

1. Write a program to convert numbers into words using Enumerations with constructors, methods and instance variables.(INPUT RANGE-0 TO 99999)
EX: 36 THIRTY SIX

```
import java.util.Scanner;
public class NumberToWords{
    enum Units {
        ZERO(""), ONE("one"), TWO("two"), THREE("three"), FOUR("four"), FIVE("five"),
        SIX("six"), SEVEN("seven"), EIGHT("eight"), NINE("nine"), TEN("ten"),
        ELEVEN("eleven"), TWELVE("twelve"), THIRTEEN("thirteen"),
        FOURTEEN("fourteen"),
        FIFTEEN("fifteen"), SIXTEEN("sixteen"), SEVENTEEN("seventeen"),
        EIGHTEEN("eighteen"),
        NINETEEN("nineteen");

        private final String word;

        Units(String word) {
            this.word = word;
        }

        public String getWord() {
            return word;
        }
    }

    enum Tens {
        ZERO(""), TEN(""), TWENTY("twenty"), THIRTY("thirty"), FORTY("forty"),
        FIFTY("fifty"),
        SIXTY("sixty"), SEVENTY("seventy"), EIGHTY("eighty"), NINETY("ninety");

        private final String word;

        Tens(String word) {
            this.word = word;
        }
        public String getWord() {
            return word;
        }
    }

    public static String converttowords(int number) {
        if (number == 0) {
            return Units.ZERO.getWord();
        }
        StringBuilder result = new StringBuilder();
        int thousands = number / 1000;
        int remaining = number % 1000;
        if (thousands > 0) {
```

```

        result.append(convert(thousands)).append(" thousand ");
    }

    result.append(convert(remaining));

    return result.toString().trim();
}

private static String convert(int number) {
    StringBuilder result = new StringBuilder();

    int hundreds = number / 100;
    int remaining = number % 100;

    if (hundreds > 0) {
        result.append(Units.values()[hundreds].getWord()).append(" hundred ");
    }

    if (remaining != 0) {
        if (remaining < 20) {
            result.append(Units.values()[remaining].getWord());
        } else {
            int tens = remaining / 10;
            int units = remaining % 10;
            result.append(Tens.values()[tens].getWord());
            if (units > 0) {
                result.append(" ").append(Units.values()[units].getWord());
            }
        }
    }

    return result.toString().trim();
}

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter a number (0 to 99999): ");
    int number = scanner.nextInt();
    if (number < 0 || number > 99999) {
        System.out.println("Number out of range.");
    } else {
        System.out.println(number + " in words: " + converttowords(number));
    }

    scanner.close();
}
}

```

2. Find the second maximum and second minimum in a set of numbers using auto boxing and unboxing.

```
import java.util.*;
```

```
public class SecondMinMax {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of elements: ");
        int n = scanner.nextInt();

        List numbers = new ArrayList();
        for (int i = 0; i < n; i++) {
            System.out.print("Enter element " + (i + 1) + ": ");
            int input = scanner.nextInt();
            numbers.add(input); // Autoboxing: primitive int to Integer object
        }

        Integer secondMin = findSecondMin(numbers);
        Integer secondMax = findSecondMax(numbers);

        if (secondMin == null || secondMax == null) {
            System.out.println("There are less than 2 elements.");
        } else {
            System.out.println("Second minimum: " + secondMin);
            System.out.println("Second maximum: " + secondMax);
        }

        scanner.close();
    }

    public static Integer findSecondMin(List numbers) {
        if (numbers.size() < 2)
            return null;

        int min = Integer.MAX_VALUE;
        int secondMin = Integer.MAX_VALUE;

        for (Object obj : numbers) {
            int num = (int) obj; // Unboxing: Object to primitive int
            if (num < min) {
                secondMin = min;
                min = num;
            } else if (num < secondMin && num != min) {
                secondMin = num;
            }
        }

        return secondMin == Integer.MAX_VALUE ? null : secondMin;
    }
}
```

```

    }

    public static Integer findSecondMax(List numbers) {
        if (numbers.size() < 2)
            return null;

        int max = Integer.MIN_VALUE;
        int secondMax = Integer.MIN_VALUE;

        for (Object obj : numbers) {
            int num = (int) obj; // Unboxing: Object to primitive int
            if (num > max) {
                secondMax = max;
                max = num;
            } else if (num > secondMax && num != max) {
                secondMax = num;
            }
        }

        return secondMax == Integer.MIN_VALUE ? null : secondMax;
    }
}

```

4. Write a java program to find words with even number of characters in a string, then swap the pair of characters in those words and also toggle the characters in a given string

EX:

Good Morning everyone

Output: oGdo vereoyen

gOOD mORNING EVERYONE

```
import java.util.*;
public class StringManipulation {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter a string: ");
        String input = scanner.nextLine();

        String swappedString = swapEvenCharacters(input);
        System.out.println("Swapped string: " + swappedString);

        String toggledString = toggleCharacters(input);
        System.out.println("Toggled string: " + toggledString);
    }

    public static String swapEvenCharacters(String input) {
        StringBuilder result = new StringBuilder();
        String[] words = input.split("\\s+");
        for (String word : words) {
            if (word.length() % 2 == 0) {
                char[] chars = word.toCharArray();
                for (int i = 0; i < chars.length - 1; i += 2) {
                    char temp = chars[i];
                    chars[i] = chars[i + 1];
                    chars[i + 1] = temp;
                }
                result.append(chars).append(" ");
            } else {
                result.append(word).append(" ");
            }
        }
        return result.toString().trim();
    }

    public static String toggleCharacters(String input) {
        StringBuilder result = new StringBuilder();
        for (char c : input.toCharArray()) {
            if (Character.isUpperCase(c)) {
                result.append(Character.toLowerCase(c));
            } else if (Character.isLowerCase(c)) {
                result.append(Character.toUpperCase(c));
            } else {
                result.append(c);
            }
        }
        return result.toString();
    }
}
```

```

        result.append(c);
    }
}
return result.toString();
}
}

```

3. Write a menu driven program to create an ArrayList and perform the following operations

- i) Adding elements
- ii) Sorting elements
- iii) Replace an element with another
- iv) Removing an element
- v) Displaying all the elements
- vi) Adding an element between two elements

```

import java.util.*;

public class ArrayListOperations {
    private static Scanner scanner = new Scanner(System.in);
    private static List<Integer> arrayList = new ArrayList<>();

    public static void main(String[] args) {
        boolean exit = false;
        while (!exit) {
            System.out.println("\nArrayList Operations Menu:");
            System.out.println("1. Add elements");
            System.out.println("2. Sort elements");
            System.out.println("3. Replace an element with another");
            System.out.println("4. Remove an element");
            System.out.println("5. Display all elements");
            System.out.println("6. Add an element between two elements");
            System.out.println("7. Exit");
            System.out.print("Enter your choice: ");

            int choice = scanner.nextInt();
            scanner.nextLine(); // Consume newline

            switch (choice) {
                case 1:
                    addElements();
                    break;
                case 2:
                    sortElements();
                    break;
                case 3:
                    replaceElement();
                    break;
            }
        }
    }
}

```

```

        case 4:
            removeElement();
            break;
        case 5:
            displayElements();
            break;
        case 6:
            addElementBetween();
            break;
        case 7:
            exit = true;
            break;
        default:
            System.out.println("Invalid choice. Please enter a number between 1 and 7.");
    }
}

scanner.close();
}

private static void addElements() {
    System.out.print("Enter the number of elements to add: ");
    int n = scanner.nextInt();
    scanner.nextLine(); // Consume newline

    System.out.println("Enter the elements:");
    for (int i = 0; i < n; i++) {
        int element = scanner.nextInt();
        arrayList.add(element);
    }
    System.out.println("Elements added successfully.");
}

private static void sortElements() {
    Collections.sort(arrayList);
    System.out.println("Elements sorted successfully.");
}

private static void replaceElement() {
    System.out.print("Enter the index of the element to replace: ");
    int index = scanner.nextInt();
    System.out.print("Enter the new element: ");
    int newElement = scanner.nextInt();
    arrayList.set(index, newElement);
    System.out.println("Element replaced successfully.");
}

private static void removeElement() {

```

```
        System.out.print("Enter the index of the element to remove: ");
        int index = scanner.nextInt();
        arrayList.remove(index);
        System.out.println("Element removed successfully.");
    }

    private static void displayElements() {
        System.out.println("Elements in the ArrayList:");
        for (Integer element : arrayList) {
            System.out.print(element + " ");
        }
        System.out.println();
    }

    private static void addElementBetween() {
        System.out.print("Enter the index after which you want to add the element: ");
        int index = scanner.nextInt();
        System.out.print("Enter the element to add: ");
        int newElement = scanner.nextInt();
        arrayList.add(index + 1, newElement);
        System.out.println("Element added successfully.");
    }
}
```