11 if 11 == 10: --- False

13 else:

14 digit = 'B'

23 res = 'B' + '4'

res = 'B4'

24 number = 11 // 16

number = 0

5 while 0 > 0: --- False

25 return 'B4'

4. square\_equal(-15, -3, 18) = '1.0 and -1.2'

3 def square\_equal(a=-15, b=-3, c=18)

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

5 D = -3\*-3 - 4\*-15\*18

D = 1089

6 if 1089 > 0: --- True

7 x1 = (--3 - sqrt(1089)) / (2\*-15)

x1 = 1.0

8 x2 = (--3 + sqrt(1089)) / (2\*-15)

x2 = -1.2

9 return str(1.0) + ' and ' + str(-1.2)

return '1.0 and -1.2'

5. square\_equal(36, -35, 15) = 'no roots'

3 def square\_equal(a=36, b=-35, c=15)

4 if 36 != 0: --- True

5 D = -35\*-35 - 4\*36\*15

D = -935

6 if -935 > 0: --- False

10 elif -935 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(25) = '5\*5'

1 def factorize(n=25)

2 res = ''

3 while 25 > 2 and 25 % 2 == 0: --- False

6 d = 3

7 while 25 > 3: --- True

8 if 25 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 25 > 5: --- True

8 if 25 % 5 == 0: --- True

9 res = '' + str(5) + '\*'

res = '5\*'

10 n = 25 // 5

n = 5

7 while 5 > 5: --- False

13 return '5\*' + str(5)

return '5\*5'

7. remove\_digit(684, 8) = 64

1 def remove\_digit(number=684, digit=8)

2 res = 0

3 power = 1

4 while 684 > 0: --- True

5 cur\_digit = 684 % 10

cur\_digit = 4

6 if 4 != 8: --- True

7 res = 0 + 4 \* 1

res = 4

8 power = 1 \* 10

power = 10

9 number = 684 // 10

number = 68

4 while 68 > 0: --- True

5 cur\_digit = 68 % 10

cur\_digit = 8

6 if 8 != 8: --- False

9 number = 68 // 10

number = 6

4 while 6 > 0: --- True

5 cur\_digit = 6 % 10

cur\_digit = 6

6 if 6 != 8: --- True

7 res = 4 + 6 \* 10

res = 64

8 power = 10 \* 10

power = 100

9 number = 6 // 10

number = 0

4 while 0 > 0: --- False

10 return 64

Вариант: 1-2-42

1. gcd(39, -51) = 3

1 def gcd(x=39, y=-51)

2 if 39 < 0: --- False

4 if -51 < 0: --- True

5 y = --51

y = 51

6 while 51 != 0: --- True

7 rem = 39 % 51

rem = 39

8 x = 51

9 y = 39

6 while 39 != 0: --- True

7 rem = 51 % 39

rem = 12

8 x = 39

9 y = 12

6 while 12 != 0: --- True

7 rem = 39 % 12

rem = 3

8 x = 12

9 y = 3

6 while 3 != 0: --- True

7 rem = 12 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(0, -86) = 86

1 def gcd(x=0, y=-86)

2 if 0 < 0: --- False

4 if -86 < 0: --- True

5 y = --86

y = 86

6 while 86 != 0: --- True

7 rem = 0 % 86

rem = 0

8 x = 86

9 y = 0

6 while 0 != 0: --- False

10 return 86

3. hex(174) = 'AE'

1 def hex(number=174)

2 if 174 == 0: --- False

4 res = ''

5 while 174 > 0: --- True

6 digit = 174 % 16

digit = 14

7 if 14 <= 9: --- False

9 elif 14 <= 13: --- False

19 elif 14 == 14: --- True

20 digit = 'E'

23 res = 'E' + ''

res = 'E'

24 number = 174 // 16

number = 10

5 while 10 > 0: --- True

6 digit = 10 % 16

digit = 10

7 if 10 <= 9: --- False

9 elif 10 <= 13: --- True

10 if 10 <= 11: --- True

11 if 10 == 10: --- True

12 digit = 'A'

23 res = 'A' + 'E'

res = 'AE'

24 number = 10 // 16

number = 0

5 while 0 > 0: --- False

25 return 'AE'

4. square\_equal(0, 75, 24) = '-0.32'

3 def square\_equal(a=0, b=75, c=24)

4 if 0 != 0: --- False

14 else:

15 if 75 != 0: --- True

16 return str(-24 / 75)

return '-0.32'

5. square\_equal(-4, -1, -6) = 'no roots'

3 def square\_equal(a=-4, b=-1, c=-6)

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

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

D = -95

6 if -95 > 0: --- False

10 elif -95 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(8) = '2\*2\*2'

1 def factorize(n=8)

2 res = ''

3 while 8 > 2 and 8 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 8 // 2

n = 4

3 while 4 > 2 and 4 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 4 // 2

n = 2

3 while 2 > 2 and 2 % 2 == 0: --- False

6 d = 3

7 while 2 > 3: --- False

13 return '2\*2\*' + str(2)

return '2\*2\*2'

7. remove\_digit(843, 4) = 83

1 def remove\_digit(number=843, digit=4)

2 res = 0

3 power = 1

4 while 843 > 0: --- True

5 cur\_digit = 843 % 10

cur\_digit = 3

6 if 3 != 4: --- True

7 res = 0 + 3 \* 1

res = 3

8 power = 1 \* 10

power = 10

9 number = 843 // 10

number = 84

4 while 84 > 0: --- True

5 cur\_digit = 84 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 84 // 10

number = 8

4 while 8 > 0: --- True

5 cur\_digit = 8 % 10

cur\_digit = 8

6 if 8 != 4: --- True

7 res = 3 + 8 \* 10

res = 83

8 power = 10 \* 10

power = 100

9 number = 8 // 10

number = 0

4 while 0 > 0: --- False

10 return 83

Вариант: 1-2-43

1. gcd(15, 39) = 3

1 def gcd(x=15, y=39)

2 if 15 < 0: --- False

4 if 39 < 0: --- False

6 while 39 != 0: --- True

7 rem = 15 % 39

rem = 15

8 x = 39

9 y = 15

6 while 15 != 0: --- True

7 rem = 39 % 15

rem = 9

8 x = 15

9 y = 9

6 while 9 != 0: --- True

7 rem = 15 % 9

rem = 6

8 x = 9

9 y = 6

6 while 6 != 0: --- True

7 rem = 9 % 6

rem = 3

8 x = 6

9 y = 3

6 while 3 != 0: --- True

7 rem = 6 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(-43, 0) = 43

1 def gcd(x=-43, y=0)

2 if -43 < 0: --- True

3 x = --43

x = 43

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 43

3. hex(246) = 'F6'

1 def hex(number=246)

2 if 246 == 0: --- False

4 res = ''

5 while 246 > 0: --- True

6 digit = 246 % 16

digit = 6

7 if 6 <= 9: --- True

8 digit = str(6)

digit = '6'

23 res = '6' + ''

res = '6'

24 number = 246 // 16

number = 15

5 while 15 > 0: --- True

6 digit = 15 % 16

digit = 15

7 if 15 <= 9: --- False

9 elif 15 <= 13: --- False

19 elif 15 == 14: --- False

21 else:

22 digit = 'F'

23 res = 'F' + '6'

res = 'F6'

24 number = 15 // 16

number = 0

5 while 0 > 0: --- False

25 return 'F6'

4. square\_equal(-90, -36, 0) = '-0.0 and -0.4'

3 def square\_equal(a=-90, b=-36, c=0)

4 if -90 != 0: --- True

5 D = -36\*-36 - 4\*-90\*0

D = 1296

6 if 1296 > 0: --- True

7 x1 = (--36 - sqrt(1296)) / (2\*-90)

x1 = -0.0

8 x2 = (--36 + sqrt(1296)) / (2\*-90)

x2 = -0.4

9 return str(-0.0) + ' and ' + str(-0.4)

return '-0.0 and -0.4'

5. square\_equal(-52, 92, -88) = 'no roots'

3 def square\_equal(a=-52, b=92, c=-88)

4 if -52 != 0: --- True

5 D = 92\*92 - 4\*-52\*-88

D = -9840

6 if -9840 > 0: --- False

10 elif -9840 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(196) = '2\*2\*7\*7'

1 def factorize(n=196)

2 res = ''

3 while 196 > 2 and 196 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 196 // 2

n = 98

3 while 98 > 2 and 98 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 98 // 2

n = 49

3 while 49 > 2 and 49 % 2 == 0: --- False

6 d = 3

7 while 49 > 3: --- True

8 if 49 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 49 > 5: --- True

8 if 49 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 49 > 7: --- True

8 if 49 % 7 == 0: --- True

9 res = '2\*2\*' + str(7) + '\*'

res = '2\*2\*7\*'

10 n = 49 // 7

n = 7

7 while 7 > 7: --- False

13 return '2\*2\*7\*' + str(7)

return '2\*2\*7\*7'

7. remove\_digit(959, 5) = 99

1 def remove\_digit(number=959, digit=5)

2 res = 0

3 power = 1

4 while 959 > 0: --- True

5 cur\_digit = 959 % 10

cur\_digit = 9

6 if 9 != 5: --- True

7 res = 0 + 9 \* 1

res = 9

8 power = 1 \* 10

power = 10

9 number = 959 // 10

number = 95

4 while 95 > 0: --- True

5 cur\_digit = 95 % 10

cur\_digit = 5

6 if 5 != 5: --- False

9 number = 95 // 10

number = 9

4 while 9 > 0: --- True

5 cur\_digit = 9 % 10

cur\_digit = 9

6 if 9 != 5: --- True

7 res = 9 + 9 \* 10

res = 99

8 power = 10 \* 10

power = 100

9 number = 9 // 10

number = 0

4 while 0 > 0: --- False

10 return 99

Вариант: 1-2-44

1. gcd(48, -64) = 16

1 def gcd(x=48, y=-64)

2 if 48 < 0: --- False

4 if -64 < 0: --- True

5 y = --64

y = 64

6 while 64 != 0: --- True

7 rem = 48 % 64

rem = 48

8 x = 64

9 y = 48

6 while 48 != 0: --- True

7 rem = 64 % 48

rem = 16

8 x = 48

9 y = 16

6 while 16 != 0: --- True

7 rem = 48 % 16

rem = 0

8 x = 16

9 y = 0

6 while 0 != 0: --- False

10 return 16

2. gcd(-94, 0) = 94

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

2 if -94 < 0: --- True

3 x = --94

x = 94

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 94

3. hex(175) = 'AF'

1 def hex(number=175)

2 if 175 == 0: --- False

4 res = ''

5 while 175 > 0: --- True

6 digit = 175 % 16

digit = 15

7 if 15 <= 9: --- False

9 elif 15 <= 13: --- False

19 elif 15 == 14: --- False

21 else:

22 digit = 'F'

23 res = 'F' + ''

res = 'F'

24 number = 175 // 16

number = 10

5 while 10 > 0: --- True

6 digit = 10 % 16

digit = 10

7 if 10 <= 9: --- False

9 elif 10 <= 13: --- True

10 if 10 <= 11: --- True

11 if 10 == 10: --- True

12 digit = 'A'

23 res = 'A' + 'F'

res = 'AF'

24 number = 10 // 16

number = 0

5 while 0 > 0: --- False

25 return 'AF'

4. square\_equal(25, -7, 0) = '0.0 and 0.28'

3 def square\_equal(a=25, b=-7, c=0)

4 if 25 != 0: --- True

5 D = -7\*-7 - 4\*25\*0

D = 49

6 if 49 > 0: --- True

7 x1 = (--7 - sqrt(49)) / (2\*25)

x1 = 0.0

8 x2 = (--7 + sqrt(49)) / (2\*25)

x2 = 0.28

9 return str(0.0) + ' and ' + str(0.28)

return '0.0 and 0.28'

5. square\_equal(42, 86, 70) = 'no roots'

3 def square\_equal(a=42, b=86, c=70)

4 if 42 != 0: --- True

5 D = 86\*86 - 4\*42\*70

D = -4364

6 if -4364 > 0: --- False

10 elif -4364 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(20) = '2\*2\*5'

1 def factorize(n=20)

2 res = ''

3 while 20 > 2 and 20 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 20 // 2

n = 10

3 while 10 > 2 and 10 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 10 // 2

n = 5

3 while 5 > 2 and 5 % 2 == 0: --- False

6 d = 3

7 while 5 > 3: --- True

8 if 5 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 5 > 5: --- False

13 return '2\*2\*' + str(5)

return '2\*2\*5'

7. remove\_digit(725, 2) = 75

1 def remove\_digit(number=725, digit=2)

2 res = 0

3 power = 1

4 while 725 > 0: --- True

5 cur\_digit = 725 % 10

cur\_digit = 5

6 if 5 != 2: --- True

7 res = 0 + 5 \* 1

res = 5

8 power = 1 \* 10

power = 10

9 number = 725 // 10

number = 72

4 while 72 > 0: --- True

5 cur\_digit = 72 % 10

cur\_digit = 2

6 if 2 != 2: --- False

9 number = 72 // 10

number = 7

4 while 7 > 0: --- True

5 cur\_digit = 7 % 10

cur\_digit = 7

6 if 7 != 2: --- True

7 res = 5 + 7 \* 10

res = 75

8 power = 10 \* 10

power = 100

9 number = 7 // 10

number = 0

4 while 0 > 0: --- False

10 return 75

Вариант: 1-2-45

1. gcd(84, 60) = 12

1 def gcd(x=84, y=60)

2 if 84 < 0: --- False

4 if 60 < 0: --- False

6 while 60 != 0: --- True

7 rem = 84 % 60

rem = 24

8 x = 60

9 y = 24

6 while 24 != 0: --- True

7 rem = 60 % 24

rem = 12

8 x = 24

9 y = 12

6 while 12 != 0: --- True

7 rem = 24 % 12

rem = 0

8 x = 12

9 y = 0

6 while 0 != 0: --- False

10 return 12

2. gcd(0, -30) = 30

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

2 if 0 < 0: --- False

4 if -30 < 0: --- True

5 y = --30

y = 30

6 while 30 != 0: --- True

7 rem = 0 % 30

rem = 0

8 x = 30

9 y = 0

6 while 0 != 0: --- False

10 return 30

3. hex(209) = 'D1'

1 def hex(number=209)

2 if 209 == 0: --- False

4 res = ''

5 while 209 > 0: --- True

6 digit = 209 % 16

digit = 1

7 if 1 <= 9: --- True

8 digit = str(1)

digit = '1'

23 res = '1' + ''

res = '1'

24 number = 209 // 16

number = 13

5 while 13 > 0: --- True

6 digit = 13 % 16

digit = 13

7 if 13 <= 9: --- False

9 elif 13 <= 13: --- True

10 if 13 <= 11: --- False

15 elif 13 == 12: --- False

17 else:

18 digit = 'D'

23 res = 'D' + '1'

res = 'D1'

24 number = 13 // 16

number = 0

5 while 0 > 0: --- False

25 return 'D1'

4. square\_equal(20, 43, 6) = '-2.0 and -0.15'

3 def square\_equal(a=20, b=43, c=6)

4 if 20 != 0: --- True

5 D = 43\*43 - 4\*20\*6

D = 1369

6 if 1369 > 0: --- True

7 x1 = (-43 - sqrt(1369)) / (2\*20)

x1 = -2.0

8 x2 = (-43 + sqrt(1369)) / (2\*20)

x2 = -0.15

9 return str(-2.0) + ' and ' + str(-0.15)

return '-2.0 and -0.15'

5. square\_equal(44, -41, 31) = 'no roots'

3 def square\_equal(a=44, b=-41, c=31)

4 if 44 != 0: --- True

5 D = -41\*-41 - 4\*44\*31

D = -3775

6 if -3775 > 0: --- False

10 elif -3775 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(162) = '2\*3\*3\*3\*3'

1 def factorize(n=162)

2 res = ''

3 while 162 > 2 and 162 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 162 // 2

n = 81

3 while 81 > 2 and 81 % 2 == 0: --- False

6 d = 3

7 while 81 > 3: --- True

8 if 81 % 3 == 0: --- True

9 res = '2\*' + str(3) + '\*'

res = '2\*3\*'

10 n = 81 // 3

n = 27

7 while 27 > 3: --- True

8 if 27 % 3 == 0: --- True

9 res = '2\*3\*' + str(3) + '\*'

res = '2\*3\*3\*'

10 n = 27 // 3

n = 9

7 while 9 > 3: --- True

8 if 9 % 3 == 0: --- True

9 res = '2\*3\*3\*' + str(3) + '\*'

res = '2\*3\*3\*3\*'

10 n = 9 // 3

n = 3

7 while 3 > 3: --- False

13 return '2\*3\*3\*3\*' + str(3)

return '2\*3\*3\*3\*3'

7. remove\_digit(671, 7) = 61

1 def remove\_digit(number=671, digit=7)

2 res = 0

3 power = 1

4 while 671 > 0: --- True

5 cur\_digit = 671 % 10

cur\_digit = 1

6 if 1 != 7: --- True

7 res = 0 + 1 \* 1

res = 1

8 power = 1 \* 10

power = 10

9 number = 671 // 10

number = 67

4 while 67 > 0: --- True

5 cur\_digit = 67 % 10

cur\_digit = 7

6 if 7 != 7: --- False

9 number = 67 // 10

number = 6

4 while 6 > 0: --- True

5 cur\_digit = 6 % 10

cur\_digit = 6

6 if 6 != 7: --- True

7 res = 1 + 6 \* 10

res = 61

8 power = 10 \* 10

power = 100

9 number = 6 // 10

number = 0

4 while 0 > 0: --- False

10 return 61

Вариант: 1-2-46

1. gcd(-55, 66) = 11

1 def gcd(x=-55, y=66)

2 if -55 < 0: --- True

3 x = --55

x = 55

4 if 66 < 0: --- False

6 while 66 != 0: --- True

7 rem = 55 % 66

rem = 55

8 x = 66

9 y = 55

6 while 55 != 0: --- True

7 rem = 66 % 55

rem = 11

8 x = 55

9 y = 11

6 while 11 != 0: --- True

7 rem = 55 % 11

rem = 0

8 x = 11

9 y = 0

6 while 0 != 0: --- False

10 return 11

2. gcd(43, 0) = 43

1 def gcd(x=43, y=0)

2 if 43 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 43

3. hex(199) = 'C7'

1 def hex(number=199)

2 if 199 == 0: --- False

4 res = ''

5 while 199 > 0: --- True

6 digit = 199 % 16

digit = 7

7 if 7 <= 9: --- True

8 digit = str(7)

digit = '7'

23 res = '7' + ''

res = '7'

24 number = 199 // 16

number = 12

5 while 12 > 0: --- True

6 digit = 12 % 16

digit = 12

7 if 12 <= 9: --- False

9 elif 12 <= 13: --- True

10 if 12 <= 11: --- False

15 elif 12 == 12: --- True

16 digit = 'C'

23 res = 'C' + '7'

res = 'C7'

24 number = 12 // 16

number = 0

5 while 0 > 0: --- False

25 return 'C7'

4. square\_equal(-30, 3, 9) = '0.6 and -0.5'

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

4 if -30 != 0: --- True

5 D = 3\*3 - 4\*-30\*9

D = 1089

6 if 1089 > 0: --- True

7 x1 = (-3 - sqrt(1089)) / (2\*-30)

x1 = 0.6

8 x2 = (-3 + sqrt(1089)) / (2\*-30)

x2 = -0.5

9 return str(0.6) + ' and ' + str(-0.5)

return '0.6 and -0.5'

5. square\_equal(-64, -90, -63) = 'no roots'

3 def square\_equal(a=-64, b=-90, c=-63)

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

5 D = -90\*-90 - 4\*-64\*-63

D = -8028

6 if -8028 > 0: --- False

10 elif -8028 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(10) = '2\*5'

1 def factorize(n=10)

2 res = ''

3 while 10 > 2 and 10 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 10 // 2

n = 5

3 while 5 > 2 and 5 % 2 == 0: --- False

6 d = 3

7 while 5 > 3: --- True

8 if 5 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 5 > 5: --- False

13 return '2\*' + str(5)

return '2\*5'

7. remove\_digit(273, 7) = 23

1 def remove\_digit(number=273, digit=7)

2 res = 0

3 power = 1

4 while 273 > 0: --- True

5 cur\_digit = 273 % 10

cur\_digit = 3

6 if 3 != 7: --- True

7 res = 0 + 3 \* 1

res = 3

8 power = 1 \* 10

power = 10

9 number = 273 // 10

number = 27

4 while 27 > 0: --- True

5 cur\_digit = 27 % 10

cur\_digit = 7

6 if 7 != 7: --- False

9 number = 27 // 10

number = 2

4 while 2 > 0: --- True

5 cur\_digit = 2 % 10

cur\_digit = 2

6 if 2 != 7: --- True

7 res = 3 + 2 \* 10

res = 23

8 power = 10 \* 10

power = 100

9 number = 2 // 10

number = 0

4 while 0 > 0: --- False

10 return 23

Вариант: 1-2-47

1. gcd(-30, -75) = 15

1 def gcd(x=-30, y=-75)

2 if -30 < 0: --- True

3 x = --30

x = 30

4 if -75 < 0: --- True

5 y = --75

y = 75

6 while 75 != 0: --- True

7 rem = 30 % 75

rem = 30

8 x = 75

9 y = 30

6 while 30 != 0: --- True

7 rem = 75 % 30

rem = 15

8 x = 30

9 y = 15

6 while 15 != 0: --- True

7 rem = 30 % 15

rem = 0

8 x = 15

9 y = 0

6 while 0 != 0: --- False

10 return 15

2. gcd(0, -33) = 33

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

2 if 0 < 0: --- False

4 if -33 < 0: --- True

5 y = --33

y = 33

6 while 33 != 0: --- True

7 rem = 0 % 33

rem = 0

8 x = 33

9 y = 0

6 while 0 != 0: --- False

10 return 33

3. hex(234) = 'EA'

1 def hex(number=234)

2 if 234 == 0: --- False

4 res = ''

5 while 234 > 0: --- True

6 digit = 234 % 16

digit = 10

7 if 10 <= 9: --- False

9 elif 10 <= 13: --- True

10 if 10 <= 11: --- True

11 if 10 == 10: --- True

12 digit = 'A'

23 res = 'A' + ''

res = 'A'

24 number = 234 // 16

number = 14

5 while 14 > 0: --- True

6 digit = 14 % 16

digit = 14

7 if 14 <= 9: --- False

9 elif 14 <= 13: --- False

19 elif 14 == 14: --- True

20 digit = 'E'

23 res = 'E' + 'A'

res = 'EA'

24 number = 14 // 16

number = 0

5 while 0 > 0: --- False

25 return 'EA'

4. square\_equal(40, -38, -51) = '-0.75 and 1.7'

3 def square\_equal(a=40, b=-38, c=-51)

4 if 40 != 0: --- True

5 D = -38\*-38 - 4\*40\*-51

D = 9604

6 if 9604 > 0: --- True

7 x1 = (--38 - sqrt(9604)) / (2\*40)

x1 = -0.75

8 x2 = (--38 + sqrt(9604)) / (2\*40)

x2 = 1.7

9 return str(-0.75) + ' and ' + str(1.7)

return '-0.75 and 1.7'

5. square\_equal(-37, 58, -24) = 'no roots'

3 def square\_equal(a=-37, b=58, c=-24)

4 if -37 != 0: --- True

5 D = 58\*58 - 4\*-37\*-24

D = -188

6 if -188 > 0: --- False

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

12 else:

13 return 'no roots'

6. factorize(175) = '5\*5\*7'

1 def factorize(n=175)

2 res = ''

3 while 175 > 2 and 175 % 2 == 0: --- False

6 d = 3

7 while 175 > 3: --- True

8 if 175 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 175 > 5: --- True

8 if 175 % 5 == 0: --- True

9 res = '' + str(5) + '\*'

res = '5\*'

10 n = 175 // 5

n = 35

7 while 35 > 5: --- True

8 if 35 % 5 == 0: --- True

9 res = '5\*' + str(5) + '\*'

res = '5\*5\*'

10 n = 35 // 5

n = 7

7 while 7 > 5: --- True

8 if 7 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 7 > 7: --- False

13 return '5\*5\*' + str(7)

return '5\*5\*7'

7. remove\_digit(498, 9) = 48

1 def remove\_digit(number=498, digit=9)

2 res = 0

3 power = 1

4 while 498 > 0: --- True

5 cur\_digit = 498 % 10

cur\_digit = 8

6 if 8 != 9: --- True

7 res = 0 + 8 \* 1

res = 8

8 power = 1 \* 10

power = 10

9 number = 498 // 10

number = 49

4 while 49 > 0: --- True

5 cur\_digit = 49 % 10

cur\_digit = 9

6 if 9 != 9: --- False

9 number = 49 // 10

number = 4

4 while 4 > 0: --- True

5 cur\_digit = 4 % 10

cur\_digit = 4

6 if 4 != 9: --- True

7 res = 8 + 4 \* 10

res = 48

8 power = 10 \* 10

power = 100

9 number = 4 // 10

number = 0

4 while 0 > 0: --- False

10 return 48

Вариант: 1-2-48

1. gcd(28, 32) = 4

1 def gcd(x=28, y=32)

2 if 28 < 0: --- False

4 if 32 < 0: --- False

6 while 32 != 0: --- True

7 rem = 28 % 32

rem = 28

8 x = 32

9 y = 28

6 while 28 != 0: --- True

7 rem = 32 % 28

rem = 4

8 x = 28

9 y = 4

6 while 4 != 0: --- True

7 rem = 28 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(-22, 0) = 22

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

2 if -22 < 0: --- True

3 x = --22

x = 22

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 22

3. hex(179) = 'B3'

1 def hex(number=179)

2 if 179 == 0: --- False

4 res = ''

5 while 179 > 0: --- True

6 digit = 179 % 16

digit = 3

7 if 3 <= 9: --- True

8 digit = str(3)

digit = '3'

23 res = '3' + ''

res = '3'

24 number = 179 // 16

number = 11

5 while 11 > 0: --- True

6 digit = 11 % 16

digit = 11

7 if 11 <= 9: --- False

9 elif 11 <= 13: --- True

10 if 11 <= 11: --- True

11 if 11 == 10: --- False

13 else:

14 digit = 'B'

23 res = 'B' + '3'

res = 'B3'

24 number = 11 // 16

number = 0

5 while 0 > 0: --- False

25 return 'B3'

4. square\_equal(15, 69, 36) = '-4.0 and -0.6'

3 def square\_equal(a=15, b=69, c=36)

4 if 15 != 0: --- True

5 D = 69\*69 - 4\*15\*36

D = 2601

6 if 2601 > 0: --- True

7 x1 = (-69 - sqrt(2601)) / (2\*15)

x1 = -4.0

8 x2 = (-69 + sqrt(2601)) / (2\*15)

x2 = -0.6

9 return str(-4.0) + ' and ' + str(-0.6)

return '-4.0 and -0.6'

5. square\_equal(79, -20, 20) = 'no roots'

3 def square\_equal(a=79, b=-20, c=20)

4 if 79 != 0: --- True

5 D = -20\*-20 - 4\*79\*20

D = -5920

6 if -5920 > 0: --- False

10 elif -5920 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(350) = '2\*5\*5\*7'

1 def factorize(n=350)

2 res = ''

3 while 350 > 2 and 350 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 350 // 2

n = 175

3 while 175 > 2 and 175 % 2 == 0: --- False

6 d = 3

7 while 175 > 3: --- True

8 if 175 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 175 > 5: --- True

8 if 175 % 5 == 0: --- True

9 res = '2\*' + str(5) + '\*'

res = '2\*5\*'

10 n = 175 // 5

n = 35

7 while 35 > 5: --- True

8 if 35 % 5 == 0: --- True

9 res = '2\*5\*' + str(5) + '\*'

res = '2\*5\*5\*'

10 n = 35 // 5

n = 7

7 while 7 > 5: --- True

8 if 7 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 7 > 7: --- False

13 return '2\*5\*5\*' + str(7)

return '2\*5\*5\*7'

7. remove\_digit(414, 1) = 44

1 def remove\_digit(number=414, digit=1)

2 res = 0

3 power = 1

4 while 414 > 0: --- True

5 cur\_digit = 414 % 10

cur\_digit = 4

6 if 4 != 1: --- True

7 res = 0 + 4 \* 1

res = 4

8 power = 1 \* 10

power = 10

9 number = 414 // 10

number = 41

4 while 41 > 0: --- True

5 cur\_digit = 41 % 10

cur\_digit = 1

6 if 1 != 1: --- False

9 number = 41 // 10

number = 4

4 while 4 > 0: --- True

5 cur\_digit = 4 % 10

cur\_digit = 4

6 if 4 != 1: --- True

7 res = 4 + 4 \* 10

res = 44

8 power = 10 \* 10

power = 100

9 number = 4 // 10

number = 0

4 while 0 > 0: --- False

10 return 44

Вариант: 1-2-49

1. gcd(77, 88) = 11

1 def gcd(x=77, y=88)

2 if 77 < 0: --- False

4 if 88 < 0: --- False

6 while 88 != 0: --- True

7 rem = 77 % 88

rem = 77

8 x = 88

9 y = 77

6 while 77 != 0: --- True

7 rem = 88 % 77

rem = 11

8 x = 77

9 y = 11

6 while 11 != 0: --- True

7 rem = 77 % 11

rem = 0

8 x = 11

9 y = 0

6 while 0 != 0: --- False

10 return 11

2. gcd(0, 22) = 22

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

2 if 0 < 0: --- False

4 if 22 < 0: --- False

6 while 22 != 0: --- True

7 rem = 0 % 22

rem = 0

8 x = 22

9 y = 0

6 while 0 != 0: --- False

10 return 22

3. hex(239) = 'EF'

1 def hex(number=239)

2 if 239 == 0: --- False

4 res = ''

5 while 239 > 0: --- True

6 digit = 239 % 16

digit = 15

7 if 15 <= 9: --- False

9 elif 15 <= 13: --- False

19 elif 15 == 14: --- False

21 else:

22 digit = 'F'

23 res = 'F' + ''

res = 'F'

24 number = 239 // 16

number = 14

5 while 14 > 0: --- True

6 digit = 14 % 16

digit = 14

7 if 14 <= 9: --- False

9 elif 14 <= 13: --- False

19 elif 14 == 14: --- True

20 digit = 'E'

23 res = 'E' + 'F'

res = 'EF'

24 number = 14 // 16

number = 0

5 while 0 > 0: --- False

25 return 'EF'

4. square\_equal(0, 25, 69) = '-2.76'

3 def square\_equal(a=0, b=25, c=69)

4 if 0 != 0: --- False

14 else:

15 if 25 != 0: --- True

16 return str(-69 / 25)

return '-2.76'

5. square\_equal(19, -2, 86) = 'no roots'

3 def square\_equal(a=19, b=-2, c=86)

4 if 19 != 0: --- True

5 D = -2\*-2 - 4\*19\*86

D = -6532

6 if -6532 > 0: --- False

10 elif -6532 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(216) = '2\*2\*2\*3\*3\*3'

1 def factorize(n=216)

2 res = ''

3 while 216 > 2 and 216 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 216 // 2

n = 108

3 while 108 > 2 and 108 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 108 // 2

n = 54

3 while 54 > 2 and 54 % 2 == 0: --- True

4 res = '2\*2\*' + '2\*'

res = '2\*2\*2\*'

5 n = 54 // 2

n = 27

3 while 27 > 2 and 27 % 2 == 0: --- False

6 d = 3

7 while 27 > 3: --- True

8 if 27 % 3 == 0: --- True

9 res = '2\*2\*2\*' + str(3) + '\*'

res = '2\*2\*2\*3\*'

10 n = 27 // 3

n = 9

7 while 9 > 3: --- True

8 if 9 % 3 == 0: --- True

9 res = '2\*2\*2\*3\*' + str(3) + '\*'

res = '2\*2\*2\*3\*3\*'

10 n = 9 // 3

n = 3

7 while 3 > 3: --- False

13 return '2\*2\*2\*3\*3\*' + str(3)

return '2\*2\*2\*3\*3\*3'

7. remove\_digit(68, 8) = 6

1 def remove\_digit(number=68, digit=8)

2 res = 0

3 power = 1

4 while 68 > 0: --- True

5 cur\_digit = 68 % 10

cur\_digit = 8

6 if 8 != 8: --- False

9 number = 68 // 10

number = 6

4 while 6 > 0: --- True

5 cur\_digit = 6 % 10

cur\_digit = 6

6 if 6 != 8: --- True

7 res = 0 + 6 \* 1

res = 6

8 power = 1 \* 10

power = 10

9 number = 6 // 10

number = 0

4 while 0 > 0: --- False

10 return 6

Вариант: 1-2-50

1. gcd(-15, -72) = 3

1 def gcd(x=-15, y=-72)

2 if -15 < 0: --- True

3 x = --15

x = 15

4 if -72 < 0: --- True

5 y = --72

y = 72

6 while 72 != 0: --- True

7 rem = 15 % 72

rem = 15

8 x = 72

9 y = 15

6 while 15 != 0: --- True

7 rem = 72 % 15

rem = 12

8 x = 15

9 y = 12

6 while 12 != 0: --- True

7 rem = 15 % 12

rem = 3

8 x = 12

9 y = 3

6 while 3 != 0: --- True

7 rem = 12 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(98, 0) = 98

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

2 if 98 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 98

3. hex(161) = 'A1'

1 def hex(number=161)

2 if 161 == 0: --- False

4 res = ''

5 while 161 > 0: --- True

6 digit = 161 % 16

digit = 1

7 if 1 <= 9: --- True

8 digit = str(1)

digit = '1'

23 res = '1' + ''

res = '1'

24 number = 161 // 16

number = 10

5 while 10 > 0: --- True

6 digit = 10 % 16

digit = 10

7 if 10 <= 9: --- False

9 elif 10 <= 13: --- True

10 if 10 <= 11: --- True

11 if 10 == 10: --- True

12 digit = 'A'

23 res = 'A' + '1'

res = 'A1'

24 number = 10 // 16

number = 0

5 while 0 > 0: --- False

25 return 'A1'

4. square\_equal(10, 57, 63) = '-4.2 and -1.5'

3 def square\_equal(a=10, b=57, c=63)

4 if 10 != 0: --- True

5 D = 57\*57 - 4\*10\*63

D = 729

6 if 729 > 0: --- True

7 x1 = (-57 - sqrt(729)) / (2\*10)

x1 = -4.2

8 x2 = (-57 + sqrt(729)) / (2\*10)

x2 = -1.5

9 return str(-4.2) + ' and ' + str(-1.5)

return '-4.2 and -1.5'

5. square\_equal(-48, 17, -8) = 'no roots'

3 def square\_equal(a=-48, b=17, c=-8)

4 if -48 != 0: --- True

5 D = 17\*17 - 4\*-48\*-8

D = -1247

6 if -1247 > 0: --- False

10 elif -1247 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(144) = '2\*2\*2\*2\*3\*3'

1 def factorize(n=144)

2 res = ''

3 while 144 > 2 and 144 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 144 // 2

n = 72

3 while 72 > 2 and 72 % 2 == 0: --- True

4 res = '2\*' + '2\*'

res = '2\*2\*'

5 n = 72 // 2

n = 36

3 while 36 > 2 and 36 % 2 == 0: --- True

4 res = '2\*2\*' + '2\*'

res = '2\*2\*2\*'

5 n = 36 // 2

n = 18

3 while 18 > 2 and 18 % 2 == 0: --- True

4 res = '2\*2\*2\*' + '2\*'

res = '2\*2\*2\*2\*'

5 n = 18 // 2

n = 9

3 while 9 > 2 and 9 % 2 == 0: --- False

6 d = 3

7 while 9 > 3: --- True

8 if 9 % 3 == 0: --- True

9 res = '2\*2\*2\*2\*' + str(3) + '\*'

res = '2\*2\*2\*2\*3\*'

10 n = 9 // 3

n = 3

7 while 3 > 3: --- False

13 return '2\*2\*2\*2\*3\*' + str(3)

return '2\*2\*2\*2\*3\*3'

7. remove\_digit(203, 0) = 23

1 def remove\_digit(number=203, digit=0)

2 res = 0

3 power = 1

4 while 203 > 0: --- True

5 cur\_digit = 203 % 10

cur\_digit = 3

6 if 3 != 0: --- True

7 res = 0 + 3 \* 1

res = 3

8 power = 1 \* 10

power = 10

9 number = 203 // 10

number = 20

4 while 20 > 0: --- True

5 cur\_digit = 20 % 10

cur\_digit = 0

6 if 0 != 0: --- False

9 number = 20 // 10

number = 2

4 while 2 > 0: --- True

5 cur\_digit = 2 % 10

cur\_digit = 2

6 if 2 != 0: --- True

7 res = 3 + 2 \* 10

res = 23

8 power = 10 \* 10

power = 100

9 number = 2 // 10

number = 0

4 while 0 > 0: --- False

10 return 23

Вариант: 1-2-51

1. gcd(27, -63) = 9

1 def gcd(x=27, y=-63)

2 if 27 < 0: --- False

4 if -63 < 0: --- True

5 y = --63

y = 63

6 while 63 != 0: --- True

7 rem = 27 % 63

rem = 27

8 x = 63

9 y = 27

6 while 27 != 0: --- True

7 rem = 63 % 27

rem = 9

8 x = 27

9 y = 9

6 while 9 != 0: --- True

7 rem = 27 % 9

rem = 0

8 x = 9

9 y = 0

6 while 0 != 0: --- False

10 return 9

2. gcd(0, 27) = 27

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

2 if 0 < 0: --- False

4 if 27 < 0: --- False

6 while 27 != 0: --- True

7 rem = 0 % 27

rem = 0

8 x = 27

9 y = 0

6 while 0 != 0: --- False

10 return 27

3. hex(193) = 'C1'

1 def hex(number=193)

2 if 193 == 0: --- False

4 res = ''

5 while 193 > 0: --- True

6 digit = 193 % 16

digit = 1

7 if 1 <= 9: --- True

8 digit = str(1)

digit = '1'

23 res = '1' + ''

res = '1'

24 number = 193 // 16

number = 12

5 while 12 > 0: --- True

6 digit = 12 % 16

digit = 12

7 if 12 <= 9: --- False

9 elif 12 <= 13: --- True

10 if 12 <= 11: --- False

15 elif 12 == 12: --- True

16 digit = 'C'

23 res = 'C' + '1'

res = 'C1'

24 number = 12 // 16

number = 0

5 while 0 > 0: --- False

25 return 'C1'

4. square\_equal(20, -42, -26) = '-0.5 and 2.6'

3 def square\_equal(a=20, b=-42, c=-26)

4 if 20 != 0: --- True

5 D = -42\*-42 - 4\*20\*-26

D = 3844

6 if 3844 > 0: --- True

7 x1 = (--42 - sqrt(3844)) / (2\*20)

x1 = -0.5

8 x2 = (--42 + sqrt(3844)) / (2\*20)

x2 = 2.6

9 return str(-0.5) + ' and ' + str(2.6)

return '-0.5 and 2.6'

5. square\_equal(4, -10, 51) = 'no roots'

3 def square\_equal(a=4, b=-10, c=51)

4 if 4 != 0: --- True

5 D = -10\*-10 - 4\*4\*51

D = -716

6 if -716 > 0: --- False

10 elif -716 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(121) = '11\*11'

1 def factorize(n=121)

2 res = ''

3 while 121 > 2 and 121 % 2 == 0: --- False

6 d = 3

7 while 121 > 3: --- True

8 if 121 % 3 == 0: --- False

11 else:

12 d = 3 + 2

d = 5

7 while 121 > 5: --- True

8 if 121 % 5 == 0: --- False

11 else:

12 d = 5 + 2

d = 7

7 while 121 > 7: --- True

8 if 121 % 7 == 0: --- False

11 else:

12 d = 7 + 2

d = 9

7 while 121 > 9: --- True

8 if 121 % 9 == 0: --- False

11 else:

12 d = 9 + 2

d = 11

7 while 121 > 11: --- True

8 if 121 % 11 == 0: --- True

9 res = '' + str(11) + '\*'

res = '11\*'

10 n = 121 // 11

n = 11

7 while 11 > 11: --- False

13 return '11\*' + str(11)

return '11\*11'

7. remove\_digit(339, 9) = 33

1 def remove\_digit(number=339, digit=9)

2 res = 0

3 power = 1

4 while 339 > 0: --- True

5 cur\_digit = 339 % 10

cur\_digit = 9

6 if 9 != 9: --- False

9 number = 339 // 10

number = 33

4 while 33 > 0: --- True

5 cur\_digit = 33 % 10

cur\_digit = 3

6 if 3 != 9: --- True

7 res = 0 + 3 \* 1

res = 3

8 power = 1 \* 10

power = 10

9 number = 33 // 10

number = 3

4 while 3 > 0: --- True

5 cur\_digit = 3 % 10

cur\_digit = 3

6 if 3 != 9: --- True

7 res = 3 + 3 \* 10

res = 33

8 power = 10 \* 10

power = 100

9 number = 3 // 10

number = 0

4 while 0 > 0: --- False

10 return 33