SUNWAY

INT'L BUSINESS SCHOOL

Programme Name: **BCS HONS**

Course Code: **CSC 2515**

Course Name: **Object Oriented Programming**

Assignment / **Lab Sheet** / Project / Case Study No. **Recursion Labsheet**

Date of Submission: **1/13/2021**

Submitted By:

Student Name**: Dipesh Tha Shrestha**

IUKL ID: **041902900028**

Semester**: Third Semester**

Intake**: September 2019**

Submitted To:

Faculty Name**: Som Prasad Shrestha**

Department**: LMS**

1.

```
/*There are a number of bunnies and each bunny has two big floppy ears.  Compute the total
number of ears for all the bunnies recursively,  without using loops or multiplication.
*/
import java.util.Scanner; public class recursion1 {


static int bunny (int n) {


if (n == 1) {


return 2;
}


return 2 + bunny(n - 1);
}


public static void main(String[] args) {


Scanner sc = new Scanner(System.in);


int n = sc.nextInt();
System.out.println(bunny(n));
}
}
```

2.

```
/*
Write a recursive program that counts the number of “E” in a given string.
*/ import java.util.Scanner; public class recursion2 {

static int letter(String str){ int count=0;

if (str.length()==0) {

return 0;
}
if (str.charAt(0)=='e'){ count++;
}
return count+ letter(str.substring(1));
}
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
String str = sc.nextLine();
System.out.println(letter(str));

}

}
```

3.

```
/*

Some dogs are standing in a line, numbered 1, 2, 3, .....

The odd dogs (1, 3, ..) have normal two ears.

The even dogs (2, 4, ....) have 3 ears.

Recursively, return the number of ears in the dogs line 1,2, ..... n  without using loops or multiplication.


*/


import java.util.Scanner;


public class recursion3 {


static int dog(int n){


if (n==1)

{

return 2;
}
else if(n==2){ return 3;
}


if (n%2==0)
{
dog(n-1); return 3;
}
dog(n-1); return 2;
}


public static void main(String[] args) { Scanner sc = new
Scanner(System.in); int n = sc.nextInt();
System.out.println(dog(n));
}
}
```

4. /* fibonacci sequence
***/**

```
import java.util.Scanner; public class recursion4 {
static int febo(int n){


if (n==0) {
```

```
return 0;

}

if(n==1){ return 1;

}



return febo(n-1)+febo(n-2);



}



public static void main(String[] args) { Scanner sc = new

Scanner(System.in); int n = sc.nextInt(); int i=0;


while(i<n){ System.out.println(febo(i)); i++;

}

}

}
```

5.

/*

Given a string, compute recursively (no loops) a new string where all appearances of "pi" have been replaced by "3.14".

***/**

```
import java.util.Scanner; public class recursion5 {



static String changePi(String str) {



if (str.equals("") || str.length() < 2) return str; if (str.charAt(0) == 'p'

&& str.charAt(1) == 'i') return "3.14" + changePi(str.substring(2));

return str.charAt(0) + changePi(str.substring(1));

}

public static void main(String[] args) {

System.out.println(changePi("pippxxppiixipi"));

}



}
```