Lab 2

DSA BS DS Fall 2022

Task 1

Write the recursive function ***dec2oct(n)*** to return the octal number equivalent to its integer parameter. You have to write the main function.

Task 2

Write a recursive function to generate Pascal's Triangle up to a given number of rows. Pascal's Triangle is a triangular array of binomial coefficients, where each number is the sum of the two numbers directly above it.

Sr#	Input	Output	Explanation
1	numRows = 3	[[1],[1,1],[1,2,1]]	The first 3 rows of Pascal's Triangle are [[1],[1,1],[1,2,1]]. [[1], [1, 1], [1, 2, 1]]
2	numRows = 5	[[1],[1,1],[1,2,1],[1,3,3,1],[1,4,6,4,1]]	The first 5 rows of Pascal's Triangle are [[1],[1,1],[1,2,1],[1,3,3,1],[1,4,6,4,1]]. [[1], [1, 1], [1, 2, 1], [1, 3, 3, 1], [1, 4, 6, 4, 1]]
3	numRows = 1	[[1]]	The first row of Pascal's Triangle is [[1]].

Task 3

A digit string is good if the digits (0-indexed) at even indices are even and the digits at odd indices are prime.

Sr #	Input	Expected Output	Description
1	02468	Not Good	Number 6 at index 3 is not a prime number.
2	23478	Good	All digits at even indices are even, and all digits at odd indices are prime. Therefore, the digit string is considered "good".
3	224365	Good	All digits at even indices are even, and all digits at odd indices are prime. Therefore, the digit string is considered "good".

Write a recursive function which takes a digit string as input and identify the given digit string is a Good or Not.

Task 4

Today, you will implement the Link List class as we have discussed it in the previous lectures. Suppose, we have the following Node and LinkList class.

Implement the following methods given in LinkList Class.

```
class Node:
    def __init__(self, val=None):
        self.info = val
        self.next = None

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

    def insert_at_head(self, val):
    def insert_at_tail(self, val):
    def insert_after(self, key, val):
    def insert_before(self, key, val):
    def search(self, key):
    def display(self):
```

```python
# Main function
if __name__ == "__main__":
    obj = LinkList()
    obj.insert_at_head(2)
    obj.insert_at_head(3)
    obj.insert_at_tail(9)
    obj.insert_after(3,4)
    obj.insert_before(9,8)

    obj.display()
```