Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Лабораторна робота №4

Дисципліна: Комп'ютерна дискретна математика

Виконав

Студент групи ПЗПІ-23-3

Харченко Федір Олександрович

Перевірив

Старший викладач каф. ПІ

Терещенко Гліб Юрійович

2023

Тема: "Number Theory operations and proves"

Мета: Understand and implement the Number Theory operations operations

Код програми:

https://github.com/rintaro129/CDM/tree/main/LAB4

```python
import math
import sympy
from sympy.calculus.util import continuous_domain, function_range
class Lab4:
    def parityChecker(n : int) -> str:
        if(n % 2 == 0):
            return "Even"
        else:
            return "Odd"

    def primeNumberChecker(n : int) -> str:
        res = True
        for i in range(2, math.trunc(math.sqrt(n))+1):
            if(n % i == 0):
                res = False
                break
        if(res):
            return "Prime"
        else:
            return "Not prime"

    def GCDCalculator(a : int, b : int) -> int:
        while b != 0:
            a = a % b
            a, b = b, a
        return a
```

```python
def primeFactorization(n : int) -> str:
    res = ""
    sqrtn = math.trunc(math.sqrt(n))
    i = 2

    while(n != 1 and i <= sqrtn):
        count = 0
        while(n % i == 0):
            n /= i
            count += 1
        if(count):
            if(res != ""):
                res += " * "
            res += str(i) + '^' + str(count)
        i += 1

    if(n != 1):
        if(res != ""):
            res += " * "
        res += str(n) + "^1"

    return res

def LCMCalculator(a : int, b : int) -> int:
    return a * b // Lab4.GCDCalculator(a, b)

```

```python
    def directProofImplementation(s : str) -> str:
        s2 = s
        s2 = s2.replace(',', '')
        s2 = s2.replace('.', '')
        a = s2.split()
        word = a[a.index("is") + 1]
        num = a[-1]
        res = '''Statement: {s}
        Hypothesis: a number is {word}
        Conclusion: it is divisible by {num}
        Direct Proof:
        '''.format(s = s, word=word, num=num)

        if(num == 1):
            return res + "A number is always divisible by 1"
        if(word == "even"):
            if(num == 2):
                return res + '''
Assuming the Hypothesis is True:
Assume that a number, let's call it n, is {word}.
According to the definition of {word} numbers, an {word} number can be expressed as n=2k, where k is an integer.
Assuming the hypothesis is true leads to the conclusion being true:
Since n=2k, it's clear that n is divisible by {num} because there exists an integer k such that n/{num}=k.
Therefore the statement {s} is true based on this proof.
                '''.format(word = word,num=num, s = s)
            else:
                return res + '''
Assuming the Hypothesis is True:
Assume that a number, let's call it n, is {word}.
According to the definition of {word} numbers, an {word} number can be expressed as n=2k, where k is an integer.
Assuming the hypothesis is true does not 100%% lead to the conclusion being true:
Since n=2k, it's unclear whether n is divisible by {num} or not.
Therefore the statement {s} cannot be proven by the direct proof.
                '''.format(word = word,num=num, s = s)
        elif(word == "odd"):
            if(num == 2):
                return res + '''
Assuming the Hypothesis is True:
Assume that a number, let's call it n, is {word}.
According to the definition of {word} numbers, an {word} number can be expressed as n=2k+1, where k is an integer.
Assuming the hypothesis is true leads to the conclusion being false:
Since n=2k+1, it's clear that n is not divisible by {num} because 2k is divisible by {num} by the definition and 1 is not.
Therefore the statement {s} is false based on this proof.
                '''.format(word = word,num=num, s = s)
            else:
                return res + '''
Assuming the Hypothesis is True:
Assume that a number, let's call it n, is {word}.
```

```python
Assume that a number, let's call it n, is {word}.
According to the definition of {word} numbers, an {word} number can be expressed as n=2k+1, where k is an integer.
Assuming the hypothesis is true does not 100%% lead to the conclusion being true:
Since n=2k+1, it's unclear whether n is divisible by {num} or not.
Therefore the statement {s} cannot be proven by the direct proof.
                '''.format(word = word,num=num, s = s)

    def proofByInduction(s) -> str:
        if(s == "The sum of the first n odd numbers is n^2 for all positive integers n."):
            return """
    Base case (n = 1):
For n = 1, we have the first odd number which is 1. 1^2 is equal to 1. Hence, the base case is correct.
    Induction hypothesis:
Let's assume that 1 + 3 + 5 +...+(2k-1)= k^2.
    Induction step:
We need to prove that the sum of the first (k+1) odd numbers is (k+1)^2, using the induction hypothesis.
The sum of the first (k+1) odd numbers can be expressed as:
1+ 3 + 5+...+(2k-1)+(2(k+1)-1)
We can rewrite the expression as:
(k^2)+(2(k+1)-1)
Expanding, we get:
k^2 + 2k + 1
Now, let's simplify the expression (k+1)^2:
(k+1)^2 = k^2 + 2k + 1
Since the expression for the sum of the first (k+1) odd numbers and (k+1)^2 are identical, this completes the induction step.
    Therefore, by the principle of mathematical induction, we can conclude that the sum of the first n odd numbers is n^2 for all positive integers n."""
```

```python
    def eulersToitient(n : int) -> int:
        amount = 0
        for k in range(1, n+1):
            if Lab4.GCDCalculator(n, k) == 1:
                amount += 1
        return amount

    def advancedFunctionEvaluation(func : str, value):
        x = sympy.Symbol('x')
        func = eval(func.replace('^', ' ** '))
        domain = continuous_domain(func, x, sympy.Reals)
        frange = function_range(func, x, domain)
        return "Function f(x) = {}\nDomain: {}\nRange: {}\nSolution: {}".format(func, domain, frange, func.subs(x, value))

print(Lab4.parityChecker(25))
print(Lab4.primeNumberChecker(17))
print(Lab4.GCDCalculator(48, 18))
print(Lab4.primeFactorization(56))
print(Lab4.LCMCalculator(15, 20))
print(Lab4.directProofImplementation("If a number is even, then it is divisible by 2."))
print(Lab4.eulersToitient(12))
print(Lab4.proofByInduction("The sum of the first n odd numbers is n^2 for all positive integers n."))
print(Lab4.advancedFunctionEvaluation("x^2 + 2*x + 1", 3))
```

Висновок: All tasks are implemented successfully.