Didžiausio bendro daliklio radimo kodas ir rezultatai:

```python
import math
import sympy as sp

def getGCD(nr1, nr2):
    divFinder = [[nr1, nr2, nr1 % nr2, nr1 // nr2]]
    n = len(divFinder)

    while divFinder[n-1][2] > 0:
        r = divFinder[n-1][1] % divFinder[n-1][2]
        d = divFinder[n-1][1] // divFinder[n-1][2]
        divFinder.append([divFinder[n-1][1], divFinder[n-1][2], r, d])
        n = len(divFinder)

    gcd = divFinder[n-1][1]
    return gcd, divFinder

gcd, divfinder = getGCD(57, 10)
print (gcd)
gcd, divfinder = getGCD(42, 56)
print (gcd)
gcd, divfinder = getGCD(152, 34)
print (gcd)
gcd, divfinder = getGCD(858, 246)
print (gcd)
```

```
[Running] python -u "c:\Users\Marius\Desktop\7 semestras\informacinės saugos pagrindai\uzduotys\6LD.py"
1
14
2
6
```

Realizuoju paskaitoje rodytą algoritmą ir pagal gautą lentelę(divFinder) randu didžiausią bendrą daliklį, grąžinu divFinder, nes naudosiu vėliau. Atsakymus patikrinau su internetine didžiausio bendro daliklio skaičiuokle.

Lentelė iš paskaitos apie kurią rašau (gaunama skaičiuojant 57 ir 10 didžiausią bendrą daliklį):

57	10	7
10	7	3
7	3	1
3	1	0

$p_n^{-1}(mod\ N) - ?$ kodas:

```
def calcReverseMod(number, mod):

    gcd, divFinder = getGCD(number, mod)
    reverseDivFinder = divFinder[::-1]
    del reverseDivFinder[0]
    if len(reverseDivFinder) == 1:
        print (number, "is not an invertable module", mod)
        print ("finding calcReverseMod(", number, ",", mod + 1, ") instead")

        gcd, divFinder = getGCD(number, mod + 1)
        reverseDivFinder = divFinder[::-1]
        del reverseDivFinder[0]

    n = len(reverseDivFinder)

    # for item in reverseDivFinder:
    #     print (item)

    formula = ""

    for item in reverseDivFinder:
        temp = ""
        if formula == "":
            temp = " " + str(item[0]) + " - " + str(item[1]) + " * " + str(item[3]) + " "
            formula = temp
        else:
            old = " " + str(item[2]) + " "
            temp = "( " + str(item[0]) + " - " + str(item[1]) + " * " + str(item[3]) + " )"
            formula = formula.replace(old, temp)

    formula = formula.replace(" " + str(reverseDivFinder[n-1][0]) + " ", " x ")
    formula = formula.replace(" " + str(reverseDivFinder[n-1][1]) + " ", " y ")
    x, y = sp.symbols('x,y')

    print ("formula:", reverseDivFinder[0][2], "=", formula)
    formula = "g = " + formula
    ldict = {'x': sp.symbols('x'), 'y': sp.symbols('y')}
    exec(formula, globals(), ldict)

    g = ldict['g']
    sp.pprint(g)
    g = str(g).replace(" ", "")

```

```

    temp = ""
    A = 0
    for char in str(g):
        if char != "*" and char != "x" and char != "y":
            temp += char
        elif char == "*" or char == "x":
            if temp == "":
                A = 1
            else:
                A = (int(temp))
            break

    print ("Didziausias bendras daliklis:", gcd)
    print ("Atsakymas:", A % mod)
    print ("")
```

Išvedimas ir rezultatai:

```
90   calcReverseMod(37, 190210129)
91   calcReverseMod(11, 23)
92   calcReverseMod(3, 1710930)

TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE

[Running] python -u "c:\Users\Marius\Desktop\7 semestras\informacinės saugos pagrindai\uzduotys\6LD.py"
formula: 1 = (( x - y * 0 )-( y -( x - y * 0 )* 5140814 )* 3 )-(( y -( x - y * 0 )* 5140814 )-(( x - y * 0 )-( y -( x - y * 0 )* 5140814 )* 3 )* 2 )* 1
51408143*x - 10*y
Didziausias bendras daliklis: 1
Atsakymas: 51408143

formula: 1 =  y -( x - y * 0 )* 2
-2*x + y
Didziausias bendras daliklis: 1
Atsakymas: 21

3 is not an invertable module 1710930
finding calcReverseMod( 3 , 1710931 ) instead
formula: 1 =  y -( x - y * 0 )* 570310
-570310*x + y
Didziausias bendras daliklis: 1
Atsakymas: 1140621
```

Pagal paskaitoje aiškintą algoritmą sudariau bendro didžiausio daliklio lygybę (pakeisdamas dalinamą skaičių į x, o daliklį į y, pvz. Jei turime sąlygą 11^{-1} (mod 23) tai 11 = x, o 23 = y), ją suprastinau skaičių šalia x paėmiau kaip A ir radau A mod y.

Tikrindamas atsakymus su Wolfram Alpha, radau kad kai y iš karto dalinasi iš x be liekanos toks modululis nėra „invertable“ todėl skaičiuodamas su 3 ir savo stud.nr (pagal skaidrėse duotą sąlygą) pridėjau 1 prie y ir tada skaičiavau.

Atsakymai:

37^{-1} (mod 190210129) = 51408143

11^{-1} (mod 23) = 21

3^{-1} (mod 1710931) = 1140621 (pagal užduoties skaidrę tik pridėjau vienetą prie savo stud.nr.)