Вариант: 1-3-1

1. gcd(-40, -88) = 8

1 def gcd(x=-40, y=-88)

2 if -40 < 0: --- True

3 x = --40

x = 40

4 if -88 < 0: --- True

5 y = --88

y = 88

6 while 88 != 0: --- True

7 rem = 40 % 88

rem = 40

8 x = 88

9 y = 40

6 while 40 != 0: --- True

7 rem = 88 % 40

rem = 8

8 x = 40

9 y = 8

6 while 8 != 0: --- True

7 rem = 40 % 8

rem = 0

8 x = 8

9 y = 0

6 while 0 != 0: --- False

10 return 8

2. gcd(0, -38) = 38

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

2 if 0 < 0: --- False

4 if -38 < 0: --- True

5 y = --38

y = 38

6 while 38 != 0: --- True

7 rem = 0 % 38

rem = 0

8 x = 38

9 y = 0

6 while 0 != 0: --- False

10 return 38

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(0, -4, 91) = '22.75'

3 def square\_equal(a=0, b=-4, c=91)

4 if 0 != 0: --- False

14 else:

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

16 return str(-91 / -4)

return '22.75'

5. square\_equal(49, 28, 53) = 'no roots'

3 def square\_equal(a=49, b=28, c=53)

4 if 49 != 0: --- True

5 D = 28\*28 - 4\*49\*53

D = -9604

6 if -9604 > 0: --- False

10 elif -9604 == 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(166, 6) = 1

1 def remove\_digit(number=166, digit=6)

2 res = 0

3 power = 1

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: 1-3-2

1. gcd(-78, 9) = 3

1 def gcd(x=-78, y=9)

2 if -78 < 0: --- True

3 x = --78

x = 78

4 if 9 < 0: --- False

6 while 9 != 0: --- True

7 rem = 78 % 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(11, 0) = 11

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

2 if 11 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 11

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(0, -36, 9) = '0.25'

3 def square\_equal(a=0, b=-36, c=9)

4 if 0 != 0: --- False

14 else:

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

16 return str(-9 / -36)

return '0.25'

5. square\_equal(96, 30, 25) = 'no roots'

3 def square\_equal(a=96, b=30, c=25)

4 if 96 != 0: --- True

5 D = 30\*30 - 4\*96\*25

D = -8700

6 if -8700 > 0: --- False

10 elif -8700 == 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(443, 3) = 44

1 def remove\_digit(number=443, digit=3)

2 res = 0

3 power = 1

4 while 443 > 0: --- True

5 cur\_digit = 443 % 10

cur\_digit = 3

6 if 3 != 3: --- False

9 number = 443 // 10

number = 44

4 while 44 > 0: --- True

5 cur\_digit = 44 % 10

cur\_digit = 4

6 if 4 != 3: --- True

7 res = 0 + 4 \* 1

res = 4

8 power = 1 \* 10

power = 10

9 number = 44 // 10

number = 4

4 while 4 > 0: --- True

5 cur\_digit = 4 % 10

cur\_digit = 4

6 if 4 != 3: --- 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: 1-3-3

1. gcd(21, 81) = 3

1 def gcd(x=21, y=81)

2 if 21 < 0: --- False

4 if 81 < 0: --- False

6 while 81 != 0: --- True

7 rem = 21 % 81

rem = 21

8 x = 81

9 y = 21

6 while 21 != 0: --- True

7 rem = 81 % 21

rem = 18

8 x = 21

9 y = 18

6 while 18 != 0: --- True

7 rem = 21 % 18

rem = 3

8 x = 18

9 y = 3

6 while 3 != 0: --- True

7 rem = 18 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(0, -57) = 57

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

2 if 0 < 0: --- False

4 if -57 < 0: --- True

5 y = --57

y = 57

6 while 57 != 0: --- True

7 rem = 0 % 57

rem = 0

8 x = 57

9 y = 0

6 while 0 != 0: --- False

10 return 57

3. hex(214) = 'D6'

1 def hex(number=214)

2 if 214 == 0: --- False

4 res = ''

5 while 214 > 0: --- True

6 digit = 214 % 16

digit = 6

7 if 6 <= 9: --- True

8 digit = str(6)

digit = '6'

23 res = '6' + ''

res = '6'

24 number = 214 // 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' + '6'

res = 'D6'

24 number = 13 // 16

number = 0

5 while 0 > 0: --- False

25 return 'D6'

4. square\_equal(0, -8, -8) = '-1.0'

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

4 if 0 != 0: --- False

14 else:

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

16 return str(--8 / -8)

return '-1.0'

5. square\_equal(14, 48, 89) = 'no roots'

3 def square\_equal(a=14, b=48, c=89)

4 if 14 != 0: --- True

5 D = 48\*48 - 4\*14\*89

D = -2680

6 if -2680 > 0: --- False

10 elif -2680 == 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(569, 9) = 56

1 def remove\_digit(number=569, digit=9)

2 res = 0

3 power = 1

4 while 569 > 0: --- True

5 cur\_digit = 569 % 10

cur\_digit = 9

6 if 9 != 9: --- False

9 number = 569 // 10

number = 56

4 while 56 > 0: --- True

5 cur\_digit = 56 % 10

cur\_digit = 6

6 if 6 != 9: --- True

7 res = 0 + 6 \* 1

res = 6

8 power = 1 \* 10

power = 10

9 number = 56 // 10

number = 5

4 while 5 > 0: --- True

5 cur\_digit = 5 % 10

cur\_digit = 5

6 if 5 != 9: --- True

7 res = 6 + 5 \* 10

res = 56

8 power = 10 \* 10

power = 100

9 number = 5 // 10

number = 0

4 while 0 > 0: --- False

10 return 56

Вариант: 1-3-4

1. gcd(-20, 35) = 5

1 def gcd(x=-20, y=35)

2 if -20 < 0: --- True

3 x = --20

x = 20

4 if 35 < 0: --- False

6 while 35 != 0: --- True

7 rem = 20 % 35

rem = 20

8 x = 35

9 y = 20

6 while 20 != 0: --- True

7 rem = 35 % 20

rem = 15

8 x = 20

9 y = 15

6 while 15 != 0: --- True

7 rem = 20 % 15

rem = 5

8 x = 15

9 y = 5

6 while 5 != 0: --- True

7 rem = 15 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

2. gcd(0, -96) = 96

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

2 if 0 < 0: --- False

4 if -96 < 0: --- True

5 y = --96

y = 96

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(198) = 'C6'

1 def hex(number=198)

2 if 198 == 0: --- False

4 res = ''

5 while 198 > 0: --- True

6 digit = 198 % 16

digit = 6

7 if 6 <= 9: --- True

8 digit = str(6)

digit = '6'

23 res = '6' + ''

res = '6'

24 number = 198 // 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' + '6'

res = 'C6'

24 number = 12 // 16

number = 0

5 while 0 > 0: --- False

25 return 'C6'

4. square\_equal(0, 60, 36) = '-0.6'

3 def square\_equal(a=0, b=60, c=36)

4 if 0 != 0: --- False

14 else:

15 if 60 != 0: --- True

16 return str(-36 / 60)

return '-0.6'

5. square\_equal(6, -20, 60) = 'no roots'

3 def square\_equal(a=6, b=-20, c=60)

4 if 6 != 0: --- True

5 D = -20\*-20 - 4\*6\*60

D = -1040

6 if -1040 > 0: --- False

10 elif -1040 == 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(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: 1-3-5

1. gcd(15, -36) = 3

1 def gcd(x=15, y=-36)

2 if 15 < 0: --- False

4 if -36 < 0: --- True

5 y = --36

y = 36

6 while 36 != 0: --- True

7 rem = 15 % 36

rem = 15

8 x = 36

9 y = 15

6 while 15 != 0: --- True

7 rem = 36 % 15

rem = 6

8 x = 15

9 y = 6

6 while 6 != 0: --- True

7 rem = 15 % 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(21, 0) = 21

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

2 if 21 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 21

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(0, -4, -1) = '-0.25'

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

4 if 0 != 0: --- False

14 else:

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

16 return str(--1 / -4)

return '-0.25'

5. square\_equal(-71, 80, -24) = 'no roots'

3 def square\_equal(a=-71, b=80, c=-24)

4 if -71 != 0: --- True

5 D = 80\*80 - 4\*-71\*-24

D = -416

6 if -416 > 0: --- False

10 elif -416 == 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(392, 9) = 32

1 def remove\_digit(number=392, digit=9)

2 res = 0

3 power = 1

4 while 392 > 0: --- True

5 cur\_digit = 392 % 10

cur\_digit = 2

6 if 2 != 9: --- True

7 res = 0 + 2 \* 1

res = 2

8 power = 1 \* 10

power = 10

9 number = 392 // 10

number = 39

4 while 39 > 0: --- True

5 cur\_digit = 39 % 10

cur\_digit = 9

6 if 9 != 9: --- False

9 number = 39 // 10

number = 3

4 while 3 > 0: --- True

5 cur\_digit = 3 % 10

cur\_digit = 3

6 if 3 != 9: --- 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: 1-3-6

1. gcd(-60, -21) = 3

1 def gcd(x=-60, y=-21)

2 if -60 < 0: --- True

3 x = --60

x = 60

4 if -21 < 0: --- True

5 y = --21

y = 21

6 while 21 != 0: --- True

7 rem = 60 % 21

rem = 18

8 x = 21

9 y = 18

6 while 18 != 0: --- True

7 rem = 21 % 18

rem = 3

8 x = 18

9 y = 3

6 while 3 != 0: --- True

7 rem = 18 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(-57, 0) = 57

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

2 if -57 < 0: --- True

3 x = --57

x = 57

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 57

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(0, 30, -42) = '1.4'

3 def square\_equal(a=0, b=30, c=-42)

4 if 0 != 0: --- False

14 else:

15 if 30 != 0: --- True

16 return str(--42 / 30)

return '1.4'

5. square\_equal(86, -14, 23) = 'no roots'

3 def square\_equal(a=86, b=-14, c=23)

4 if 86 != 0: --- True

5 D = -14\*-14 - 4\*86\*23

D = -7716

6 if -7716 > 0: --- False

10 elif -7716 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(56) = '2\*2\*2\*7'

1 def factorize(n=56)

2 res = ''

3 while 56 > 2 and 56 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 56 // 2

n = 28

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

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

res = '2\*2\*'

5 n = 28 // 2

n = 14

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

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

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

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

7. remove\_digit(64, 4) = 6

1 def remove\_digit(number=64, digit=4)

2 res = 0

3 power = 1

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: 1-3-7

1. gcd(-95, 100) = 5

1 def gcd(x=-95, y=100)

2 if -95 < 0: --- True

3 x = --95

x = 95

4 if 100 < 0: --- False

6 while 100 != 0: --- True

7 rem = 95 % 100

rem = 95

8 x = 100

9 y = 95

6 while 95 != 0: --- True

7 rem = 100 % 95

rem = 5

8 x = 95

9 y = 5

6 while 5 != 0: --- True

7 rem = 95 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

2. gcd(-78, 0) = 78

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

2 if -78 < 0: --- True

3 x = --78

x = 78

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 78

3. hex(237) = 'ED'

1 def hex(number=237)

2 if 237 == 0: --- False

4 res = ''

5 while 237 > 0: --- True

6 digit = 237 % 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 = 237 // 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' + 'D'

res = 'ED'

24 number = 14 // 16

number = 0

5 while 0 > 0: --- False

25 return 'ED'

4. square\_equal(0, -20, 84) = '4.2'

3 def square\_equal(a=0, b=-20, c=84)

4 if 0 != 0: --- False

14 else:

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

16 return str(-84 / -20)

return '4.2'

5. square\_equal(33, 52, 81) = 'no roots'

3 def square\_equal(a=33, b=52, c=81)

4 if 33 != 0: --- True

5 D = 52\*52 - 4\*33\*81

D = -7988

6 if -7988 > 0: --- False

10 elif -7988 == 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(396, 6) = 39

1 def remove\_digit(number=396, digit=6)

2 res = 0

3 power = 1

4 while 396 > 0: --- True

5 cur\_digit = 396 % 10

cur\_digit = 6

6 if 6 != 6: --- False

9 number = 396 // 10

number = 39

4 while 39 > 0: --- True

5 cur\_digit = 39 % 10

cur\_digit = 9

6 if 9 != 6: --- True

7 res = 0 + 9 \* 1

res = 9

8 power = 1 \* 10

power = 10

9 number = 39 // 10

number = 3

4 while 3 > 0: --- True

5 cur\_digit = 3 % 10

cur\_digit = 3

6 if 3 != 6: --- True

7 res = 9 + 3 \* 10

res = 39

8 power = 10 \* 10

power = 100

9 number = 3 // 10

number = 0

4 while 0 > 0: --- False

10 return 39

Вариант: 1-3-8

1. gcd(-21, -87) = 3

1 def gcd(x=-21, y=-87)

2 if -21 < 0: --- True

3 x = --21

x = 21

4 if -87 < 0: --- True

5 y = --87

y = 87

6 while 87 != 0: --- True

7 rem = 21 % 87

rem = 21

8 x = 87

9 y = 21

6 while 21 != 0: --- True

7 rem = 87 % 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(0, -26) = 26

1 def gcd(x=0, y=-26)

2 if 0 < 0: --- False

4 if -26 < 0: --- True

5 y = --26

y = 26

6 while 26 != 0: --- True

7 rem = 0 % 26

rem = 0

8 x = 26

9 y = 0

6 while 0 != 0: --- False

10 return 26

3. hex(162) = 'A2'

1 def hex(number=162)

2 if 162 == 0: --- False

4 res = ''

5 while 162 > 0: --- True

6 digit = 162 % 16

digit = 2

7 if 2 <= 9: --- True

8 digit = str(2)

digit = '2'

23 res = '2' + ''

res = '2'

24 number = 162 // 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' + '2'

res = 'A2'

24 number = 10 // 16

number = 0

5 while 0 > 0: --- False

25 return 'A2'

4. square\_equal(0, -32, 88) = '2.75'

3 def square\_equal(a=0, b=-32, c=88)

4 if 0 != 0: --- False

14 else:

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

16 return str(-88 / -32)

return '2.75'

5. square\_equal(-35, 22, -53) = 'no roots'

3 def square\_equal(a=-35, b=22, c=-53)

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

5 D = 22\*22 - 4\*-35\*-53

D = -6936

6 if -6936 > 0: --- False

10 elif -6936 == 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(638, 8) = 63

1 def remove\_digit(number=638, digit=8)

2 res = 0

3 power = 1

4 while 638 > 0: --- True

5 cur\_digit = 638 % 10

cur\_digit = 8

6 if 8 != 8: --- False

9 number = 638 // 10

number = 63

4 while 63 > 0: --- True

5 cur\_digit = 63 % 10

cur\_digit = 3

6 if 3 != 8: --- True

7 res = 0 + 3 \* 1

res = 3

8 power = 1 \* 10

power = 10

9 number = 63 // 10

number = 6

4 while 6 > 0: --- True

5 cur\_digit = 6 % 10

cur\_digit = 6

6 if 6 != 8: --- True

7 res = 3 + 6 \* 10

res = 63

8 power = 10 \* 10

power = 100

9 number = 6 // 10

number = 0

4 while 0 > 0: --- False

10 return 63

Вариант: 1-3-9

1. gcd(54, -99) = 9

1 def gcd(x=54, y=-99)

2 if 54 < 0: --- False

4 if -99 < 0: --- True

5 y = --99

y = 99

6 while 99 != 0: --- True

7 rem = 54 % 99

rem = 54

8 x = 99

9 y = 54

6 while 54 != 0: --- True

7 rem = 99 % 54

rem = 45

8 x = 54

9 y = 45

6 while 45 != 0: --- True

7 rem = 54 % 45

rem = 9

8 x = 45

9 y = 9

6 while 9 != 0: --- True

7 rem = 45 % 9

rem = 0

8 x = 9

9 y = 0

6 while 0 != 0: --- False

10 return 9

2. gcd(79, 0) = 79

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

2 if 79 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 79

3. hex(243) = 'F3'

1 def hex(number=243)

2 if 243 == 0: --- False

4 res = ''

5 while 243 > 0: --- True

6 digit = 243 % 16

digit = 3

7 if 3 <= 9: --- True

8 digit = str(3)

digit = '3'

23 res = '3' + ''

res = '3'

24 number = 243 // 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' + '3'

res = 'F3'

24 number = 15 // 16

number = 0

5 while 0 > 0: --- False

25 return 'F3'

4. square\_equal(0, -52, 65) = '1.25'

3 def square\_equal(a=0, b=-52, c=65)

4 if 0 != 0: --- False

14 else:

15 if -52 != 0: --- True

16 return str(-65 / -52)

return '1.25'

5. square\_equal(0, 50, 64) = '-1.28'

3 def square\_equal(a=0, b=50, c=64)

4 if 0 != 0: --- False

14 else:

15 if 50 != 0: --- True

16 return str(-64 / 50)

return '-1.28'

6. factorize(200) = '2\*2\*2\*5\*5'

1 def factorize(n=200)

2 res = ''

3 while 200 > 2 and 200 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 200 // 2

n = 100

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

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

res = '2\*2\*'

5 n = 100 // 2

n = 50

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

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

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

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

10 n = 25 // 5

n = 5

7 while 5 > 5: --- False

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

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

7. remove\_digit(627, 2) = 67

1 def remove\_digit(number=627, digit=2)

2 res = 0

3 power = 1

4 while 627 > 0: --- True

5 cur\_digit = 627 % 10

cur\_digit = 7

6 if 7 != 2: --- True

7 res = 0 + 7 \* 1

res = 7

8 power = 1 \* 10

power = 10

9 number = 627 // 10

number = 62

4 while 62 > 0: --- True

5 cur\_digit = 62 % 10

cur\_digit = 2

6 if 2 != 2: --- False

9 number = 62 // 10

number = 6

4 while 6 > 0: --- True

5 cur\_digit = 6 % 10

cur\_digit = 6

6 if 6 != 2: --- 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: 1-3-10

1. gcd(55, -35) = 5

1 def gcd(x=55, y=-35)

2 if 55 < 0: --- False

4 if -35 < 0: --- True

5 y = --35

y = 35

6 while 35 != 0: --- True

7 rem = 55 % 35

rem = 20

8 x = 35

9 y = 20

6 while 20 != 0: --- True

7 rem = 35 % 20

rem = 15

8 x = 20

9 y = 15

6 while 15 != 0: --- True

7 rem = 20 % 15

rem = 5

8 x = 15

9 y = 5

6 while 5 != 0: --- True

7 rem = 15 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

2. gcd(-56, 0) = 56

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

2 if -56 < 0: --- True

3 x = --56

x = 56

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 56

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(20, 47, -13) = '-2.6 and 0.25'

3 def square\_equal(a=20, b=47, c=-13)

4 if 20 != 0: --- True

5 D = 47\*47 - 4\*20\*-13

D = 3249

6 if 3249 > 0: --- True

7 x1 = (-47 - sqrt(3249)) / (2\*20)

x1 = -2.6

8 x2 = (-47 + sqrt(3249)) / (2\*20)

x2 = 0.25

9 return str(-2.6) + ' and ' + str(0.25)

return '-2.6 and 0.25'

5. square\_equal(46, -24, 5) = 'no roots'

3 def square\_equal(a=46, b=-24, c=5)

4 if 46 != 0: --- True

5 D = -24\*-24 - 4\*46\*5

D = -344

6 if -344 > 0: --- False

10 elif -344 == 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(106, 6) = 10

1 def remove\_digit(number=106, digit=6)

2 res = 0

3 power = 1

4 while 106 > 0: --- True

5 cur\_digit = 106 % 10

cur\_digit = 6

6 if 6 != 6: --- False

9 number = 106 // 10

number = 10

4 while 10 > 0: --- True

5 cur\_digit = 10 % 10

cur\_digit = 0

6 if 0 != 6: --- True

7 res = 0 + 0 \* 1

res = 0

8 power = 1 \* 10

power = 10

9 number = 10 // 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 \* 10

res = 10

8 power = 10 \* 10

power = 100

9 number = 1 // 10

number = 0

4 while 0 > 0: --- False

10 return 10

Вариант: 1-3-11

1. gcd(70, 55) = 5

1 def gcd(x=70, y=55)

2 if 70 < 0: --- False

4 if 55 < 0: --- False

6 while 55 != 0: --- True

7 rem = 70 % 55

rem = 15

8 x = 55

9 y = 15

6 while 15 != 0: --- True

7 rem = 55 % 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, 52) = 52

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

2 if 0 < 0: --- False

4 if 52 < 0: --- False

6 while 52 != 0: --- True

7 rem = 0 % 52

rem = 0

8 x = 52

9 y = 0

6 while 0 != 0: --- False

10 return 52

3. hex(242) = 'F2'

1 def hex(number=242)

2 if 242 == 0: --- False

4 res = ''

5 while 242 > 0: --- True

6 digit = 242 % 16

digit = 2

7 if 2 <= 9: --- True

8 digit = str(2)

digit = '2'

23 res = '2' + ''

res = '2'

24 number = 242 // 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' + '2'

res = 'F2'

24 number = 15 // 16

number = 0

5 while 0 > 0: --- False

25 return 'F2'

4. square\_equal(0, 100, -2) = '0.02'

3 def square\_equal(a=0, b=100, c=-2)

4 if 0 != 0: --- False

14 else:

15 if 100 != 0: --- True

16 return str(--2 / 100)

return '0.02'

5. square\_equal(21, 67, 67) = 'no roots'

3 def square\_equal(a=21, b=67, c=67)

4 if 21 != 0: --- True

5 D = 67\*67 - 4\*21\*67

D = -1139

6 if -1139 > 0: --- False

10 elif -1139 == 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(534, 3) = 54

1 def remove\_digit(number=534, digit=3)

2 res = 0

3 power = 1

4 while 534 > 0: --- True

5 cur\_digit = 534 % 10

cur\_digit = 4

6 if 4 != 3: --- True

7 res = 0 + 4 \* 1

res = 4

8 power = 1 \* 10

power = 10

9 number = 534 // 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 = 4 + 5 \* 10

res = 54

8 power = 10 \* 10

power = 100

9 number = 5 // 10

number = 0

4 while 0 > 0: --- False

10 return 54

Вариант: 1-3-12

1. gcd(-87, 90) = 3

1 def gcd(x=-87, y=90)

2 if -87 < 0: --- True

3 x = --87

x = 87

4 if 90 < 0: --- False

6 while 90 != 0: --- True

7 rem = 87 % 90

rem = 87

8 x = 90

9 y = 87

6 while 87 != 0: --- True

7 rem = 90 % 87

rem = 3

8 x = 87

9 y = 3

6 while 3 != 0: --- True

7 rem = 87 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(0, -10) = 10

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

2 if 0 < 0: --- False

4 if -10 < 0: --- True

5 y = --10

y = 10

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(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(0, 57, -57) = '1.0'

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

4 if 0 != 0: --- False

14 else:

15 if 57 != 0: --- True

16 return str(--57 / 57)

return '1.0'

5. square\_equal(0, 90, 99) = '-1.1'

3 def square\_equal(a=0, b=90, c=99)

4 if 0 != 0: --- False

14 else:

15 if 90 != 0: --- True

16 return str(-99 / 90)

return '-1.1'

6. factorize(72) = '2\*2\*2\*3\*3'

1 def factorize(n=72)

2 res = ''

3 while 72 > 2 and 72 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 72 // 2

n = 36

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

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

res = '2\*2\*'

5 n = 36 // 2

n = 18

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

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

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

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

10 n = 9 // 3

n = 3

7 while 3 > 3: --- False

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

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

7. remove\_digit(717, 1) = 77

1 def remove\_digit(number=717, digit=1)

2 res = 0

3 power = 1

4 while 717 > 0: --- True

5 cur\_digit = 717 % 10

cur\_digit = 7

6 if 7 != 1: --- True

7 res = 0 + 7 \* 1

res = 7

8 power = 1 \* 10

power = 10

9 number = 717 // 10

number = 71

4 while 71 > 0: --- True

5 cur\_digit = 71 % 10

cur\_digit = 1

6 if 1 != 1: --- False

9 number = 71 // 10

number = 7

4 while 7 > 0: --- True

5 cur\_digit = 7 % 10

cur\_digit = 7

6 if 7 != 1: --- 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: 1-3-13

1. gcd(76, 40) = 4

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

2 if 76 < 0: --- False

4 if 40 < 0: --- False

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, 93) = 93

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

2 if 0 < 0: --- False

4 if 93 < 0: --- False

6 while 93 != 0: --- True

7 rem = 0 % 93

rem = 0

8 x = 93

9 y = 0

6 while 0 != 0: --- False

10 return 93

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(0, 12, -66) = '5.5'

3 def square\_equal(a=0, b=12, c=-66)

4 if 0 != 0: --- False

14 else:

15 if 12 != 0: --- True

16 return str(--66 / 12)

return '5.5'

5. square\_equal(-22, -19, -39) = 'no roots'

3 def square\_equal(a=-22, b=-19, c=-39)

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

5 D = -19\*-19 - 4\*-22\*-39

D = -3071

6 if -3071 > 0: --- False

10 elif -3071 == 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(358, 8) = 35

1 def remove\_digit(number=358, digit=8)

2 res = 0

3 power = 1

4 while 358 > 0: --- True

5 cur\_digit = 358 % 10

cur\_digit = 8

6 if 8 != 8: --- False

9 number = 358 // 10

number = 35

4 while 35 > 0: --- True

5 cur\_digit = 35 % 10

cur\_digit = 5

6 if 5 != 8: --- True

7 res = 0 + 5 \* 1

res = 5

8 power = 1 \* 10

power = 10

9 number = 35 // 10

number = 3

4 while 3 > 0: --- True

5 cur\_digit = 3 % 10

cur\_digit = 3

6 if 3 != 8: --- True

7 res = 5 + 3 \* 10

res = 35

8 power = 10 \* 10

power = 100

9 number = 3 // 10

number = 0

4 while 0 > 0: --- False

10 return 35

Вариант: 1-3-14

1. gcd(-45, -24) = 3

1 def gcd(x=-45, y=-24)

2 if -45 < 0: --- True

3 x = --45

x = 45

4 if -24 < 0: --- True

5 y = --24

y = 24

6 while 24 != 0: --- True

7 rem = 45 % 24

rem = 21

8 x = 24

9 y = 21

6 while 21 != 0: --- True

7 rem = 24 % 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(0, 1) = 1

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

2 if 0 < 0: --- False

4 if 1 < 0: --- False

6 while 1 != 0: --- True

7 rem = 0 % 1

rem = 0

8 x = 1

9 y = 0

6 while 0 != 0: --- False

10 return 1

3. hex(191) = 'BF'

1 def hex(number=191)

2 if 191 == 0: --- False

4 res = ''

5 while 191 > 0: --- True

6 digit = 191 % 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 = 191 // 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' + 'F'

res = 'BF'

24 number = 11 // 16

number = 0

5 while 0 > 0: --- False

25 return 'BF'

4. square\_equal(0, 85, 68) = '-0.8'

3 def square\_equal(a=0, b=85, c=68)

4 if 0 != 0: --- False

14 else:

15 if 85 != 0: --- True

16 return str(-68 / 85)

return '-0.8'

5. square\_equal(-69, 70, -41) = 'no roots'

3 def square\_equal(a=-69, b=70, c=-41)

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

5 D = 70\*70 - 4\*-69\*-41

D = -6416

6 if -6416 > 0: --- False

10 elif -6416 == 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(496, 9) = 46

1 def remove\_digit(number=496, digit=9)

2 res = 0

3 power = 1

4 while 496 > 0: --- True

5 cur\_digit = 496 % 10

cur\_digit = 6

6 if 6 != 9: --- True

7 res = 0 + 6 \* 1

res = 6

8 power = 1 \* 10

power = 10

9 number = 496 // 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 = 6 + 4 \* 10

res = 46

8 power = 10 \* 10

power = 100

9 number = 4 // 10

number = 0

4 while 0 > 0: --- False

10 return 46

Вариант: 1-3-15

1. gcd(40, 95) = 5

1 def gcd(x=40, y=95)

2 if 40 < 0: --- False

4 if 95 < 0: --- False

6 while 95 != 0: --- True

7 rem = 40 % 95

rem = 40

8 x = 95

9 y = 40

6 while 40 != 0: --- True

7 rem = 95 % 40

rem = 15

8 x = 40

9 y = 15

6 while 15 != 0: --- True

7 rem = 40 % 15

rem = 10

8 x = 15

9 y = 10

6 while 10 != 0: --- True

7 rem = 15 % 10

rem = 5

8 x = 10

9 y = 5

6 while 5 != 0: --- True

7 rem = 10 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

2. gcd(-100, 0) = 100

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

2 if -100 < 0: --- True

3 x = --100

x = 100

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 100

3. hex(178) = 'B2'

1 def hex(number=178)

2 if 178 == 0: --- False

4 res = ''

5 while 178 > 0: --- True

6 digit = 178 % 16

digit = 2

7 if 2 <= 9: --- True

8 digit = str(2)

digit = '2'

23 res = '2' + ''

res = '2'

24 number = 178 // 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' + '2'

res = 'B2'

24 number = 11 // 16

number = 0

5 while 0 > 0: --- False

25 return 'B2'

4. square\_equal(0, 4, -24) = '6.0'

3 def square\_equal(a=0, b=4, c=-24)

4 if 0 != 0: --- False

14 else:

15 if 4 != 0: --- True

16 return str(--24 / 4)

return '6.0'

5. square\_equal(65, -7, 31) = 'no roots'

3 def square\_equal(a=65, b=-7, c=31)

4 if 65 != 0: --- True

5 D = -7\*-7 - 4\*65\*31

D = -8011

6 if -8011 > 0: --- False

10 elif -8011 == 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(246, 4) = 26

1 def remove\_digit(number=246, digit=4)

2 res = 0

3 power = 1

4 while 246 > 0: --- True

5 cur\_digit = 246 % 10

cur\_digit = 6

6 if 6 != 4: --- True

7 res = 0 + 6 \* 1

res = 6

8 power = 1 \* 10

power = 10

9 number = 246 // 10

number = 24

4 while 24 > 0: --- True

5 cur\_digit = 24 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 24 // 10

number = 2

4 while 2 > 0: --- True

5 cur\_digit = 2 % 10

cur\_digit = 2

6 if 2 != 4: --- True

7 res = 6 + 2 \* 10

res = 26

8 power = 10 \* 10

power = 100

9 number = 2 // 10

number = 0

4 while 0 > 0: --- False

10 return 26

Вариант: 1-3-16

1. gcd(70, 77) = 7

1 def gcd(x=70, y=77)

2 if 70 < 0: --- False

4 if 77 < 0: --- False

6 while 77 != 0: --- True

7 rem = 70 % 77

rem = 70

8 x = 77

9 y = 70

6 while 70 != 0: --- True

7 rem = 77 % 70

rem = 7

8 x = 70

9 y = 7

6 while 7 != 0: --- True

7 rem = 70 % 7

rem = 0

8 x = 7

9 y = 0

6 while 0 != 0: --- False

10 return 7

2. gcd(0, 25) = 25

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

2 if 0 < 0: --- False

4 if 25 < 0: --- False

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(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(0, 5, -66) = '13.2'

3 def square\_equal(a=0, b=5, c=-66)

4 if 0 != 0: --- False

14 else:

15 if 5 != 0: --- True

16 return str(--66 / 5)

return '13.2'

5. square\_equal(-36, -37, -12) = 'no roots'

3 def square\_equal(a=-36, b=-37, c=-12)

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

5 D = -37\*-37 - 4\*-36\*-12

D = -359

6 if -359 > 0: --- False

10 elif -359 == 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(251, 5) = 21

1 def remove\_digit(number=251, digit=5)

2 res = 0

3 power = 1

4 while 251 > 0: --- True

5 cur\_digit = 251 % 10

cur\_digit = 1

6 if 1 != 5: --- True

7 res = 0 + 1 \* 1

res = 1

8 power = 1 \* 10

power = 10

9 number = 251 // 10

number = 25

4 while 25 > 0: --- True

5 cur\_digit = 25 % 10

cur\_digit = 5

6 if 5 != 5: --- False

9 number = 25 // 10

number = 2

4 while 2 > 0: --- True

5 cur\_digit = 2 % 10

cur\_digit = 2

6 if 2 != 5: --- 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: 1-3-17

1. gcd(42, 81) = 3

1 def gcd(x=42, y=81)

2 if 42 < 0: --- False

4 if 81 < 0: --- False

6 while 81 != 0: --- True

7 rem = 42 % 81

rem = 42

8 x = 81

9 y = 42

6 while 42 != 0: --- True

7 rem = 81 % 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, 72) = 72

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

2 if 0 < 0: --- False

4 if 72 < 0: --- False

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(204) = 'CC'

1 def hex(number=204)

2 if 204 == 0: --- False

4 res = ''

5 while 204 > 0: --- True

6 digit = 204 % 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 = 204 // 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' + 'C'

res = 'CC'

24 number = 12 // 16

number = 0

5 while 0 > 0: --- False

25 return 'CC'

4. square\_equal(0, -2, 20) = '10.0'

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

4 if 0 != 0: --- False

14 else:

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

16 return str(-20 / -2)

return '10.0'

5. square\_equal(0, 5, 76) = '-15.2'

3 def square\_equal(a=0, b=5, c=76)

4 if 0 != 0: --- False

14 else:

15 if 5 != 0: --- True

16 return str(-76 / 5)

return '-15.2'

6. factorize(500) = '2\*2\*5\*5\*5'

1 def factorize(n=500)

2 res = ''

3 while 500 > 2 and 500 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 500 // 2

n = 250

3 while 250 > 2 and 250 % 2 == 0: --- True

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

res = '2\*2\*'

5 n = 250 // 2

n = 125

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

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

10 n = 125 // 5

n = 25

7 while 25 > 5: --- True

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

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

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

10 n = 25 // 5

n = 5

7 while 5 > 5: --- False

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

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

7. remove\_digit(936, 6) = 93

1 def remove\_digit(number=936, digit=6)

2 res = 0

3 power = 1

4 while 936 > 0: --- True

5 cur\_digit = 936 % 10

cur\_digit = 6

6 if 6 != 6: --- False

9 number = 936 // 10

number = 93

4 while 93 > 0: --- True

5 cur\_digit = 93 % 10

cur\_digit = 3

6 if 3 != 6: --- True

7 res = 0 + 3 \* 1

res = 3

8 power = 1 \* 10

power = 10

9 number = 93 // 10

number = 9

4 while 9 > 0: --- True

5 cur\_digit = 9 % 10

cur\_digit = 9

6 if 9 != 6: --- True

7 res = 3 + 9 \* 10

res = 93

8 power = 10 \* 10

power = 100

9 number = 9 // 10

number = 0

4 while 0 > 0: --- False

10 return 93

Вариант: 1-3-18

1. gcd(-15, 50) = 5

1 def gcd(x=-15, y=50)

2 if -15 < 0: --- True

3 x = --15

x = 15

4 if 50 < 0: --- False

6 while 50 != 0: --- True

7 rem = 15 % 50

rem = 15

8 x = 50

9 y = 15

6 while 15 != 0: --- True

7 rem = 50 % 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(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(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(0, -4, -19) = '-4.75'

3 def square\_equal(a=0, b=-4, c=-19)

4 if 0 != 0: --- False

14 else:

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

16 return str(--19 / -4)

return '-4.75'

5. square\_equal(42, 28, 16) = 'no roots'

3 def square\_equal(a=42, b=28, c=16)

4 if 42 != 0: --- True

5 D = 28\*28 - 4\*42\*16

D = -1904

6 if -1904 > 0: --- False

10 elif -1904 == 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(987, 8) = 97

1 def remove\_digit(number=987, digit=8)

2 res = 0

3 power = 1

4 while 987 > 0: --- True

5 cur\_digit = 987 % 10

cur\_digit = 7

6 if 7 != 8: --- True

7 res = 0 + 7 \* 1

res = 7

8 power = 1 \* 10

power = 10

9 number = 987 // 10

number = 98

4 while 98 > 0: --- True

5 cur\_digit = 98 % 10

cur\_digit = 8

6 if 8 != 8: --- False

9 number = 98 // 10

number = 9

4 while 9 > 0: --- True

5 cur\_digit = 9 % 10

cur\_digit = 9

6 if 9 != 8: --- True

7 res = 7 + 9 \* 10

res = 97

8 power = 10 \* 10

power = 100

9 number = 9 // 10

number = 0

4 while 0 > 0: --- False

10 return 97

Вариант: 1-3-19

1. gcd(96, -57) = 3

1 def gcd(x=96, y=-57)

2 if 96 < 0: --- False

4 if -57 < 0: --- True

5 y = --57

y = 57

6 while 57 != 0: --- True

7 rem = 96 % 57

rem = 39

8 x = 57

9 y = 39

6 while 39 != 0: --- True

7 rem = 57 % 39

rem = 18

8 x = 39

9 y = 18

6 while 18 != 0: --- True

7 rem = 39 % 18

rem = 3

8 x = 18

9 y = 3

6 while 3 != 0: --- True

7 rem = 18 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(0, -76) = 76

1 def gcd(x=0, y=-76)

2 if 0 < 0: --- False

4 if -76 < 0: --- True

5 y = --76

y = 76

6 while 76 != 0: --- True

7 rem = 0 % 76

rem = 0

8 x = 76

9 y = 0

6 while 0 != 0: --- False

10 return 76

3. hex(201) = 'C9'

1 def hex(number=201)

2 if 201 == 0: --- False

4 res = ''

5 while 201 > 0: --- True

6 digit = 201 % 16

digit = 9

7 if 9 <= 9: --- True

8 digit = str(9)

digit = '9'

23 res = '9' + ''

res = '9'

24 number = 201 // 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' + '9'

res = 'C9'

24 number = 12 // 16

number = 0

5 while 0 > 0: --- False

25 return 'C9'

4. square\_equal(0, -25, 29) = '1.16'

3 def square\_equal(a=0, b=-25, c=29)

4 if 0 != 0: --- False

14 else:

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

16 return str(-29 / -25)

return '1.16'

5. square\_equal(93, -95, 33) = 'no roots'

3 def square\_equal(a=93, b=-95, c=33)

4 if 93 != 0: --- True

5 D = -95\*-95 - 4\*93\*33

D = -3251

6 if -3251 > 0: --- False

10 elif -3251 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(140) = '2\*2\*5\*7'

1 def factorize(n=140)

2 res = ''

3 while 140 > 2 and 140 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 140 // 2

n = 70

3 while 70 > 2 and 70 % 2 == 0: --- True

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

res = '2\*2\*'

5 n = 70 // 2

n = 35

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

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

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

7. remove\_digit(605, 5) = 60

1 def remove\_digit(number=605, digit=5)

2 res = 0

3 power = 1

4 while 605 > 0: --- True

5 cur\_digit = 605 % 10

cur\_digit = 5

6 if 5 != 5: --- False

9 number = 605 // 10

number = 60

4 while 60 > 0: --- True

5 cur\_digit = 60 % 10

cur\_digit = 0

6 if 0 != 5: --- True

7 res = 0 + 0 \* 1

res = 0

8 power = 1 \* 10

power = 10

9 number = 60 // 10

number = 6

4 while 6 > 0: --- True

5 cur\_digit = 6 % 10

cur\_digit = 6

6 if 6 != 5: --- True

7 res = 0 + 6 \* 10

res = 60

8 power = 10 \* 10

power = 100

9 number = 6 // 10

number = 0

4 while 0 > 0: --- False

10 return 60

Вариант: 1-3-20

1. gcd(88, -96) = 8

1 def gcd(x=88, y=-96)

2 if 88 < 0: --- False

4 if -96 < 0: --- True

5 y = --96

y = 96

6 while 96 != 0: --- True

7 rem = 88 % 96

rem = 88

8 x = 96

9 y = 88

6 while 88 != 0: --- True

7 rem = 96 % 88

rem = 8

8 x = 88

9 y = 8

6 while 8 != 0: --- True

7 rem = 88 % 8

rem = 0

8 x = 8

9 y = 0

6 while 0 != 0: --- False

10 return 8

2. gcd(-7, 0) = 7

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

2 if -7 < 0: --- True

3 x = --7

x = 7

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 7

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(16, -56, -95) = '-1.25 and 4.75'

3 def square\_equal(a=16, b=-56, c=-95)

4 if 16 != 0: --- True

5 D = -56\*-56 - 4\*16\*-95

D = 9216

6 if 9216 > 0: --- True

7 x1 = (--56 - sqrt(9216)) / (2\*16)

x1 = -1.25

8 x2 = (--56 + sqrt(9216)) / (2\*16)

x2 = 4.75

9 return str(-1.25) + ' and ' + str(4.75)

return '-1.25 and 4.75'

5. square\_equal(-89, 11, -28) = 'no roots'

3 def square\_equal(a=-89, b=11, c=-28)

4 if -89 != 0: --- True

5 D = 11\*11 - 4\*-89\*-28

D = -9847

6 if -9847 > 0: --- False

10 elif -9847 == 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(894, 4) = 89

1 def remove\_digit(number=894, digit=4)

2 res = 0

3 power = 1

4 while 894 > 0: --- True

5 cur\_digit = 894 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 894 // 10

number = 89

4 while 89 > 0: --- True

5 cur\_digit = 89 % 10

cur\_digit = 9

6 if 9 != 4: --- True

7 res = 0 + 9 \* 1

res = 9

8 power = 1 \* 10

power = 10

9 number = 89 // 10

number = 8

4 while 8 > 0: --- True

5 cur\_digit = 8 % 10

cur\_digit = 8

6 if 8 != 4: --- True

7 res = 9 + 8 \* 10

res = 89

8 power = 10 \* 10

power = 100

9 number = 8 // 10

number = 0

4 while 0 > 0: --- False

10 return 89

Вариант: 1-3-21

1. gcd(-55, 20) = 5

1 def gcd(x=-55, y=20)

2 if -55 < 0: --- True

3 x = --55

x = 55

4 if 20 < 0: --- False

6 while 20 != 0: --- True

7 rem = 55 % 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: --- True

3 x = --82

x = 82

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 82

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(0, 75, 75) = '-1.0'

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

4 if 0 != 0: --- False

14 else:

15 if 75 != 0: --- True

16 return str(-75 / 75)

return '-1.0'

5. square\_equal(-33, -40, -53) = 'no roots'

3 def square\_equal(a=-33, b=-40, c=-53)

4 if -33 != 0: --- True

5 D = -40\*-40 - 4\*-33\*-53

D = -5396

6 if -5396 > 0: --- False

10 elif -5396 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(12) = '2\*2\*3'

1 def factorize(n=12)

2 res = ''

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

4 res = '' + '2\*'

res = '2\*'

5 n = 12 // 2

n = 6

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

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

res = '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\*' + str(3)

return '2\*2\*3'

7. remove\_digit(986, 6) = 98

1 def remove\_digit(number=986, digit=6)

2 res = 0

3 power = 1

4 while 986 > 0: --- True

5 cur\_digit = 986 % 10

cur\_digit = 6

6 if 6 != 6: --- False

9 number = 986 // 10

number = 98

4 while 98 > 0: --- True

5 cur\_digit = 98 % 10

cur\_digit = 8

6 if 8 != 6: --- True

7 res = 0 + 8 \* 1

res = 8

8 power = 1 \* 10

power = 10

9 number = 98 // 10

number = 9

4 while 9 > 0: --- True

5 cur\_digit = 9 % 10

cur\_digit = 9

6 if 9 != 6: --- True

7 res = 8 + 9 \* 10

res = 98

8 power = 10 \* 10

power = 100

9 number = 9 // 10

number = 0

4 while 0 > 0: --- False

10 return 98

Вариант: 1-3-22

1. gcd(96, 99) = 3

1 def gcd(x=96, y=99)

2 if 96 < 0: --- False

4 if 99 < 0: --- False

6 while 99 != 0: --- True

7 rem = 96 % 99

rem = 96

8 x = 99

9 y = 96

6 while 96 != 0: --- True

7 rem = 99 % 96

rem = 3

8 x = 96

9 y = 3

6 while 3 != 0: --- True

7 rem = 96 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(0, 38) = 38

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

2 if 0 < 0: --- False

4 if 38 < 0: --- False

6 while 38 != 0: --- True

7 rem = 0 % 38

rem = 0

8 x = 38

9 y = 0

6 while 0 != 0: --- False

10 return 38

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, -40, -40) = '-1.0'

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

4 if 0 != 0: --- False

14 else:

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

16 return str(--40 / -40)

return '-1.0'

5. square\_equal(46, -90, 52) = 'no roots'

3 def square\_equal(a=46, b=-90, c=52)

4 if 46 != 0: --- True

5 D = -90\*-90 - 4\*46\*52

D = -1468

6 if -1468 > 0: --- False

10 elif -1468 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(245) = '5\*7\*7'

1 def factorize(n=245)

2 res = ''

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

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

res = '5\*7\*'

10 n = 49 // 7

n = 7

7 while 7 > 7: --- False

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

return '5\*7\*7'

7. remove\_digit(127, 7) = 12

1 def remove\_digit(number=127, digit=7)

2 res = 0

3 power = 1

4 while 127 > 0: --- True

5 cur\_digit = 127 % 10

cur\_digit = 7

6 if 7 != 7: --- False

9 number = 127 // 10

number = 12

4 while 12 > 0: --- True

5 cur\_digit = 12 % 10

cur\_digit = 2

6 if 2 != 7: --- True

7 res = 0 + 2 \* 1

res = 2

8 power = 1 \* 10

power = 10

9 number = 12 // 10

number = 1

4 while 1 > 0: --- True

5 cur\_digit = 1 % 10

cur\_digit = 1

6 if 1 != 7: --- True

7 res = 2 + 1 \* 10

res = 12

8 power = 10 \* 10

power = 100

9 number = 1 // 10

number = 0

4 while 0 > 0: --- False

10 return 12

Вариант: 1-3-23

1. gcd(36, -80) = 4

1 def gcd(x=36, y=-80)

2 if 36 < 0: --- False

4 if -80 < 0: --- True

5 y = --80

y = 80

6 while 80 != 0: --- True

7 rem = 36 % 80

rem = 36

8 x = 80

9 y = 36

6 while 36 != 0: --- True

7 rem = 80 % 36

rem = 8

8 x = 36

9 y = 8

6 while 8 != 0: --- True

7 rem = 36 % 8

rem = 4

8 x = 8

9 y = 4

6 while 4 != 0: --- True

7 rem = 8 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(30, 0) = 30

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

2 if 30 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 30

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(0, -75, 78) = '1.04'

3 def square\_equal(a=0, b=-75, c=78)

4 if 0 != 0: --- False

14 else:

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

16 return str(-78 / -75)

return '1.04'

5. square\_equal(-21, -16, -44) = 'no roots'

3 def square\_equal(a=-21, b=-16, c=-44)

4 if -21 != 0: --- True

5 D = -16\*-16 - 4\*-21\*-44

D = -3440

6 if -3440 > 0: --- False

10 elif -3440 == 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(237, 3) = 27

1 def remove\_digit(number=237, digit=3)

2 res = 0

3 power = 1

4 while 237 > 0: --- True

5 cur\_digit = 237 % 10

cur\_digit = 7

6 if 7 != 3: --- True

7 res = 0 + 7 \* 1

res = 7

8 power = 1 \* 10

power = 10

9 number = 237 // 10

number = 23

4 while 23 > 0: --- True

5 cur\_digit = 23 % 10

cur\_digit = 3

6 if 3 != 3: --- False

9 number = 23 // 10

number = 2

4 while 2 > 0: --- True

5 cur\_digit = 2 % 10

cur\_digit = 2

6 if 2 != 3: --- True

7 res = 7 + 2 \* 10

res = 27

8 power = 10 \* 10

power = 100

9 number = 2 // 10

number = 0

4 while 0 > 0: --- False

10 return 27

Вариант: 1-3-24

1. gcd(88, 33) = 11

1 def gcd(x=88, y=33)

2 if 88 < 0: --- False

4 if 33 < 0: --- False

6 while 33 != 0: --- True

7 rem = 88 % 33

rem = 22

8 x = 33

9 y = 22

6 while 22 != 0: --- True

7 rem = 33 % 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(33, 0) = 33

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

2 if 33 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 33

3. hex(167) = 'A7'

1 def hex(number=167)

2 if 167 == 0: --- False

4 res = ''

5 while 167 > 0: --- True

6 digit = 167 % 16

digit = 7

7 if 7 <= 9: --- True

8 digit = str(7)

digit = '7'

23 res = '7' + ''

res = '7'

24 number = 167 // 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' + '7'

res = 'A7'

24 number = 10 // 16

number = 0

5 while 0 > 0: --- False

25 return 'A7'

4. square\_equal(32, 88, 36) = '-2.25 and -0.5'

3 def square\_equal(a=32, b=88, c=36)

4 if 32 != 0: --- True

5 D = 88\*88 - 4\*32\*36

D = 3136

6 if 3136 > 0: --- True

7 x1 = (-88 - sqrt(3136)) / (2\*32)

x1 = -2.25

8 x2 = (-88 + sqrt(3136)) / (2\*32)

x2 = -0.5

9 return str(-2.25) + ' and ' + str(-0.5)

return '-2.25 and -0.5'

5. square\_equal(-53, 87, -58) = 'no roots'

3 def square\_equal(a=-53, b=87, c=-58)

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

5 D = 87\*87 - 4\*-53\*-58

D = -4727

6 if -4727 > 0: --- False

10 elif -4727 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(375) = '3\*5\*5\*5'

1 def factorize(n=375)

2 res = ''

3 while 375 > 2 and 375 % 2 == 0: --- False

6 d = 3

7 while 375 > 3: --- True

8 if 375 % 3 == 0: --- True

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

res = '3\*'

10 n = 375 // 3

n = 125

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

res = '3\*5\*'

10 n = 125 // 5

n = 25

7 while 25 > 5: --- True

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

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

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

10 n = 25 // 5

n = 5

7 while 5 > 5: --- False

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

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

7. remove\_digit(872, 7) = 82

1 def remove\_digit(number=872, digit=7)

2 res = 0

3 power = 1

4 while 872 > 0: --- True

5 cur\_digit = 872 % 10

cur\_digit = 2

6 if 2 != 7: --- True

7 res = 0 + 2 \* 1

res = 2

8 power = 1 \* 10

power = 10

9 number = 872 // 10

number = 87

4 while 87 > 0: --- True

5 cur\_digit = 87 % 10

cur\_digit = 7

6 if 7 != 7: --- False

9 number = 87 // 10

number = 8

4 while 8 > 0: --- True

5 cur\_digit = 8 % 10

cur\_digit = 8

6 if 8 != 7: --- True

7 res = 2 + 8 \* 10

res = 82

8 power = 10 \* 10

power = 100

9 number = 8 // 10

number = 0

4 while 0 > 0: --- False

10 return 82

Вариант: 1-3-25

1. gcd(70, -55) = 5

1 def gcd(x=70, y=-55)

2 if 70 < 0: --- False

4 if -55 < 0: --- True

5 y = --55

y = 55

6 while 55 != 0: --- True

7 rem = 70 % 55

rem = 15

8 x = 55

9 y = 15

6 while 15 != 0: --- True

7 rem = 55 % 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, 31) = 31

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

2 if 0 < 0: --- False

4 if 31 < 0: --- False

6 while 31 != 0: --- True

7 rem = 0 % 31

rem = 0

8 x = 31

9 y = 0

6 while 0 != 0: --- False

10 return 31

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(0, 25, -73) = '2.92'

3 def square\_equal(a=0, b=25, c=-73)

4 if 0 != 0: --- False

14 else:

15 if 25 != 0: --- True

16 return str(--73 / 25)

return '2.92'

5. square\_equal(-13, 26, -26) = 'no roots'

3 def square\_equal(a=-13, b=26, c=-26)

4 if -13 != 0: --- True

5 D = 26\*26 - 4\*-13\*-26

D = -676

6 if -676 > 0: --- False

10 elif -676 == 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(367, 6) = 37

1 def remove\_digit(number=367, digit=6)

2 res = 0

3 power = 1

4 while 367 > 0: --- True

5 cur\_digit = 367 % 10

cur\_digit = 7

6 if 7 != 6: --- True

7 res = 0 + 7 \* 1

res = 7

8 power = 1 \* 10

power = 10

9 number = 367 // 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 = 7 + 3 \* 10

res = 37

8 power = 10 \* 10

power = 100

9 number = 3 // 10

number = 0

4 while 0 > 0: --- False

10 return 37

Вариант: 1-3-26

1. gcd(9, 66) = 3

1 def gcd(x=9, y=66)

2 if 9 < 0: --- False

4 if 66 < 0: --- False

6 while 66 != 0: --- True

7 rem = 9 % 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(-95, 0) = 95

1 def gcd(x=-95, y=0)

2 if -95 < 0: --- True

3 x = --95

x = 95

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 95

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(0, 4, 56) = '-14.0'

3 def square\_equal(a=0, b=4, c=56)

4 if 0 != 0: --- False

14 else:

15 if 4 != 0: --- True

16 return str(-56 / 4)

return '-14.0'

5. square\_equal(-87, 85, -26) = 'no roots'

3 def square\_equal(a=-87, b=85, c=-26)

4 if -87 != 0: --- True

5 D = 85\*85 - 4\*-87\*-26

D = -1823

6 if -1823 > 0: --- False

10 elif -1823 == 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(804, 4) = 80

1 def remove\_digit(number=804, digit=4)

2 res = 0

3 power = 1

4 while 804 > 0: --- True

5 cur\_digit = 804 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 804 // 10

number = 80

4 while 80 > 0: --- True

5 cur\_digit = 80 % 10

cur\_digit = 0

6 if 0 != 4: --- 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 != 4: --- 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: 1-3-27

1. gcd(-60, 63) = 3

1 def gcd(x=-60, y=63)

2 if -60 < 0: --- True

3 x = --60

x = 60

4 if 63 < 0: --- False

6 while 63 != 0: --- True

7 rem = 60 % 63

rem = 60

8 x = 63

9 y = 60

6 while 60 != 0: --- True

7 rem = 63 % 60

rem = 3

8 x = 60

9 y = 3

6 while 3 != 0: --- True

7 rem = 60 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

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(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(0, -85, -68) = '-0.8'

3 def square\_equal(a=0, b=-85, c=-68)

4 if 0 != 0: --- False

14 else:

15 if -85 != 0: --- True

16 return str(--68 / -85)

return '-0.8'

5. square\_equal(49, 17, 46) = 'no roots'

3 def square\_equal(a=49, b=17, c=46)

4 if 49 != 0: --- True

5 D = 17\*17 - 4\*49\*46

D = -8727

6 if -8727 > 0: --- False

10 elif -8727 == 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(837, 7) = 83

1 def remove\_digit(number=837, digit=7)

2 res = 0

3 power = 1

4 while 837 > 0: --- True

5 cur\_digit = 837 % 10

cur\_digit = 7

6 if 7 != 7: --- False

9 number = 837 // 10

number = 83

4 while 83 > 0: --- True

5 cur\_digit = 83 % 10

cur\_digit = 3

6 if 3 != 7: --- True

7 res = 0 + 3 \* 1

res = 3

8 power = 1 \* 10

power = 10

9 number = 83 // 10

number = 8

4 while 8 > 0: --- True

5 cur\_digit = 8 % 10

cur\_digit = 8

6 if 8 != 7: --- 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: 1-3-28

1. gcd(68, -44) = 4

1 def gcd(x=68, y=-44)

2 if 68 < 0: --- False

4 if -44 < 0: --- True

5 y = --44

y = 44

6 while 44 != 0: --- True

7 rem = 68 % 44

rem = 24

8 x = 44

9 y = 24

6 while 24 != 0: --- True

7 rem = 44 % 24

rem = 20

8 x = 24

9 y = 20

6 while 20 != 0: --- True

7 rem = 24 % 20

rem = 4

8 x = 20

9 y = 4

6 while 4 != 0: --- True

7 rem = 20 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, -31) = 31

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

2 if 0 < 0: --- False

4 if -31 < 0: --- True

5 y = --31

y = 31

6 while 31 != 0: --- True

7 rem = 0 % 31

rem = 0

8 x = 31

9 y = 0

6 while 0 != 0: --- False

10 return 31

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(0, -4, -13) = '-3.25'

3 def square\_equal(a=0, b=-4, c=-13)

4 if 0 != 0: --- False

14 else:

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

16 return str(--13 / -4)

return '-3.25'

5. square\_equal(53, 64, 56) = 'no roots'

3 def square\_equal(a=53, b=64, c=56)

4 if 53 != 0: --- True

5 D = 64\*64 - 4\*53\*56

D = -7776

6 if -7776 > 0: --- False

10 elif -7776 == 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(562, 6) = 52

1 def remove\_digit(number=562, digit=6)

2 res = 0

3 power = 1

4 while 562 > 0: --- True

5 cur\_digit = 562 % 10

cur\_digit = 2

6 if 2 != 6: --- True

7 res = 0 + 2 \* 1

res = 2

8 power = 1 \* 10

power = 10

9 number = 562 // 10

number = 56

4 while 56 > 0: --- True

5 cur\_digit = 56 % 10

cur\_digit = 6

6 if 6 != 6: --- False

9 number = 56 // 10

number = 5

4 while 5 > 0: --- True

5 cur\_digit = 5 % 10

cur\_digit = 5

6 if 5 != 6: --- 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: 1-3-29

1. gcd(51, -57) = 3

1 def gcd(x=51, y=-57)

2 if 51 < 0: --- False

4 if -57 < 0: --- True

5 y = --57

y = 57

6 while 57 != 0: --- True

7 rem = 51 % 57

rem = 51

8 x = 57

9 y = 51

6 while 51 != 0: --- True

7 rem = 57 % 51

rem = 6

8 x = 51

9 y = 6

6 while 6 != 0: --- True

7 rem = 51 % 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, 58) = 58

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

2 if 0 < 0: --- False

4 if 58 < 0: --- False

6 while 58 != 0: --- True

7 rem = 0 % 58

rem = 0

8 x = 58

9 y = 0

6 while 0 != 0: --- False

10 return 58

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(1, 62, 0) = '-62.0 and 0.0'

3 def square\_equal(a=1, b=62, c=0)

4 if 1 != 0: --- True

5 D = 62\*62 - 4\*1\*0

D = 3844

6 if 3844 > 0: --- True

7 x1 = (-62 - sqrt(3844)) / (2\*1)

x1 = -62.0

8 x2 = (-62 + sqrt(3844)) / (2\*1)

x2 = 0.0

9 return str(-62.0) + ' and ' + str(0.0)

return '-62.0 and 0.0'

5. square\_equal(-93, 27, -23) = 'no roots'

3 def square\_equal(a=-93, b=27, c=-23)

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

5 D = 27\*27 - 4\*-93\*-23

D = -7827

6 if -7827 > 0: --- False

10 elif -7827 == 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(454, 5) = 44

1 def remove\_digit(number=454, digit=5)

2 res = 0

3 power = 1

4 while 454 > 0: --- True

5 cur\_digit = 454 % 10

cur\_digit = 4

6 if 4 != 5: --- True

7 res = 0 + 4 \* 1

res = 4

8 power = 1 \* 10

power = 10

9 number = 454 // 10

number = 45

4 while 45 > 0: --- True

5 cur\_digit = 45 % 10

cur\_digit = 5

6 if 5 != 5: --- False

9 number = 45 // 10

number = 4

4 while 4 > 0: --- True

5 cur\_digit = 4 % 10

cur\_digit = 4

6 if 4 != 5: --- 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: 1-3-30

1. gcd(-40, 15) = 5

1 def gcd(x=-40, y=15)

2 if -40 < 0: --- True

3 x = --40

x = 40

4 if 15 < 0: --- False

6 while 15 != 0: --- True

7 rem = 40 % 15

rem = 10

8 x = 15

9 y = 10

6 while 10 != 0: --- True

7 rem = 15 % 10

rem = 5

8 x = 10

9 y = 5

6 while 5 != 0: --- True

7 rem = 10 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

2. gcd(0, -5) = 5

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

2 if 0 < 0: --- False

4 if -5 < 0: --- True

5 y = --5

y = 5

6 while 5 != 0: --- True

7 rem = 0 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

3. hex(220) = 'DC'

1 def hex(number=220)

2 if 220 == 0: --- False

4 res = ''

5 while 220 > 0: --- True

6 digit = 220 % 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 = 220 // 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' + 'C'

res = 'DC'

24 number = 13 // 16

number = 0

5 while 0 > 0: --- False

25 return 'DC'

4. square\_equal(0, 100, 39) = '-0.39'

3 def square\_equal(a=0, b=100, c=39)

4 if 0 != 0: --- False

14 else:

15 if 100 != 0: --- True

16 return str(-39 / 100)

return '-0.39'

5. square\_equal(-15, 29, -62) = 'no roots'

3 def square\_equal(a=-15, b=29, c=-62)

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

5 D = 29\*29 - 4\*-15\*-62

D = -2879

6 if -2879 > 0: --- False

10 elif -2879 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(16) = '2\*2\*2\*2'

1 def factorize(n=16)

2 res = ''

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

4 res = '' + '2\*'

res = '2\*'

5 n = 16 // 2

n = 8

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

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

res = '2\*2\*'

5 n = 8 // 2

n = 4

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

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

res = '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\*' + str(2)

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

7. remove\_digit(2444, 4) = 2

1 def remove\_digit(number=2444, digit=4)

2 res = 0

3 power = 1

4 while 2444 > 0: --- True

5 cur\_digit = 2444 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 2444 // 10

number = 244

4 while 244 > 0: --- True

5 cur\_digit = 244 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 244 // 10

number = 24

4 while 24 > 0: --- True

5 cur\_digit = 24 % 10

cur\_digit = 4

6 if 4 != 4: --- False

9 number = 24 // 10

number = 2

4 while 2 > 0: --- True

5 cur\_digit = 2 % 10

cur\_digit = 2

6 if 2 != 4: --- True

7 res = 0 + 2 \* 1

res = 2

8 power = 1 \* 10

power = 10

9 number = 2 // 10

number = 0

4 while 0 > 0: --- False

10 return 2

Вариант: 1-3-31

1. gcd(-24, 42) = 6

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

2 if -24 < 0: --- True

3 x = --24

x = 24

4 if 42 < 0: --- False

6 while 42 != 0: --- True

7 rem = 24 % 42

rem = 24

8 x = 42

9 y = 24

6 while 24 != 0: --- True

7 rem = 42 % 24

rem = 18

8 x = 24

9 y = 18

6 while 18 != 0: --- True

7 rem = 24 % 18

rem = 6

8 x = 18

9 y = 6

6 while 6 != 0: --- True

7 rem = 18 % 6

rem = 0

8 x = 6

9 y = 0

6 while 0 != 0: --- False

10 return 6

2. gcd(70, 0) = 70

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

2 if 70 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 70

3. hex(202) = 'CA'

1 def hex(number=202)

2 if 202 == 0: --- False

4 res = ''

5 while 202 > 0: --- True

6 digit = 202 % 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 = 202 // 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' + 'A'

res = 'CA'

24 number = 12 // 16

number = 0

5 while 0 > 0: --- False

25 return 'CA'

4. square\_equal(0, -12, -42) = '-3.5'

3 def square\_equal(a=0, b=-12, c=-42)

4 if 0 != 0: --- False

14 else:

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

16 return str(--42 / -12)

return '-3.5'

5. square\_equal(-32, 23, -22) = 'no roots'

3 def square\_equal(a=-32, b=23, c=-22)

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

5 D = 23\*23 - 4\*-32\*-22

D = -2287

6 if -2287 > 0: --- False

10 elif -2287 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(81) = '3\*3\*3\*3'

1 def factorize(n=81)

2 res = ''

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

res = '3\*'

10 n = 81 // 3

n = 27

7 while 27 > 3: --- True

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

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

res = '3\*3\*'

10 n = 27 // 3

n = 9

7 while 9 > 3: --- True

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

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

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

10 n = 9 // 3

n = 3

7 while 3 > 3: --- False

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

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

7. remove\_digit(386, 8) = 36

1 def remove\_digit(number=386, digit=8)

2 res = 0

3 power = 1

4 while 386 > 0: --- True

5 cur\_digit = 386 % 10

cur\_digit = 6

6 if 6 != 8: --- True

7 res = 0 + 6 \* 1

res = 6

8 power = 1 \* 10

power = 10

9 number = 386 // 10

number = 38

4 while 38 > 0: --- True

5 cur\_digit = 38 % 10

cur\_digit = 8

6 if 8 != 8: --- False

9 number = 38 // 10

number = 3

4 while 3 > 0: --- True

5 cur\_digit = 3 % 10

cur\_digit = 3

6 if 3 != 8: --- True

7 res = 6 + 3 \* 10

res = 36

8 power = 10 \* 10

power = 100

9 number = 3 // 10

number = 0

4 while 0 > 0: --- False

10 return 36

Вариант: 1-3-32

1. gcd(12, -28) = 4

1 def gcd(x=12, y=-28)

2 if 12 < 0: --- False

4 if -28 < 0: --- True

5 y = --28

y = 28

6 while 28 != 0: --- True

7 rem = 12 % 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, 57) = 57

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

2 if 0 < 0: --- False

4 if 57 < 0: --- False

6 while 57 != 0: --- True

7 rem = 0 % 57

rem = 0

8 x = 57

9 y = 0

6 while 0 != 0: --- False

10 return 57

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, 8, -50) = '6.25'

3 def square\_equal(a=0, b=8, c=-50)

4 if 0 != 0: --- False

14 else:

15 if 8 != 0: --- True

16 return str(--50 / 8)

return '6.25'

5. square\_equal(-33, -32, -60) = 'no roots'

3 def square\_equal(a=-33, b=-32, c=-60)

4 if -33 != 0: --- True

5 D = -32\*-32 - 4\*-33\*-60

D = -6896

6 if -6896 > 0: --- False

10 elif -6896 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(49) = '7\*7'

1 def factorize(n=49)

2 res = ''

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

res = '7\*'

10 n = 49 // 7

n = 7

7 while 7 > 7: --- False

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

return '7\*7'

7. remove\_digit(186, 8) = 16

1 def remove\_digit(number=186, digit=8)

2 res = 0

3 power = 1

4 while 186 > 0: --- True

5 cur\_digit = 186 % 10

cur\_digit = 6

6 if 6 != 8: --- True

7 res = 0 + 6 \* 1

res = 6

8 power = 1 \* 10

power = 10

9 number = 186 // 10

number = 18

4 while 18 > 0: --- True

5 cur\_digit = 18 % 10

cur\_digit = 8

6 if 8 != 8: --- False

9 number = 18 // 10

number = 1

4 while 1 > 0: --- True

5 cur\_digit = 1 % 10

cur\_digit = 1

6 if 1 != 8: --- True

7 res = 6 + 1 \* 10

res = 16

8 power = 10 \* 10

power = 100

9 number = 1 // 10

number = 0

4 while 0 > 0: --- False

10 return 16

Вариант: 1-3-33

1. gcd(69, 9) = 3

1 def gcd(x=69, y=9)

2 if 69 < 0: --- False

4 if 9 < 0: --- False

6 while 9 != 0: --- True

7 rem = 69 % 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(0, -36) = 36

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

2 if 0 < 0: --- False

4 if -36 < 0: --- True

5 y = --36

y = 36

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(197) = 'C5'

1 def hex(number=197)

2 if 197 == 0: --- False

4 res = ''

5 while 197 > 0: --- True

6 digit = 197 % 16

digit = 5

7 if 5 <= 9: --- True

8 digit = str(5)

digit = '5'

23 res = '5' + ''

res = '5'

24 number = 197 // 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' + '5'

res = 'C5'

24 number = 12 // 16

number = 0

5 while 0 > 0: --- False

25 return 'C5'

4. square\_equal(0, 5, -14) = '2.8'

3 def square\_equal(a=0, b=5, c=-14)

4 if 0 != 0: --- False

14 else:

15 if 5 != 0: --- True

16 return str(--14 / 5)

return '2.8'

5. square\_equal(38, -77, 54) = 'no roots'

3 def square\_equal(a=38, b=-77, c=54)

4 if 38 != 0: --- True

5 D = -77\*-77 - 4\*38\*54

D = -2279

6 if -2279 > 0: --- False

10 elif -2279 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(120) = '2\*2\*2\*3\*5'

1 def factorize(n=120)

2 res = ''

3 while 120 > 2 and 120 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 120 // 2

n = 60

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

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

res = '2\*2\*'

5 n = 60 // 2

n = 30

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

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

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

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

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

7. remove\_digit(704, 0) = 74

1 def remove\_digit(number=704, digit=0)

2 res = 0

3 power = 1

4 while 704 > 0: --- True

5 cur\_digit = 704 % 10

cur\_digit = 4

6 if 4 != 0: --- True

7 res = 0 + 4 \* 1

res = 4

8 power = 1 \* 10

power = 10

9 number = 704 // 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 = 4 + 7 \* 10

res = 74

8 power = 10 \* 10

power = 100

9 number = 7 // 10

number = 0

4 while 0 > 0: --- False

10 return 74

Вариант: 1-3-34

1. gcd(21, -48) = 3

1 def gcd(x=21, y=-48)

2 if 21 < 0: --- False

4 if -48 < 0: --- True

5 y = --48

y = 48

6 while 48 != 0: --- True

7 rem = 21 % 48

rem = 21

8 x = 48

9 y = 21

6 while 21 != 0: --- True

7 rem = 48 % 21

rem = 6

8 x = 21

9 y = 6

6 while 6 != 0: --- True

7 rem = 21 % 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, -35) = 35

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

2 if 0 < 0: --- False

4 if -35 < 0: --- True

5 y = --35

y = 35

6 while 35 != 0: --- True

7 rem = 0 % 35

rem = 0

8 x = 35

9 y = 0

6 while 0 != 0: --- False

10 return 35

3. hex(250) = 'FA'

1 def hex(number=250)

2 if 250 == 0: --- False

4 res = ''

5 while 250 > 0: --- True

6 digit = 250 % 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 = 250 // 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' + 'A'

res = 'FA'

24 number = 15 // 16

number = 0

5 while 0 > 0: --- False

25 return 'FA'

4. square\_equal(0, 3, 84) = '-28.0'

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

4 if 0 != 0: --- False

14 else:

15 if 3 != 0: --- True

16 return str(-84 / 3)

return '-28.0'

5. square\_equal(-81, -87, -39) = 'no roots'

3 def square\_equal(a=-81, b=-87, c=-39)

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

5 D = -87\*-87 - 4\*-81\*-39

D = -5067

6 if -5067 > 0: --- False

10 elif -5067 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(147) = '3\*7\*7'

1 def factorize(n=147)

2 res = ''

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

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

res = '3\*7\*'

10 n = 49 // 7

n = 7

7 while 7 > 7: --- False

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

return '3\*7\*7'

7. remove\_digit(275, 5) = 27

1 def remove\_digit(number=275, digit=5)

2 res = 0

3 power = 1

4 while 275 > 0: --- True

5 cur\_digit = 275 % 10

cur\_digit = 5

6 if 5 != 5: --- False

9 number = 275 // 10

number = 27

4 while 27 > 0: --- True

5 cur\_digit = 27 % 10

cur\_digit = 7

6 if 7 != 5: --- True

7 res = 0 + 7 \* 1

res = 7

8 power = 1 \* 10

power = 10

9 number = 27 // 10

number = 2

4 while 2 > 0: --- True

5 cur\_digit = 2 % 10

cur\_digit = 2

6 if 2 != 5: --- True

7 res = 7 + 2 \* 10

res = 27

8 power = 10 \* 10

power = 100

9 number = 2 // 10

number = 0

4 while 0 > 0: --- False

10 return 27

Вариант: 1-3-35

1. gcd(-6, -63) = 3

1 def gcd(x=-6, y=-63)

2 if -6 < 0: --- True

3 x = --6

x = 6

4 if -63 < 0: --- True

5 y = --63

y = 63

6 while 63 != 0: --- True

7 rem = 6 % 63

rem = 6

8 x = 63

9 y = 6

6 while 6 != 0: --- True

7 rem = 63 % 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(20, 0) = 20

1 def gcd(x=20, y=0)

2 if 20 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 20

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(65, 65, 0) = '-1.0 and 0.0'

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

4 if 65 != 0: --- True

5 D = 65\*65 - 4\*65\*0

D = 4225

6 if 4225 > 0: --- True

7 x1 = (-65 - sqrt(4225)) / (2\*65)

x1 = -1.0

8 x2 = (-65 + sqrt(4225)) / (2\*65)

x2 = 0.0

9 return str(-1.0) + ' and ' + str(0.0)

return '-1.0 and 0.0'

5. square\_equal(-93, 60, -24) = 'no roots'

3 def square\_equal(a=-93, b=60, c=-24)

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

5 D = 60\*60 - 4\*-93\*-24

D = -5328

6 if -5328 > 0: --- False

10 elif -5328 == 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(149, 9) = 14

1 def remove\_digit(number=149, digit=9)

2 res = 0

3 power = 1

4 while 149 > 0: --- True

5 cur\_digit = 149 % 10

cur\_digit = 9

6 if 9 != 9: --- False

9 number = 149 // 10

number = 14

4 while 14 > 0: --- True

5 cur\_digit = 14 % 10

cur\_digit = 4

6 if 4 != 9: --- True

7 res = 0 + 4 \* 1

res = 4

8 power = 1 \* 10

power = 10

9 number = 14 // 10

number = 1

4 while 1 > 0: --- True

5 cur\_digit = 1 % 10

cur\_digit = 1

6 if 1 != 9: --- True

7 res = 4 + 1 \* 10

res = 14

8 power = 10 \* 10

power = 100

9 number = 1 // 10

number = 0

4 while 0 > 0: --- False

10 return 14

Вариант: 1-3-36

1. gcd(9, 84) = 3

1 def gcd(x=9, y=84)

2 if 9 < 0: --- False

4 if 84 < 0: --- False

6 while 84 != 0: --- True

7 rem = 9 % 84

rem = 9

8 x = 84

9 y = 9

6 while 9 != 0: --- True

7 rem = 84 % 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, -39) = 39

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

2 if 0 < 0: --- False

4 if -39 < 0: --- True

5 y = --39

y = 39

6 while 39 != 0: --- True

7 rem = 0 % 39

rem = 0

8 x = 39

9 y = 0

6 while 0 != 0: --- False

10 return 39

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(0, -58, 87) = '1.5'

3 def square\_equal(a=0, b=-58, c=87)

4 if 0 != 0: --- False

14 else:

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

16 return str(-87 / -58)

return '1.5'

5. square\_equal(86, -39, 11) = 'no roots'

3 def square\_equal(a=86, b=-39, c=11)

4 if 86 != 0: --- True

5 D = -39\*-39 - 4\*86\*11

D = -2263

6 if -2263 > 0: --- False

10 elif -2263 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(250) = '2\*5\*5\*5'

1 def factorize(n=250)

2 res = ''

3 while 250 > 2 and 250 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 250 // 2

n = 125

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

res = '2\*5\*'

10 n = 125 // 5

n = 25

7 while 25 > 5: --- True

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

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

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

10 n = 25 // 5

n = 5

7 while 5 > 5: --- False

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

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

7. remove\_digit(422, 2) = 4

1 def remove\_digit(number=422, digit=2)

2 res = 0

3 power = 1

4 while 422 > 0: --- True

5 cur\_digit = 422 % 10

cur\_digit = 2

6 if 2 != 2: --- False

9 number = 422 // 10

number = 42

4 while 42 > 0: --- True

5 cur\_digit = 42 % 10

cur\_digit = 2

6 if 2 != 2: --- False

9 number = 42 // 10

number = 4

4 while 4 > 0: --- True

5 cur\_digit = 4 % 10

cur\_digit = 4

6 if 4 != 2: --- True

7 res = 0 + 4 \* 1

res = 4

8 power = 1 \* 10

power = 10

9 number = 4 // 10

number = 0

4 while 0 > 0: --- False

10 return 4

Вариант: 1-3-37

1. gcd(-60, -78) = 6

1 def gcd(x=-60, y=-78)

2 if -60 < 0: --- True

3 x = --60

x = 60

4 if -78 < 0: --- True

5 y = --78

y = 78

6 while 78 != 0: --- True

7 rem = 60 % 78

rem = 60

8 x = 78

9 y = 60

6 while 60 != 0: --- True

7 rem = 78 % 60

rem = 18

8 x = 60

9 y = 18

6 while 18 != 0: --- True

7 rem = 60 % 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, -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(228) = 'E4'

1 def hex(number=228)

2 if 228 == 0: --- False

4 res = ''

5 while 228 > 0: --- True

6 digit = 228 % 16

digit = 4

7 if 4 <= 9: --- True

8 digit = str(4)

digit = '4'

23 res = '4' + ''

res = '4'

24 number = 228 // 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' + '4'

res = 'E4'

24 number = 14 // 16

number = 0

5 while 0 > 0: --- False

25 return 'E4'

4. square\_equal(-60, -36, 0) = '-0.0 and -0.6'

3 def square\_equal(a=-60, b=-36, c=0)

4 if -60 != 0: --- True

5 D = -36\*-36 - 4\*-60\*0

D = 1296

6 if 1296 > 0: --- True

7 x1 = (--36 - sqrt(1296)) / (2\*-60)

x1 = -0.0

8 x2 = (--36 + sqrt(1296)) / (2\*-60)

x2 = -0.6

9 return str(-0.0) + ' and ' + str(-0.6)

return '-0.0 and -0.6'

5. square\_equal(31, 75, 58) = 'no roots'

3 def square\_equal(a=31, b=75, c=58)

4 if 31 != 0: --- True

5 D = 75\*75 - 4\*31\*58

D = -1567

6 if -1567 > 0: --- False

10 elif -1567 == 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(376, 7) = 36

1 def remove\_digit(number=376, digit=7)

2 res = 0

3 power = 1

4 while 376 > 0: --- True

5 cur\_digit = 376 % 10

cur\_digit = 6

6 if 6 != 7: --- True

7 res = 0 + 6 \* 1

res = 6

8 power = 1 \* 10

power = 10

9 number = 376 // 10

number = 37

4 while 37 > 0: --- True

5 cur\_digit = 37 % 10

cur\_digit = 7

6 if 7 != 7: --- False

9 number = 37 // 10

number = 3

4 while 3 > 0: --- True

5 cur\_digit = 3 % 10

cur\_digit = 3

6 if 3 != 7: --- True

7 res = 6 + 3 \* 10

res = 36

8 power = 10 \* 10

power = 100

9 number = 3 // 10

number = 0

4 while 0 > 0: --- False

10 return 36

Вариант: 1-3-38

1. gcd(69, -57) = 3

1 def gcd(x=69, y=-57)

2 if 69 < 0: --- False

4 if -57 < 0: --- True

5 y = --57

y = 57

6 while 57 != 0: --- True

7 rem = 69 % 57

rem = 12

8 x = 57

9 y = 12

6 while 12 != 0: --- True

7 rem = 57 % 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(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(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(0, -10, 36) = '3.6'

3 def square\_equal(a=0, b=-10, c=36)

4 if 0 != 0: --- False

14 else:

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

16 return str(-36 / -10)

return '3.6'

5. square\_equal(19, 53, 83) = 'no roots'

3 def square\_equal(a=19, b=53, c=83)

4 if 19 != 0: --- True

5 D = 53\*53 - 4\*19\*83

D = -3499

6 if -3499 > 0: --- False

10 elif -3499 == 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(451, 5) = 41

1 def remove\_digit(number=451, digit=5)

2 res = 0

3 power = 1

4 while 451 > 0: --- True

5 cur\_digit = 451 % 10

cur\_digit = 1

6 if 1 != 5: --- True

7 res = 0 + 1 \* 1

res = 1

8 power = 1 \* 10

power = 10

9 number = 451 // 10

number = 45

4 while 45 > 0: --- True

5 cur\_digit = 45 % 10

cur\_digit = 5

6 if 5 != 5: --- False

9 number = 45 // 10

number = 4

4 while 4 > 0: --- True

5 cur\_digit = 4 % 10

cur\_digit = 4

6 if 4 != 5: --- True

7 res = 1 + 4 \* 10

res = 41

8 power = 10 \* 10

power = 100

9 number = 4 // 10

number = 0

4 while 0 > 0: --- False

10 return 41

Вариант: 1-3-39

1. gcd(99, 15) = 3

1 def gcd(x=99, y=15)

2 if 99 < 0: --- False

4 if 15 < 0: --- False

6 while 15 != 0: --- True

7 rem = 99 % 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(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(213) = 'D5'

1 def hex(number=213)

2 if 213 == 0: --- False

4 res = ''

5 while 213 > 0: --- True

6 digit = 213 % 16

digit = 5

7 if 5 <= 9: --- True

8 digit = str(5)

digit = '5'

23 res = '5' + ''

res = '5'

24 number = 213 // 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' + '5'

res = 'D5'

24 number = 13 // 16

number = 0

5 while 0 > 0: --- False

25 return 'D5'

4. square\_equal(0, 2, 18) = '-9.0'

3 def square\_equal(a=0, b=2, c=18)

4 if 0 != 0: --- False

14 else:

15 if 2 != 0: --- True

16 return str(-18 / 2)

return '-9.0'

5. square\_equal(42, -88, 68) = 'no roots'

3 def square\_equal(a=42, b=-88, c=68)

4 if 42 != 0: --- True

5 D = -88\*-88 - 4\*42\*68

D = -3680

6 if -3680 > 0: --- False

10 elif -3680 == 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(471, 1) = 47

1 def remove\_digit(number=471, digit=1)

2 res = 0

3 power = 1

4 while 471 > 0: --- True

5 cur\_digit = 471 % 10

cur\_digit = 1

6 if 1 != 1: --- False

9 number = 471 // 10

number = 47

4 while 47 > 0: --- True

5 cur\_digit = 47 % 10

cur\_digit = 7

6 if 7 != 1: --- 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 != 1: --- 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: 1-3-40

1. gcd(18, 45) = 9

1 def gcd(x=18, y=45)

2 if 18 < 0: --- False

4 if 45 < 0: --- False

6 while 45 != 0: --- True

7 rem = 18 % 45

rem = 18

8 x = 45

9 y = 18

6 while 18 != 0: --- True

7 rem = 45 % 18

rem = 9

8 x = 18

9 y = 9

6 while 9 != 0: --- True

7 rem = 18 % 9

rem = 0

8 x = 9

9 y = 0

6 while 0 != 0: --- False

10 return 9

2. gcd(-2, 0) = 2

1 def gcd(x=-2, y=0)

2 if -2 < 0: --- True

3 x = --2

x = 2

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 2

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(0, 2, 43) = '-21.5'

3 def square\_equal(a=0, b=2, c=43)

4 if 0 != 0: --- False

14 else:

15 if 2 != 0: --- True

16 return str(-43 / 2)

return '-21.5'

5. square\_equal(75, -55, 34) = 'no roots'

3 def square\_equal(a=75, b=-55, c=34)

4 if 75 != 0: --- True

5 D = -55\*-55 - 4\*75\*34

D = -7175

6 if -7175 > 0: --- False

10 elif -7175 == 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(871, 7) = 81

1 def remove\_digit(number=871, digit=7)

2 res = 0

3 power = 1

4 while 871 > 0: --- True

5 cur\_digit = 871 % 10

cur\_digit = 1

6 if 1 != 7: --- True

7 res = 0 + 1 \* 1

res = 1

8 power = 1 \* 10

power = 10

9 number = 871 // 10

number = 87

4 while 87 > 0: --- True

5 cur\_digit = 87 % 10

cur\_digit = 7

6 if 7 != 7: --- False

9 number = 87 // 10

number = 8

4 while 8 > 0: --- True

5 cur\_digit = 8 % 10

cur\_digit = 8

6 if 8 != 7: --- True

7 res = 1 + 8 \* 10

res = 81

8 power = 10 \* 10

power = 100

9 number = 8 // 10

number = 0

4 while 0 > 0: --- False

10 return 81

Вариант: 1-3-41

1. gcd(-45, -55) = 5

1 def gcd(x=-45, y=-55)

2 if -45 < 0: --- True

3 x = --45

x = 45

4 if -55 < 0: --- True

5 y = --55

y = 55

6 while 55 != 0: --- True

7 rem = 45 % 55

rem = 45

8 x = 55

9 y = 45

6 while 45 != 0: --- True

7 rem = 55 % 45

rem = 10

8 x = 45

9 y = 10

6 while 10 != 0: --- True

7 rem = 45 % 10

rem = 5

8 x = 10

9 y = 5

6 while 5 != 0: --- True

7 rem = 10 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

2. gcd(0, 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(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, 96, -48) = '0.5'

3 def square\_equal(a=0, b=96, c=-48)

4 if 0 != 0: --- False

14 else:

15 if 96 != 0: --- True

16 return str(--48 / 96)

return '0.5'

5. square\_equal(0, -8, 4) = '0.5'

3 def square\_equal(a=0, b=-8, c=4)

4 if 0 != 0: --- False

14 else:

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

16 return str(-4 / -8)

return '0.5'

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(923, 2) = 93

1 def remove\_digit(number=923, digit=2)

2 res = 0

3 power = 1

4 while 923 > 0: --- True

5 cur\_digit = 923 % 10

cur\_digit = 3

6 if 3 != 2: --- True

7 res = 0 + 3 \* 1

res = 3

8 power = 1 \* 10

power = 10

9 number = 923 // 10

number = 92

4 while 92 > 0: --- True

5 cur\_digit = 92 % 10

cur\_digit = 2

6 if 2 != 2: --- False

9 number = 92 // 10

number = 9

4 while 9 > 0: --- True

5 cur\_digit = 9 % 10

cur\_digit = 9

6 if 9 != 2: --- True

7 res = 3 + 9 \* 10

res = 93

8 power = 10 \* 10

power = 100

9 number = 9 // 10

number = 0

4 while 0 > 0: --- False

10 return 93

Вариант: 1-3-42

1. gcd(18, 75) = 3

1 def gcd(x=18, y=75)

2 if 18 < 0: --- False

4 if 75 < 0: --- False

6 while 75 != 0: --- True

7 rem = 18 % 75

rem = 18

8 x = 75

9 y = 18

6 while 18 != 0: --- True

7 rem = 75 % 18

rem = 3

8 x = 18

9 y = 3

6 while 3 != 0: --- True

7 rem = 18 % 3

rem = 0

8 x = 3

9 y = 0

6 while 0 != 0: --- False

10 return 3

2. gcd(-39, 0) = 39

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

2 if -39 < 0: --- True

3 x = --39

x = 39

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 39

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(0, 100, -39) = '0.39'

3 def square\_equal(a=0, b=100, c=-39)

4 if 0 != 0: --- False

14 else:

15 if 100 != 0: --- True

16 return str(--39 / 100)

return '0.39'

5. square\_equal(-17, -29, -19) = 'no roots'

3 def square\_equal(a=-17, b=-29, c=-19)

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

5 D = -29\*-29 - 4\*-17\*-19

D = -451

6 if -451 > 0: --- False

10 elif -451 == 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(433, 3) = 4

1 def remove\_digit(number=433, digit=3)

2 res = 0

3 power = 1

4 while 433 > 0: --- True

5 cur\_digit = 433 % 10

cur\_digit = 3

6 if 3 != 3: --- False

9 number = 433 // 10

number = 43

4 while 43 > 0: --- True

5 cur\_digit = 43 % 10

cur\_digit = 3

6 if 3 != 3: --- False

9 number = 43 // 10

number = 4

4 while 4 > 0: --- True

5 cur\_digit = 4 % 10

cur\_digit = 4

6 if 4 != 3: --- True

7 res = 0 + 4 \* 1

res = 4

8 power = 1 \* 10

power = 10

9 number = 4 // 10

number = 0

4 while 0 > 0: --- False

10 return 4

Вариант: 1-3-43

1. gcd(56, -98) = 14

1 def gcd(x=56, y=-98)

2 if 56 < 0: --- False

4 if -98 < 0: --- True

5 y = --98

y = 98

6 while 98 != 0: --- True

7 rem = 56 % 98

rem = 56

8 x = 98

9 y = 56

6 while 56 != 0: --- True

7 rem = 98 % 56

rem = 42

8 x = 56

9 y = 42

6 while 42 != 0: --- True

7 rem = 56 % 42

rem = 14

8 x = 42

9 y = 14

6 while 14 != 0: --- True

7 rem = 42 % 14

rem = 0

8 x = 14

9 y = 0

6 while 0 != 0: --- False

10 return 14

2. gcd(0, -23) = 23

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

2 if 0 < 0: --- False

4 if -23 < 0: --- True

5 y = --23

y = 23

6 while 23 != 0: --- True

7 rem = 0 % 23

rem = 0

8 x = 23

9 y = 0

6 while 0 != 0: --- False

10 return 23

3. hex(203) = 'CB'

1 def hex(number=203)

2 if 203 == 0: --- False

4 res = ''

5 while 203 > 0: --- True

6 digit = 203 % 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 = 203 // 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' + 'B'

res = 'CB'

24 number = 12 // 16

number = 0

5 while 0 > 0: --- False

25 return 'CB'

4. square\_equal(-1, -16, 0) = '-0.0 and -16.0'

3 def square\_equal(a=-1, b=-16, c=0)

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

5 D = -16\*-16 - 4\*-1\*0

D = 256

6 if 256 > 0: --- True

7 x1 = (--16 - sqrt(256)) / (2\*-1)

x1 = -0.0

8 x2 = (--16 + sqrt(256)) / (2\*-1)

x2 = -16.0

9 return str(-0.0) + ' and ' + str(-16.0)

return '-0.0 and -16.0'

5. square\_equal(-64, -20, -18) = 'no roots'

3 def square\_equal(a=-64, b=-20, c=-18)

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

5 D = -20\*-20 - 4\*-64\*-18

D = -4208

6 if -4208 > 0: --- False

10 elif -4208 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(84) = '2\*2\*3\*7'

1 def factorize(n=84)

2 res = ''

3 while 84 > 2 and 84 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 84 // 2

n = 42

3 while 42 > 2 and 42 % 2 == 0: --- True

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

res = '2\*2\*'

5 n = 42 // 2

n = 21

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

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

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

7. remove\_digit(819, 1) = 89

1 def remove\_digit(number=819, digit=1)

2 res = 0

3 power = 1

4 while 819 > 0: --- True

5 cur\_digit = 819 % 10

cur\_digit = 9

6 if 9 != 1: --- True

7 res = 0 + 9 \* 1

res = 9

8 power = 1 \* 10

power = 10

9 number = 819 // 10

number = 81

4 while 81 > 0: --- True

5 cur\_digit = 81 % 10

cur\_digit = 1

6 if 1 != 1: --- False

9 number = 81 // 10

number = 8

4 while 8 > 0: --- True

5 cur\_digit = 8 % 10

cur\_digit = 8

6 if 8 != 1: --- True

7 res = 9 + 8 \* 10

res = 89

8 power = 10 \* 10

power = 100

9 number = 8 // 10

number = 0

4 while 0 > 0: --- False

10 return 89

Вариант: 1-3-44

1. gcd(-100, 35) = 5

1 def gcd(x=-100, y=35)

2 if -100 < 0: --- True

3 x = --100

x = 100

4 if 35 < 0: --- False

6 while 35 != 0: --- True

7 rem = 100 % 35

rem = 30

8 x = 35

9 y = 30

6 while 30 != 0: --- True

7 rem = 35 % 30

rem = 5

8 x = 30

9 y = 5

6 while 5 != 0: --- True

7 rem = 30 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

2. gcd(23, 0) = 23

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

2 if 23 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 23

3. hex(208) = 'D0'

1 def hex(number=208)

2 if 208 == 0: --- False

4 res = ''

5 while 208 > 0: --- True

6 digit = 208 % 16

digit = 0

7 if 0 <= 9: --- True

8 digit = str(0)

digit = '0'

23 res = '0' + ''

res = '0'

24 number = 208 // 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' + '0'

res = 'D0'

24 number = 13 // 16

number = 0

5 while 0 > 0: --- False

25 return 'D0'

4. square\_equal(59, 0, 0) = '0.0'

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

4 if 59 != 0: --- True

5 D = 0\*0 - 4\*59\*0

D = 0

6 if 0 > 0: --- False

10 elif 0 == 0: --- True

11 return str(-0 / (2\*59))

return '0.0'

5. square\_equal(0, 4, 81) = '-20.25'

3 def square\_equal(a=0, b=4, c=81)

4 if 0 != 0: --- False

14 else:

15 if 4 != 0: --- True

16 return str(-81 / 4)

return '-20.25'

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(535, 3) = 55

1 def remove\_digit(number=535, digit=3)

2 res = 0

3 power = 1

4 while 535 > 0: --- True

5 cur\_digit = 535 % 10

cur\_digit = 5

6 if 5 != 3: --- True

7 res = 0 + 5 \* 1

res = 5

8 power = 1 \* 10

power = 10

9 number = 535 // 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 = 5 + 5 \* 10

res = 55

8 power = 10 \* 10

power = 100

9 number = 5 // 10

number = 0

4 while 0 > 0: --- False

10 return 55

Вариант: 1-3-45

1. gcd(-24, 84) = 12

1 def gcd(x=-24, y=84)

2 if -24 < 0: --- True

3 x = --24

x = 24

4 if 84 < 0: --- False

6 while 84 != 0: --- True

7 rem = 24 % 84

rem = 24

8 x = 84

9 y = 24

6 while 24 != 0: --- True

7 rem = 84 % 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, 17) = 17

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

2 if 0 < 0: --- False

4 if 17 < 0: --- False

6 while 17 != 0: --- True

7 rem = 0 % 17

rem = 0

8 x = 17

9 y = 0

6 while 0 != 0: --- False

10 return 17

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(0, 50, -44) = '0.88'

3 def square\_equal(a=0, b=50, c=-44)

4 if 0 != 0: --- False

14 else:

15 if 50 != 0: --- True

16 return str(--44 / 50)

return '0.88'

5. square\_equal(9, 1, 49) = 'no roots'

3 def square\_equal(a=9, b=1, c=49)

4 if 9 != 0: --- True

5 D = 1\*1 - 4\*9\*49

D = -1763

6 if -1763 > 0: --- False

10 elif -1763 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(686) = '2\*7\*7\*7'

1 def factorize(n=686)

2 res = ''

3 while 686 > 2 and 686 % 2 == 0: --- True

4 res = '' + '2\*'

res = '2\*'

5 n = 686 // 2

n = 343

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 = '2\*' + str(7) + '\*'

res = '2\*7\*'

10 n = 343 // 7

n = 49

7 while 49 > 7: --- True

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

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

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

10 n = 49 // 7

n = 7

7 while 7 > 7: --- False

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

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

7. remove\_digit(838, 3) = 88

1 def remove\_digit(number=838, digit=3)

2 res = 0

3 power = 1

4 while 838 > 0: --- True

5 cur\_digit = 838 % 10

cur\_digit = 8

6 if 8 != 3: --- True

7 res = 0 + 8 \* 1

res = 8

8 power = 1 \* 10

power = 10

9 number = 838 // 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 = 8 + 8 \* 10

res = 88

8 power = 10 \* 10

power = 100

9 number = 8 // 10

number = 0

4 while 0 > 0: --- False

10 return 88

Вариант: 1-3-46

1. gcd(56, 72) = 8

1 def gcd(x=56, y=72)

2 if 56 < 0: --- False

4 if 72 < 0: --- False

6 while 72 != 0: --- True

7 rem = 56 % 72

rem = 56

8 x = 72

9 y = 56

6 while 56 != 0: --- True

7 rem = 72 % 56

rem = 16

8 x = 56

9 y = 16

6 while 16 != 0: --- True

7 rem = 56 % 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(28, 0) = 28

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

2 if 28 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 28

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(0, 5, -36) = '7.2'

3 def square\_equal(a=0, b=5, c=-36)

4 if 0 != 0: --- False

14 else:

15 if 5 != 0: --- True

16 return str(--36 / 5)

return '7.2'

5. square\_equal(16, -50, 55) = 'no roots'

3 def square\_equal(a=16, b=-50, c=55)

4 if 16 != 0: --- True

5 D = -50\*-50 - 4\*16\*55

D = -1020

6 if -1020 > 0: --- False

10 elif -1020 == 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(905, 0) = 95

1 def remove\_digit(number=905, digit=0)

2 res = 0

3 power = 1

4 while 905 > 0: --- True

5 cur\_digit = 905 % 10

cur\_digit = 5

6 if 5 != 0: --- True

7 res = 0 + 5 \* 1

res = 5

8 power = 1 \* 10

power = 10

9 number = 905 // 10

number = 90

4 while 90 > 0: --- True

5 cur\_digit = 90 % 10

cur\_digit = 0

6 if 0 != 0: --- False

9 number = 90 // 10

number = 9

4 while 9 > 0: --- True

5 cur\_digit = 9 % 10

cur\_digit = 9

6 if 9 != 0: --- True

7 res = 5 + 9 \* 10

res = 95

8 power = 10 \* 10

power = 100

9 number = 9 // 10

number = 0

4 while 0 > 0: --- False

10 return 95

Вариант: 1-3-47

1. gcd(-44, 16) = 4

1 def gcd(x=-44, y=16)

2 if -44 < 0: --- True

3 x = --44

x = 44

4 if 16 < 0: --- False

6 while 16 != 0: --- True

7 rem = 44 % 16

rem = 12

8 x = 16

9 y = 12

6 while 12 != 0: --- True

7 rem = 16 % 12

rem = 4

8 x = 12

9 y = 4

6 while 4 != 0: --- True

7 rem = 12 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(0, -90) = 90

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

2 if 0 < 0: --- False

4 if -90 < 0: --- True

5 y = --90

y = 90

6 while 90 != 0: --- True

7 rem = 0 % 90

rem = 0

8 x = 90

9 y = 0

6 while 0 != 0: --- False

10 return 90

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(0, 100, 92) = '-0.92'

3 def square\_equal(a=0, b=100, c=92)

4 if 0 != 0: --- False

14 else:

15 if 100 != 0: --- True

16 return str(-92 / 100)

return '-0.92'

5. square\_equal(18, 58, 71) = 'no roots'

3 def square\_equal(a=18, b=58, c=71)

4 if 18 != 0: --- True

5 D = 58\*58 - 4\*18\*71

D = -1748

6 if -1748 > 0: --- False

10 elif -1748 == 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(806, 0) = 86

1 def remove\_digit(number=806, digit=0)

2 res = 0

3 power = 1

4 while 806 > 0: --- True

5 cur\_digit = 806 % 10

cur\_digit = 6

6 if 6 != 0: --- True

7 res = 0 + 6 \* 1

res = 6

8 power = 1 \* 10

power = 10

9 number = 806 // 10

number = 80

4 while 80 > 0: --- True

5 cur\_digit = 80 % 10

cur\_digit = 0

6 if 0 != 0: --- False

9 number = 80 // 10

number = 8

4 while 8 > 0: --- True

5 cur\_digit = 8 % 10

cur\_digit = 8

6 if 8 != 0: --- True

7 res = 6 + 8 \* 10

res = 86

8 power = 10 \* 10

power = 100

9 number = 8 // 10

number = 0

4 while 0 > 0: --- False

10 return 86

Вариант: 1-3-48

1. gcd(55, -30) = 5

1 def gcd(x=55, y=-30)

2 if 55 < 0: --- False

4 if -30 < 0: --- True

5 y = --30

y = 30

6 while 30 != 0: --- True

7 rem = 55 % 30

rem = 25

8 x = 30

9 y = 25

6 while 25 != 0: --- True

7 rem = 30 % 25

rem = 5

8 x = 25

9 y = 5

6 while 5 != 0: --- True

7 rem = 25 % 5

rem = 0

8 x = 5

9 y = 0

6 while 0 != 0: --- False

10 return 5

2. gcd(-25, 0) = 25

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

2 if -25 < 0: --- True

3 x = --25

x = 25

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 25

3. hex(170) = 'AA'

1 def hex(number=170)

2 if 170 == 0: --- False

4 res = ''

5 while 170 > 0: --- True

6 digit = 170 % 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 = 170 // 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' + 'A'

res = 'AA'

24 number = 10 // 16

number = 0

5 while 0 > 0: --- False

25 return 'AA'

4. square\_equal(0, 20, 0) = '0.0'

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

4 if 0 != 0: --- False

14 else:

15 if 20 != 0: --- True

16 return str(-0 / 20)

return '0.0'

5. square\_equal(-45, -97, -72) = 'no roots'

3 def square\_equal(a=-45, b=-97, c=-72)

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

5 D = -97\*-97 - 4\*-45\*-72

D = -3551

6 if -3551 > 0: --- False

10 elif -3551 == 0: --- False

12 else:

13 return 'no roots'

6. factorize(4) = '2\*2'

1 def factorize(n=4)

2 res = ''

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

4 res = '' + '2\*'

res = '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\*' + str(2)

return '2\*2'

7. remove\_digit(928, 8) = 92

1 def remove\_digit(number=928, digit=8)

2 res = 0

3 power = 1

4 while 928 > 0: --- True

5 cur\_digit = 928 % 10

cur\_digit = 8

6 if 8 != 8: --- False

9 number = 928 // 10

number = 92

4 while 92 > 0: --- True

5 cur\_digit = 92 % 10

cur\_digit = 2

6 if 2 != 8: --- True

7 res = 0 + 2 \* 1

res = 2

8 power = 1 \* 10

power = 10

9 number = 92 // 10

number = 9

4 while 9 > 0: --- True

5 cur\_digit = 9 % 10

cur\_digit = 9

6 if 9 != 8: --- True

7 res = 2 + 9 \* 10

res = 92

8 power = 10 \* 10

power = 100

9 number = 9 // 10

number = 0

4 while 0 > 0: --- False

10 return 92

Вариант: 1-3-49

1. gcd(100, 52) = 4

1 def gcd(x=100, y=52)

2 if 100 < 0: --- False

4 if 52 < 0: --- False

6 while 52 != 0: --- True

7 rem = 100 % 52

rem = 48

8 x = 52

9 y = 48

6 while 48 != 0: --- True

7 rem = 52 % 48

rem = 4

8 x = 48

9 y = 4

6 while 4 != 0: --- True

7 rem = 48 % 4

rem = 0

8 x = 4

9 y = 0

6 while 0 != 0: --- False

10 return 4

2. gcd(76, 0) = 76

1 def gcd(x=76, y=0)

2 if 76 < 0: --- False

4 if 0 < 0: --- False

6 while 0 != 0: --- False

10 return 76

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(10, 75, -40) = '-8.0 and 0.5'

3 def square\_equal(a=10, b=75, c=-40)

4 if 10 != 0: --- True

5 D = 75\*75 - 4\*10\*-40

D = 7225

6 if 7225 > 0: --- True

7 x1 = (-75 - sqrt(7225)) / (2\*10)

x1 = -8.0

8 x2 = (-75 + sqrt(7225)) / (2\*10)

x2 = 0.5

9 return str(-8.0) + ' and ' + str(0.5)

return '-8.0 and 0.5'

5. square\_equal(-19, 17, -95) = 'no roots'

3 def square\_equal(a=-19, b=17, c=-95)

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

5 D = 17\*17 - 4\*-19\*-95

D = -6931

6 if -6931 > 0: --- False

10 elif -6931 == 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(797, 9) = 77

1 def remove\_digit(number=797, digit=9)

2 res = 0

3 power = 1

4 while 797 > 0: --- True

5 cur\_digit = 797 % 10

cur\_digit = 7

6 if 7 != 9: --- True

7 res = 0 + 7 \* 1

res = 7

8 power = 1 \* 10

power = 10

9 number = 797 // 10

number = 79

4 while 79 > 0: --- True

5 cur\_digit = 79 % 10

cur\_digit = 9

6 if 9 != 9: --- False

9 number = 79 // 10

number = 7

4 while 7 > 0: --- True

5 cur\_digit = 7 % 10

cur\_digit = 7

6 if 7 != 9: --- 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: 1-3-50

1. gcd(57, -63) = 3

1 def gcd(x=57, y=-63)

2 if 57 < 0: --- False

4 if -63 < 0: --- True

5 y = --63

y = 63

6 while 63 != 0: --- True

7 rem = 57 % 63

rem = 57

8 x = 63

9 y = 57

6 while 57 != 0: --- True

7 rem = 63 % 57

rem = 6

8 x = 57

9 y = 6

6 while 6 != 0: --- True

7 rem = 57 % 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, -16) = 16

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

2 if 0 < 0: --- False

4 if -16 < 0: --- True

5 y = --16

y = 16

6 while 16 != 0: --- True

7 rem = 0 % 16

rem = 0

8 x = 16

9 y = 0

6 while 0 != 0: --- False

10 return 16

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(0, 48, -96) = '2.0'

3 def square\_equal(a=0, b=48, c=-96)

4 if 0 != 0: --- False

14 else:

15 if 48 != 0: --- True

16 return str(--96 / 48)

return '2.0'

5. square\_equal(54, -34, 13) = 'no roots'

3 def square\_equal(a=54, b=-34, c=13)

4 if 54 != 0: --- True

5 D = -34\*-34 - 4\*54\*13

D = -1652

6 if -1652 > 0: --- False

10 elif -1652 == 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(300, 0) = 3

1 def remove\_digit(number=300, digit=0)

2 res = 0

3 power = 1

4 while 300 > 0: --- True

5 cur\_digit = 300 % 10

cur\_digit = 0

6 if 0 != 0: --- False

9 number = 300 // 10

number = 30

4 while 30 > 0: --- True

5 cur\_digit = 30 % 10

cur\_digit = 0

6 if 0 != 0: --- False

9 number = 30 // 10

number = 3

4 while 3 > 0: --- True

5 cur\_digit = 3 % 10

cur\_digit = 3

6 if 3 != 0: --- 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: 1-3-51

1. gcd(-75, -27) = 3

1 def gcd(x=-75, y=-27)

2 if -75 < 0: --- True

3 x = --75

x = 75

4 if -27 < 0: --- True

5 y = --27

y = 27

6 while 27 != 0: --- True

7 rem = 75 % 27

rem = 21

8 x = 27

9 y = 21

6 while 21 != 0: --- True

7 rem = 27 % 21

rem = 6

8 x = 21

9 y = 6

6 while 6 != 0: --- True

7 rem = 21 % 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, -99) = 99

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

2 if 0 < 0: --- False

4 if -99 < 0: --- True

5 y = --99

y = 99

6 while 99 != 0: --- True

7 rem = 0 % 99

rem = 0

8 x = 99

9 y = 0

6 while 0 != 0: --- False

10 return 99

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(0, 8, -14) = '1.75'

3 def square\_equal(a=0, b=8, c=-14)

4 if 0 != 0: --- False

14 else:

15 if 8 != 0: --- True

16 return str(--14 / 8)

return '1.75'

5. square\_equal(79, 42, 21) = 'no roots'

3 def square\_equal(a=79, b=42, c=21)

4 if 79 != 0: --- True

5 D = 42\*42 - 4\*79\*21

D = -4872

6 if -4872 > 0: --- False

10 elif -4872 == 0: --- False

12 else:

13 return 'no roots'

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

1 def factorize(n=6)

2 res = ''

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

4 res = '' + '2\*'

res = '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\*' + str(3)

return '2\*3'

7. remove\_digit(736, 6) = 73

1 def remove\_digit(number=736, digit=6)

2 res = 0

3 power = 1

4 while 736 > 0: --- True

5 cur\_digit = 736 % 10

cur\_digit = 6

6 if 6 != 6: --- False

9 number = 736 // 10

number = 73

4 while 73 > 0: --- True

5 cur\_digit = 73 % 10

cur\_digit = 3

6 if 3 != 6: --- True

7 res = 0 + 3 \* 1

res = 3

8 power = 1 \* 10

power = 10

9 number = 73 // 10

number = 7

4 while 7 > 0: --- True

5 cur\_digit = 7 % 10

cur\_digit = 7

6 if 7 != 6: --- True

7 res = 3 + 7 \* 10

res = 73

8 power = 10 \* 10

power = 100

9 number = 7 // 10

number = 0

4 while 0 > 0: --- False

10 return 73