```
"""5. a. Determine the greatest common divisors of elements from a list.
(recursive)
b. Insert an element on the n-position in a list. (recursive)"""


class Node:
    def __init__(self, value):
        self.value = value
        self.next = None


class List:
    def __init__(self):
        self.head = None


def createList():
    list = List()
    list.head = createList_rec()
    return list


def createList_rec():
    x = int(input("x = "))
    if x == 0:
        return None
    else:
        node = Node(x)
        node.next = createList_rec()
        return node


def gcd(a, b):
    if b == 0:
        return a
    else:
        return gcd(b, a % b)

def lcm(a, b):
    x = gcd(a,b)
    p = a * b
    return p/x

def lcmList_rec(node):
    if node is None:
        return 0
    elif node.next is None:
        return node.value
    else:
```

```
        return lcm(node.value, lcmList_rec(node.next))


def gcdList_rec(node):
    if node is None:
        return 0
    elif node.next is None:
        return node.value
    else:
        return gcd(node.value, gcdList_rec(node.next))


def insert_pos_rec(node, value, pos):
    if pos == 0:
        new_node = Node(value)
        new_node.next = node
        return new_node
    elif node is None:
        return None
    else:
        node.next = insert_pos_rec(node.next, value, pos - 1)
        return node

def insert_pos(list, value, pos):
    list.head = insert_pos_rec(list.head, value, pos)

def printList(list):
    node = list.head
    while node is not None:
        print(node.value)
        node = node.next


def menu():
    print("1. Create list")
    print("2. Determine GCD of list elements")
    print("3. Insert element at n-position")
    print ("4. Print list")
    print("5. LCM")
    print("6. Exit")
    list = None
    while True:
        choice = int(input("Choice: "))
        if choice == 1:
            list = createList()
        elif choice == 2:
            if list is None:
                print("List is empty")
            else:
```

```
                print("GCD:", gcdList_rec(list.head))
        elif choice == 3:
            if list is None:
                print("List is empty")
            else:
                value = int(input("Value to insert: "))
                pos = int(input("Position: "))
                insert_pos(list, value, pos)
                printList(list)
        elif choice == 4:
            printList(list)
        elif choice == 5:
            print("LCM:", int(lcmList_rec(list.head)))

        elif choice == 6:
            break
        else:
            print("Invalid choice")

if __name__ == "__main__":
    menu()
```

a) Let L=[l1,l2,...,ln] be a single linked list
gcd(a,b)= {a, if b = 0
{gcd(b, a mod b), if b != 0

gcdList_rec([]) = 0 (empty list)
gcdList_rec([l1]) = l1 (single element)
gcdList_rec([l1,l2,l3,..,ln]) = gcd(l1, gcdList_rec[l2,l3,...,ln])

b) Let L=[l1,l2,...,ln] be a single linked list, pos the position where we insert the element and value the value we insert
insert_pos_rec([l1,l2,l3,...,ln], value, pos) = [l1] U insert_pos_rec([l2,...,ln], value, pos - 1)
L([]) = None
L([l1,l2,...,ln],value,0) = L[value, l1, …, ln]

Examples:

E:\PLF-Lab1\.venv\Scripts\python.exe E:\PLF-Lab1\main.py
1. Create list
2. Determine GCD of list elements

3. Insert element at n-position
4. Print list
5. LCM
6. Exit
Choice: 1
x = 4
x = 6
x = 8
x = 2
x = 0
Choice: 2
GCD: 2
Choice: 3
Value to insert: 5
Position: 0
5
4
6
8
2
Choice:

E:\PLF-Lab1\.venv\Scripts\python.exe E:\PLF-Lab1\main.py
1. Create list
2. Determine GCD of list elements
3. Insert element at n-position
4. Print list
5. LCM
6. Exit
Choice: 1
x = 3
x = 4
x = 5
x = 6
x = 0
Choice: 2
GCD: 1
Choice: 3
Value to insert: 7
Position: 5
3
4
5

6
Choice: 3
Value to insert: 7
Position: 4
3
4
5
6
7