In [1]:

```
import random
def computer_turn(current_number):
    max_pick = min(3, 20 - current_number)
    computer_choice = random.randint(1, max_pick)
    next_numbers = list(range(current_number + 1, current_number + 1 + computer_choice))
    print(f"Computer plays: {next_numbers}")
    return next_numbers[-1]
def user_turn(current_number):
    while True:
        try:
            user_input = input("Enter 1, 2, or 3 numbers in sequence: ").strip()
            user_choice = list(map(int, user_input.split()))
            if (1 <= len(user_choice) <= 3 and
                    all(x == current_number + i + 1 for i, x in enumerate(user_choice))
                    user_choice[-1] <= 20):
                print(f"You play: {user_choice}")
                return user_choice[-1]
            else:
                print("Invalid input. Please enter 1 to 3 sequential numbers starting fr
        except ValueError:
            print("Invalid input. Please enter numbers only.")
def number_game():
    current_number = 0
    while current_number < 20:
        current_number = user_turn(current_number)
        if current_number >= 20:
            print("Congratulations! You reached 20 and won the game.")
            break
        current_number = computer_turn(current_number)
        if current_number >= 20:
            print("Computer reached 20 and won the game. Better luck next time!")
            break
number_game()
```

```
Enter 1, 2, or 3 numbers in sequence: 1
You play: [1]
Computer plays: [2, 3]
Enter 1, 2, or 3 numbers in sequence: 2
Invalid input. Please enter 1 to 3 sequential numbers starting from the last number.
Enter 1, 2, or 3 numbers in sequence: 3
Invalid input. Please enter 1 to 3 sequential numbers starting from the last number.
Enter 1, 2, or 3 numbers in sequence: 4
You play: [4]
Computer plays: [5]
Enter 1, 2, or 3 numbers in sequence: 5
Invalid input. Please enter 1 to 3 sequential numbers starting from the last number.
Enter 1, 2, or 3 numbers in sequence: 6
You play: [6]
Computer plays: [7]
Enter 1, 2, or 3 numbers in sequence: 7
Invalid input. Please enter 1 to 3 sequential numbers starting from the last number.
Enter 1, 2, or 3 numbers in sequence: 8
You play: [8]
Computer plays: [9]
Enter 1, 2, or 3 numbers in sequence: 10
You play: [10]
Computer plays: [11, 12]
Enter 1, 2, or 3 numbers in sequence: 13
You play: [13]
Computer plays: [14, 15]
Enter 1, 2, or 3 numbers in sequence: 16
You play: [16]
Computer plays: [17, 18, 19]
Enter 1, 2, or 3 numbers in sequence: 20
```

```
You play: [20]
Congratulations! You reached 20 and won the game.
```

In [ ]:

```
```

In [3]:

```
def factorial(num):
    if num == 0 or num == 1:
        return 1
    return num * factorial(num - 1)
def ncr(n, r):
    return factorial(n) // (factorial(r) * factorial(n - r))
def print_pascal_triangle(rows):
    for i in range(rows):
        print(" " * (rows - i), end="")
        for j in range(i + 1):
            print(ncr(i, j), end=" ")
        print()
rows = int(input("Enter the number of rows for Pascal's Triangle: "))
print_pascal_triangle(rows)
```

```
Enter the number of rows for Pascal's Triangle: 6
      1 
     1 1 
    1 2 1 
   1 3 3 1 
  1 4 6 4 1 
 1 5 10 10 5 1 
```

In [ ]:

In [ ]:

In [2]:

```python
from collections import Counter
n = int(input("Enter the number of elements in the list: "))
numbers = []
for _ in range(n):
    number = int(input("Enter a number: "))
    numbers.append(number)
frequency_count = Counter(numbers)
print("Repeated elements with their frequency count:")
for element, count in frequency_count.items():
    if count > 1:
        print(f"{element}: {count}")
```

```
Enter the number of elements in the list: 5
Enter a number: 1
Enter a number: 2
Enter a number: 3
Enter a number: 3
Enter a number: 2
Repeated elements with their frequency count:
2: 2
3: 2
```

In [9]:

```
def read_matrix(filename):
    with open(filename, 'r') as file:
        lines = file.readlines()
        matrix_a = [list(map(int, lines[i].split())) for i in range(2)]
        matrix_b = [list(map(int, lines[i].split())) for i in range(2, 4)]
    return matrix_a, matrix_b
def add_matrices(matrix_a, matrix_b):
    result = [[0, 0], [0, 0]]
    for i in range(2):
        for j in range(2):
            result[i][j] = matrix_a[i][j] + matrix_b[i][j]
    return result
def print_matrix(matrix, name):
    print(f"Matrix {name}:")
    for row in matrix:
        print(" ".join(map(str, row)))
    print()
```

In [10]:

```
filename = "matrices.txt"
matrix_a, matrix_b = read_matrix(filename)
result_matrix = add_matrices(matrix_a, matrix_b)
```

In [11]:

```
print_matrix(matrix_a, "A")
print_matrix(matrix_b, "B")
print_matrix(result_matrix, "A + B")
```

```
Matrix A:
10 2
5 4

Matrix B:
8 6
9 8

Matrix A + B:
18 8
14 12
```

In [ ]:

In [ ]:

In [1]:

```python
from math import gcd
class Fraction:
    def _init_(self, numerator, denominator):
        self.numerator = numerator
        self.denominator = denominator
    def _add_(self, other):
        if isinstance(other, Fraction):
            new_nemerator = (self.numerator * other.denominator + other.numerator * self
            new_denominator = self.denominator * other.denominator
            common_divisor = gcd(new_numerator, new_denominator)
            new_numerator //= commom_divisor
            new_denominator //= common_divisor
            return Fraction(new_numerator, new_denominator)
        else:
            return NotImplemented
    def _str_(self):
        return f"{self.numerator}/{self.denominator}"
```

In [2]:

```python
from fractions import Fraction
fraction1 = Fraction(1, 2)
fraction2 = Fraction(1, 3)
result = fraction1 + fraction2
print(f"The result of adding {fraction1} and {fraction2} is: {result}")
```

```
The result of adding 1/2 and 1/3 is: 5/6
```

In [ ]: