1. (30) Write a complete, stand alone program in a single class called StringWork, which works as follows:

First, your program should include this array of Strings:

String[] strings = {"Jo","Jim","Dana","Wilma","Lana","Ned","Nathaniel","Jill","Joe"};

Your program should read in a non-negative int value from the keyboard - say n, and then print, in a column, all strings that are shorter (smaller length) than n. For example if n is 3 then only Jo should be printed.

Note: your program should include the strings array, as shown, but to save time you can simply enter this line in your program: String[] strings = {...as above...};

```java
import java.util.Scanner;
public class StringWork {
    Scanner scan = new Scanner(System.in);  
    String[] strings = {
    "Jo","Jim","Dana","Wilma","Lana","Ned", 
    "Nathaniel","Jill","Joe"};
    int n = scan.nextInt();
    for(int i =  0; i  < strings.length; i++)
    {
        if(strings[i].length() < num)
            System.out.println(strings[i]);
    } //end of for loop

    //could also use a for each loop
    for(String s: strings)
    {
        if((s.length() < num)
            System.out.println(s);
    } //end of for each loop
}
```

2. The method mystery, below, from the RecurFns class is recursive.

```java
public static void mystery(int[] s, int k){
    if((k >= 0) && (k < s.length)){
        System.out.print(s[k]);
        mystery(s,k-1);
    }
}
```
Suppose a is this array: \{2, 4, 6\}

What happens when you execute each of the following statements:

- RecurFns.mystery(a, -1);
  Nothing
- RecurFns.mystery(a, 0);
  2
- RecurFns.mystery(a, 1);
  42
- RecurFns.mystery(a, 2);
  642
- RecurFns.mystery(a, 3);
  Nothing

3. Suppose arr is any array of ints, and assume that arr has at least one element.

3a. Write a single statement that prints to the screen the very last element in arr.

```
System.out.println(arr[arr.length - 1]);
```

3b Write a code fragment (in general requiring multiple lines) that exchanges the value stored at the very last position in arr and the value at the first position (value at index position 0).

```
int temp = arr[0];
arr[0] = arr[arr.length - 1];
arr[arr.length - 1] = temp;
```

3c Write a code fragment that prints the average value in the array arr (in general this won’t be a whole number).

```
double average = 0.0;
for(int i = 0; i < arr.length; i++)
{
    average+=arr[i];
}
System.out.println(average/arr.length);
```

3d Write a code fragment that prints the largest value in the array arr (in case of ties, any will do).
// set max to the lowest number possible
int max = Integer.MIN_VALUE;

// can also set it to the first value in the array
max = arr[0];

for(int i = 0; i < arr.length; i++)
{
    if(arr[i] > max)
        max = arr[i];
}

System.out.println(max);

3e Use a for-each loop to print in a column all even values in the array arr.

for(int num: arr)
{
    if(num % 2 == 0)
        System.out.println(num)
}

4 Consider the following code fragment. Rewrite it using a while loop. Here s is any String.

    int count = 0;
    s = s.toLowerCase();
    for(int j=0; j < s.length(); j++){
        if(Character.isLetter(s.charAt(j))) count++;
    }
    System.out.println(count);

    int count = 0;
    s = s.toLowerCase();
    int j = 0; // initialize j outside of loop
    while(j < s.length())
    {
        if (Character.isLetter(s.charAt(j)))
            count++;
        j++; // increment j outside of if statement
    }
    System.out.println(count);
5. A phone company offers cell service, and uses this simple Phone class to model its accounts. It offers service according to three different plans, 1, 2, and 3.

```java
public class Phone{
    private String customer;
    private int plan;

    public Phone(String customer, int plan){
        this.customer = customer;
        this.plan = plan;
    }

    public String getCustomer(){return owner;}
    public int getPlan(){return plan;} // sets level of service
    public int setPlan(int p){plan = p;}
    public String toString(){return (customer + " " + plan);}  
}
```

5a A Phone plan object with a plan value of 1 is considered to be a low service plan. When the plan value is 2 or 3, the plan is considered a high service plan. Write a method, to be added to the Phone class, called lowService, which takes no arguments, and which returns true if its calling object has plan value 1; otherwise the method should return false.

```java
public boolean lowService()
{
    if(this.plan == 1)
        return true;
    else
        return false;
}
```

5b The phone company above has decided to offer landline service as well as cell service. To reflect this, write the MultiPlan class, a subclass of the Phone class with one new attribute, landline, of type boolean. When landline is true, the customer's phone service includes both cell service (at some plan level) and landline service. When landline is false, the customer is getting only cell service.

Your MultiPlan class must involve one extra attribute, landline. The MultiPlan constructor must use super. Also, there must be getLandline and setLandline methods. Finally, MultiPlan should have its own toString method, which returns a String that gives a customer's name, plan level, and landline status (true or false).
public class MultiPlan
{
    private boolean landline;

    public MultiPlan(String customer, int plan)
    {
        super(customer, plan);
        landline = false;
    }

    public void setLandline(boolean newLandline)
    {
        landline = newLandline;
    }

    public boolean getLandline()
    {
        return landline;
    }
}

5c In a single statement in a driver class make a MultiPlan object called danaAcct, with customer Dana, plan level 3, and no landline.

public class MultiPlanDriver
{
    public static void main(String[] args)
    {
        MultiPlan danaAcct = new MultiPlan("Dana", 3, false);
    }
}

5d Write static method fancyService in the MultiPlan class, which takes an array of MultiPlan objects as a parameter, and which prints to the screen the names of all customers with plans that include both the highest service (plan attribute value equals 3), and a landline.

public static void fancyService(MultiPlan[] plans)
{
    for(MultiPlan p : plans)
    {
        if(p.getPlan() == 3 && p.landline)
        {
            //need to print out just the name
            System.out.println(p.getCustomer());
        }
    }
}