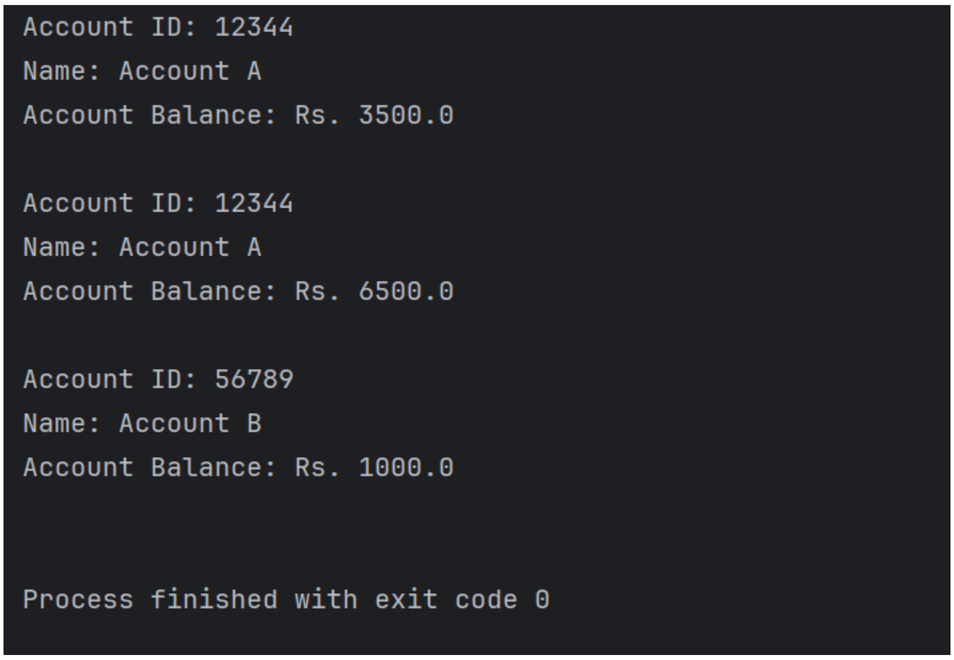
1. Create Account A and Account B with an initial balance of 5000 and 2500 respectively. Transfer amount of 1500 from Account A to B and an amount of 3000 from Account B to A.

Print the receipt with the following details after each transaction

Output:
Account id: 12344,
Name: XXXX,
Account Balance: Rs.____

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

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

   public void deposit(double amount) {
      balance += amount;
   }

   public boolean withdraw(double amount) {
      if (balance >= amount) {
         balance -= amount;
         return true;
      } else {
         System.out.println("Insufficient funds.");
         return false;
      }
   }

   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 Main {
   public static void main(String[] args) {

      BankAccount accountA = new BankAccount(12344, "Account A", 5000);
      BankAccount accountB = new BankAccount(56789, "Account B", 2500);

      accountA.withdraw(1500);
      accountB.deposit(1500);

      accountA.printReceipt();
```

```
        accountB.withdraw(3000);
        accountA.deposit(3000);

        accountA.printReceipt();

        accountB.printReceipt();

    }
}
```



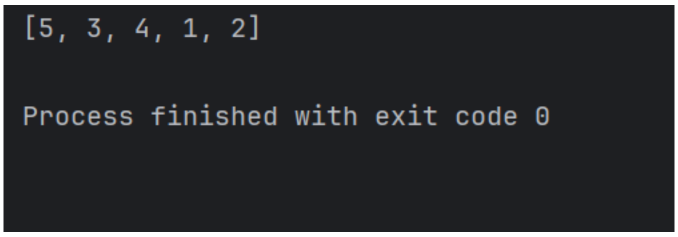
```
Account ID: 12344
Name: Account A
Account Balance: Rs. 3500.0

Account ID: 12344
Name: Account A
Account Balance: Rs. 6500.0

Account ID: 56789
Name: Account B
Account Balance: Rs. 1000.0


Process finished with exit code 0
```

2. Given an array and a partition size, you have to partition the array with that value , then we will specify the 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.ArrayList;
import java.util.Arrays;
import java.util.List;

public class Main{
    public static void main(String[] args) {
        int[] array = {1, 2, 3, 4, 5};
        int partitionSize = 2;
        int[] partitionOrder = {3, 2, 1};


        List<Integer[]> partitions = partitionArray(array, partitionSize);


        int[] mergedArray = mergePartitions(partitions, partitionOrder);
        System.out.println(Arrays.toString(mergedArray));
    }

    private static List<Integer[]> partitionArray(int[] array, int partitionSize) {
        List<Integer[]> partitions = new ArrayList<>();
        for (int i = 0; i < array.length; i += partitionSize) {
            int endIndex = Math.min(i + partitionSize, array.length);
            Integer[] partition = new Integer[endIndex - i];
            for (int j = i; j < endIndex; j++) {
                partition[j - i] = array[j];
            }
            partitions.add(partition);
        }
        return partitions;
    }

    private static int[] mergePartitions(List<Integer[]> partitions, int[] partitionOrder) {
        List<Integer> result = new ArrayList<>();
        for (int order : partitionOrder) {
            if (order <= partitions.size()) {
                result.addAll(Arrays.asList(partitions.get(order - 1)));
            }
        }


        return result.stream().mapToInt(Integer::intValue).toArray();
    }
}
```



```
[5, 3, 4, 1, 2]

Process finished with exit code 0
```

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.

Create a class PalPrime with a parameterised constructor PalPrime(int number, String message).

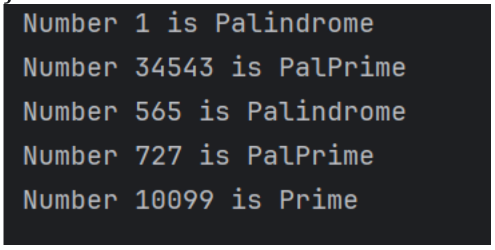
Given an positive integer array of numbers, you have to traverse the array and print the message "Number ___ is Prime/Palindrome/PalPrime".

Note: Message should be printed via constructor of PalPrime class.

Input :
Array: [1, 34543, 565, 727, 10099]

Output -> Predict the output

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

    public static boolean isPalindrome(int number) {
        String numStr = Integer.toString(number);
        String reversedNumStr = new StringBuilder(numStr).reverse().toString();
        return numStr.equals(reversedNumStr);
    }

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

    public static void main(String[] args) {
        int[] numbers = {1, 34543, 565, 727, 10099};

        for (int number : numbers) {
            boolean isPalindrome = isPalindrome(number);
            boolean isPrime = isPrime(number);

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



```
Number 1 is Palindrome
Number 34543 is PalPrime
Number 565 is Palindrome
Number 727 is PalPrime
Number 10099 is Prime
```