DATE:28.12.23 DAY3- JAVA- QUIZ-1

1. Create Account A and Account B with an initial balance of 5000 and 2500 respectively. Trans
amount of 1500 from Account A to B and an amount of 3000 from Account B to A. Print the re
with the following details after each transaction

```
import java.util.Scanner;

class BankAccount {
   private int accountId;
   private String name;
   private double balance;

   public BankAccount(int accountId, String name, double balance) {
      this.accountId = accountId;
      this.name = name;
      this.balance = balance;
   }

   public void transferTo(BankAccount recipient, double amount) {
      if (balance >= amount) {
         balance -= amount;
         recipient.balance += amount;
         printReceipt();
      } else {
         System.out.println("Insufficient funds for the transfer.");
      }
   }

   public void printReceipt() {
      System.out.println("Account id: " + accountId);
      System.out.println("Name: " + name);
      System.out.println("Account Balance: Rs." + balance);
      System.out.println();
   }
}

public class BankTransaction {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);

      BankAccount accountA = new BankAccount(12344, "Account A", 5000);

      BankAccount accountB = new BankAccount(56789, "Account B", 2500);


      System.out.print("Enter the amount to transfer from Account A to B: Rs.");
      double transferAmountAB = scanner.nextDouble();
      accountA.transferTo(accountB, transferAmountAB);

      System.out.print("Enter the amount to transfer from Account B to A: Rs.");
      double transferAmountBA = scanner.nextDouble();
      accountB.transferTo(accountA, transferAmountBA);

      scanner.close();
```

```
    }

}
```

2. Given an array and a partition size, you have to partition the array with that value , then we will s
partition order, you have to merge based on that order

Input:
Array : 1 2 3 4 5
Partition size 2 (so the array will be partitioned as 1 2**, 3 4,** 5**)**
Partition order 3 2 1

Output:
5 3 4 1 2

```
import java.util.Scanner;

public class ArrayPartitionAndMerge {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter elements of the array (space-separated): ");

        String[] inputArray = scanner.nextLine().split(" ");

        System.out.print("Enter the partition size: ");

        int partitionSize = scanner.nextInt();

        System.out.print("Enter the partition order (space-separated): ");

        int[] partitionOrder = new int[inputArray.length / partitionSize];

        for (int i = 0; i < partitionOrder.length; i++) {

            partitionOrder[i] = scanner.nextInt();

        }

        String[] partitionedArray = new String[inputArray.length];

        int k = 0;

        for (int i = 0; i < inputArray.length; i += partitionSize) {

            for (int j = 0; j < partitionSize; j++) {

                partitionedArray[k++] = inputArray[i + j];

            }
```

```
    }     String[] mergedArray = new String[inputArray.length];

    int index = 0;

    for (int i : partitionOrder) {

        int start = (i - 1) * partitionSize;

        for (int j = start; j < start + partitionSize; j++) {

            mergedArray[index++] = partitionedArray[j];

        }       }

    System.out.print("Merged Array: ");

    for (String element : mergedArray) {

        System.out.print(element + " ");

    }

    scanner.close();

  }
}
```

3. A palindrome number - number that remains the same after reversing each digit of that number. A prime number - number that is divisible by only one or itself. A number that satisfies both the properties is said to be PalPrime Number.

CODE:

```
import java.util.Scanner;

class PalPrime {
   public PalPrime(int number, String message) {
      System.out.println("Number " + number + " is " + message);
   }

   public static boolean isPalindrome(int num) {
      String strNum = String.valueOf(num);
      String reversedStr = new StringBuilder(strNum).reverse().toString();
      return strNum.equals(reversedStr);
   }

   public static boolean isPrime(int num) {
      if (num <= 1) {
         return false;
      }
      for (int i = 2; i <= Math.sqrt(num); i++) {
         if (num % i == 0) {
            return false;
         }
      }
```

```
        return true;
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of elements in the array: ");
        int n = scanner.nextInt();

        int[] numbers = new int[n];
        System.out.println("Enter the elements of the array:");

        for (int i = 0; i < n; i++) {
            numbers[i] = scanner.nextInt();
        }

        for (int num : numbers) {
            boolean isPalindrome = PalPrime.isPalindrome(num);
            boolean isPrime = PalPrime.isPrime(num);

            if (isPalindrome && isPrime) {
                new PalPrime(num, "PalPrime");
            } else if (isPalindrome) {
                new PalPrime(num, "Palindrome");
            } else if (isPrime) {
                new PalPrime(num, "Prime");
            }
        }

        scanner.close();
    }
}
```