assignment-1

November 3, 2024

1 Question 1: Number Game Between User and Computer

[5]:
```
import random

def computer_move(current_max):
    return list(range(current_max + 1, current_max + random.randint(1, 3) + 1))

def play_number_game():
    current_max = 0
    while current_max < 20:
        try:
            user_moves = list(map(int, input(f"Player: Enter 1, 2, or 3␣
 ↪consecutive numbers starting from {current_max + 1}: ").split()))
            if user_moves[0] != current_max + 1 or len(user_moves) not in [1,␣
 ↪2, 3] or any(user_moves[i] != user_moves[0] + i for i in␣
 ↪range(len(user_moves))):
                raise ValueError
            current_max = user_moves[-1]
            if current_max >= 20:
                print("Player Wins!!!")
                return
        except ValueError:
            print("Invalid move! Enter consecutive numbers in sequence.")
            continue

        computer_moves = computer_move(current_max)
        print(f"Computer played: {computer_moves}")
        current_max = computer_moves[-1]
        if current_max >= 20:
            print("Computer Wins!!!")
            return

play_number_game()
```

```
Player: Enter 1, 2, or 3 consecutive numbers starting from 1: 1 2 3
Computer played: [4, 5, 6]
Player: Enter 1, 2, or 3 consecutive numbers starting from 7: 7 8
```

```
Computer played: [9, 10]
Player: Enter 1, 2, or 3 consecutive numbers starting from 11: 11 12
Computer played: [13, 14]
Player: Enter 1, 2, or 3 consecutive numbers starting from 15: 15
Computer played: [16]
Player: Enter 1, 2, or 3 consecutive numbers starting from 17: 17
Computer played: [18]
Player: Enter 1, 2, or 3 consecutive numbers starting from 19: 19 20
Player Wins!!!
```

2 Question 2: Pascal's Triangle Using nCr Function

[8]:
```python
def ncr(n, r):
    result = 1
    for i in range(r):
        result = result * (n - i) // (i + 1)
    return result
def print_pascal_triangle(rows):
    for i in range(rows):
        row = 1
        print(" " * (rows - i), end="")
        for j in range(i + 1):
            print(row, end=" ")
            row = row * (i - j) // (j + 1)
        print()
rows = int(input("Enter number of rows for Pascal's triangle: "))
print_pascal_triangle(rows)
```

```
Enter number of rows for Pascal's triangle: 8
        1
       1 1
      1 2 1
     1 3 3 1
    1 4 6 4 1
   1 5 10 10 5 1
  1 6 15 20 15 6 1
 1 7 21 35 35 21 7 1
```

3 Question 3: Print Repeated Elements with Frequency Count

[10]:
```python
from collections import Counter

def print_frequency_count(nums):
    counter = Counter(nums)
    for element, frequency in counter.items():
        print(f"Element {element} has come {frequency} times")
```

```
nums = list(map(int, input("Enter numbers separated by spaces: ").split()))
print_frequency_count(nums)
```

```
Enter numbers separated by spaces: 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5 6 6 6 6 6 6 7 7
7 7 7 7 7 8 8 8 8 8 8 8 8
Element 1 has come 1 times
Element 2 has come 2 times
Element 3 has come 3 times
Element 4 has come 4 times
Element 5 has come 5 times
Element 6 has come 6 times
Element 7 has come 7 times
Element 8 has come 8 times
```

4 Question 4: Addition of 2x2 Matrices from File

[18]:
```
def read_matrix(filename):
    with open(filename, 'r') as file:
        lines = file.readlines()
        matrix_lines = [line.strip() for line in lines if line.strip() and not␣
 ↪line.startswith("Matrix")]
        matrix_a = [list(map(int, matrix_lines[0].split())), list(map(int,␣
 ↪matrix_lines[1].split()))]
        matrix_b = [list(map(int, matrix_lines[2].split())), list(map(int,␣
 ↪matrix_lines[3].split()))]
        print("Matrix A:", matrix_a)
        print("Matrix B:", matrix_b)

    return matrix_a, matrix_b


def add_matrices(matrix_a, matrix_b):
    result = [[matrix_a[i][j] + matrix_b[i][j] for j in range(2)] for i in␣
 ↪range(2)]
    return result
matrix_a, matrix_b = read_matrix('matrices.txt')
result_matrix = add_matrices(matrix_a, matrix_b)
print("Result of A + B:")
for row in result_matrix:
    print(row)
```

```
Matrix A: [[1, 2], [3, 4]]
Matrix B: [[5, 6], [7, 8]]
Result of A + B:
[6, 8]
```

```
[10, 12]
```

5 Question 5: Overloading + Operator for Fraction Class

[20]:
```python
from math import gcd

class Fraction:
    def __init__(self, numerator, denominator):
        self.numerator = numerator
        self.denominator = denominator
        self.simplify()

    def simplify(self):
        common_divisor = gcd(self.numerator, self.denominator)
        self.numerator //= common_divisor
        self.denominator //= common_divisor

    def __add__(self, other):
        if isinstance(other, Fraction):
            new_numerator = (self.numerator * other.denominator) + (other.
 ↪numerator * self.denominator)
            new_denominator = self.denominator * other.denominator
            return Fraction(new_numerator, new_denominator)
        return NotImplemented

    def __str__(self):
        return f"{self.numerator}/{self.denominator}"
frac1 = Fraction(4, 2)
frac2 = Fraction(5, 3)
result = frac1 + frac2
print("Result of addition:", result)
```

```
Result of addition: 11/3
```

[ ]:
```python

```