1. Write a program to take integer inputs from a user into an arrayList and perform following tasks:

- add n elements to the array list
- print all the multiples of x.
- search for an element

```
import java.util.ArrayList;
import java.util.Iterator;
import java.util.Scanner;

public class ArraylistSample {
    ArrayList<Integer> al;

    ArraylistSample(){
        al = new ArrayList<Integer>();
    }

    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        ArraylistSample ob =new ArraylistSample();
        ob.addElement();
        System.out.println("Enter number");
        int x=sc.nextInt();
        ob.printMultiples(x);
        System.out.println("\nEnter search element");
        ob.searchElement(sc.nextInt());

    }

    private void searchElement(int n) {

        if(al.indexOf(n)==-1)
            System.out.println("Element is not found.");
```

```
        else
            System.out.println("Element found at index "
+ al.indexOf(n));



    }

    private void printMultiples(int n) {
        Iterator<Integer> it= al.iterator();
        System.out.print("Multiples of "+n+": ");
        while (it.hasNext()){
            int a=it.next();
            if(a%n==0){
                System.out.print(a + " ");
            }
        }
    }

    private void addElement() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of
elements to be added :");
        int n=sc.nextInt();
        System.out.println("Enter the elements");
        for(int i=1;i<=n;i++){
            al.add(sc.nextInt());
        }
        System.out.println("Elements added");

    }
}
```

2. Write a program to reverse a string, using stack.

```
import java.util.LinkedList;
import java.util.Scanner;

public class Reverse {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        LinkedList<Character> stk= new
LinkedList<Character>();
        String rev="";
        System.out.println("Enter a string");
        char[] chArray=sc.nextLine().toCharArray();
        for(char ch : chArray){
            stk.push(ch);
        }
        while(!stk.isEmpty()){
            rev+=stk.pop();
        }
        System.out.println("Reverse : "+rev);
    }
}
```

3. Input a word and print the following pattern
eg: “Apple”

Apple
ppleA
pleAp
leApp
eAppl

```java
import java.util.Scanner;

public class Pattern {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a word :");
        String wd = sc.next();
        int n=wd.length();
        for(int i=0;i<n;i++){
            for(int j=i;j<n;j++) //for 1st part
                System.out.print(wd.charAt(j)+" ");
            for(int j=0;j<i;j++) //for 2nd part
                System.out.print(wd.charAt(j)+" ");
            System.out.println();
        }
    }
}
```

4.Write a program to input a set of names from the user and print the largest and the smallest words (lexicographically)

```java
import java.util.Scanner;
import java.util.TreeSet;

public class lexico {
    public static void main(String[] args) {
        TreeSet<String> names = new
TreeSet<String>();
        Scanner sc =new Scanner(System.in);
        System.out.println("Enter number of names");
```

```
        int n = sc.nextInt();
        sc.nextLine();
        System.out.println("Enter the names");

        for (int i = 0; i < n; i++) {
            names.add(sc.nextLine());
        }
        System.out.println(names);
        System.out.println("Smallest name :
"+names.first());
        System.out.println("Largest name :
"+names.last());

    }
}
```