

100+

JAVA PROGRAMS CHEAT SHEET

Exercises



Solutions



Java



1. Write a Java program to print 'Hello' on screen and then print your name on a separate line.

Expected Output :

Hello
Juan Dela Cruz!

Solution:

```
public class Exercise1 {  
  
    public static void main(String[] args) {  
        System.out.println("Hello\nJuan Dela Cruz!");  
    }  
  
}
```

2. Write a Java program to print the sum of two numbers

Test Data:

24 + 26

Expected Output :

50

Solution:

```
public class Exercise2 {  
  
    public static void main(String[] args) {  
        System.out.println(24+26);  
    }  
  
}
```

3. Write a Java program to divide two numbers and print on the screen.

Test Data:

66/3

Expected Output :

22

Solution:

```
public class Exercise3 {  
  
    public static void main(String[] args) {  
        System.out.println(66/3);  
    }  
  
}
```

4. Write a Java program to print the result of the following operations.

Test Data:

- a. $-5 + 8 * 6$
- b. $(55+9) \% 9$
- c. $20 + -3*5 / 8$
- d. $5 + 15 / 3 * 2 - 8 \% 3$

Expected Output :

43
1
19
13

Solution:

```
public class Exercise4 {  
  
    public static void main(String[] args) {  
        System.out.println(-5 + 8 * 6);  
        System.out.println((55+9) % 9);  
        System.out.println(20 + -3*5 / 8);  
        System.out.println(5 + 15 / 3 * 2 - 8 % 3);  
    }  
  
}
```

5. Write a Java program that takes two numbers as input and display the product of two numbers.

Test Data:

Input first number: 25

Input second number: 5

Expected Output :

$25 \times 5 = 125$

Solution:

```
import java.util.Scanner;

public class Exercise5 {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);

        System.out.print("Input first number: ");
        int num1 = in.nextInt();

        System.out.print("Input second number: ");
        int num2 = in.nextInt();

        System.out.println(num1 + " x " + num2 + " = " + num1 * num2);
    }
}
```

6. Write a Java program to print the sum (addition), multiply, subtract, divide and remainder of two numbers.

Test Data:

Input first number: 125

Input second number: 24

Expected Output :

$125 + 24 = 149$

$125 - 24 = 101$

$125 \times 24 = 3000$

$125 / 24 = 5$

$125 \bmod 24 = 5$

```
import java.util.Scanner;

public class Exercise6 {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);

        System.out.print("Input first number: ");
        int num1 = in.nextInt();

        System.out.print("Input second number: ");
        int num2 = in.nextInt();

        System.out.println(num1 + " + " + num2 + " = " +
                           (num1 + num2));

        System.out.println(num1 + " - " + num2 + " = " +
                           (num1 - num2));

        System.out.println(num1 + " x " + num2 + " = " +
                           (num1 * num2));

        System.out.println(num1 + " / " + num2 + " = " +
                           (num1 / num2));

        System.out.println(num1 + " mod " + num2 + " = " +
                           (num1 % num2));
    }
}
```

7. Write a Java program that takes a number as input and prints its multiplication table upto 10.

Test Data:

Input a number: 8

Expected Output :

$8 \times 1 = 8$

$8 \times 2 = 16$

$8 \times 3 = 24$

...

$8 \times 10 = 80$

Sample Solution:

```
import java.util.Scanner;

public class Exercise7 {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);

        System.out.print("Input a number: ");
        int num1 = in.nextInt();

        for (int i=0; i< 10; i++){
            System.out.println(num1 + " x " + (i+1) + " = " +
                (num1 * (i+1)));
        }
    }
}
```

8. Write a Java program to display the following pattern.

Sample Pattern :

```
J   a   v   v   a  
J   a a   v   v   a a  
J J  aaaaa  V V  aaaaa  
JJ  a       a   V   a       a
```

Solution:

```
public class Exercise8 {  
  
    public static void main(String[] args) {  
  
        System.out.println("   J   a   v   v   a ");  
        System.out.println("   J   a a   v   v   a a");  
        System.out.println("J J  aaaaa  V V  aaaaa");  
        System.out.println("JJ  a       a   V   a       a");  
    }  
}
```

9. Write a Java program to compute the specified expressions and print the output.

Test Data:

$((25.5 * 3.5 - 3.5 * 3.5) / (40.5 - 4.5))$

Expected Output

2.13888888888889

Solution:

```
public class Exercise9 {  
    public static void main(String[] arg) {  
        System.out.println((25.5 * 3.5 - 3.5 * 3.5) / (40.5 - 4.5));  
    }  
}
```

10. Write a Java program to compute a specified formula.

Specified Formula :

$$4.0 * (1 - (1.0/3) + (1.0/5) - (1.0/7) + (1.0/9) - (1.0/11))$$

Expected Output

2.9760461760461765

Solution:

```
public class Exercise10 {  
  
    public static void main(String[] args) {  
  
        double pi = 4.0 * (1 - (1.0/3) + (1.0/5) - (1.0/7) + (1.0/9) -  
        (1.0/11));  
        System.out.println(pi); //  
    }  
}
```

11. Write a Java program to print the area and perimeter of a circle.

Test Data:

Radius = 7.5

Expected Output

Perimeter is = 47.12388980384689

Area is = 176.71458676442586

Solution:

```
public class Exercise11 {  
    private static final double radius = 7.5;  
  
    public static void main(String[] args) {  
  
        double perimeter = 2 * Math.PI * radius;  
        double area = Math.PI * radius * radius;  
  
        System.out.println("Perimeter is = " + perimeter);  
        System.out.println("Area is = " + area);  
    }  
}
```

12. Write a Java program that takes three numbers as input to calculate and print the average of the numbers.

Test Data:

Input first number: 10
Input second number: 20
Input third number: 30
Input fourth number: 40
Enter fifth number: 50

Solution:

```
import java.util.Scanner;

public class Exercise12 {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);

        System.out.print("Input first number: ");
        int num1 = in.nextInt();

        System.out.print("Input second number: ");
        int num2 = in.nextInt();

        System.out.print("Input third number: ");
        int num3 = in.nextInt();

        System.out.print("Input fourth number: ");
        int num4 = in.nextInt();

        System.out.print("Enter fifth number: ");
        int num5 = in.nextInt();

        System.out.println("Average of five numbers is: " +
            (num1 + num2 + num3 + num4 + num5) / 5);
    }
}
```

13. Write a Java program to print the area and perimeter of a rectangle

Test Data:

Width = 5.5 Height = 8.5

Expected Output

Area is $5.6 * 8.5 = 47.60$

Perimeter is $2 * (5.6 + 8.5) = 28.20$

Solution:

```
public class Exercise13 {  
  
    public static void main(String[] strings) {  
  
        final double width = 5.6;  
        final double height = 8.5;  
  
        double perimeter = 2*(height + width);  
  
        double area = width * height;  
  
        System.out.printf("Perimeter is 2*%.1f + %.1f = %.2f \n",  
                          height, width, perimeter);  
  
        System.out.printf("Area is %.1f * %.1f = %.2f \n", width,  
                          height, area);  
    }  
}
```

14. Write a Java program to print an American flag on the screen.

Expected Output

Solution:

```
public class Exercise14 {  
    public static void main(String[] args)  
    {  
        System.out.println("* * * * *  
=====");  
        System.out.println(" * * * * *  
=====");  
        System.out.println("* * * * * *  
=====");  
        System.out.println(" * * * * *  
=====");  
        System.out.println("* * * * * *  
=====");  
        System.out.println(" * * * * *  
=====");  
        System.out.println("* * * * * *  
=====");  
        System.out.println(" * * * * *  
=====");  
        System.out.println("* * * * * *  
=====");  
        System.out.println(" * * * * *  
=====");  
    }  
}
```

```
System.out.println("=====");  
System.out.println("=====");  
System.out.println("=====");  
System.out.println("=====");  
System.out.println("=====");  
System.out.println("=====");  
}  
}
```

15. Write a Java program to swap two variables.

Swapping two variables refers to mutually exchanging the values of the variables. Generally, this is done with the data in memory.

Solution:

```
public class Exercise15 {  
  
    public static void main(String[] args) {  
  
        int a, b, temp;  
        a = 15;  
        b = 27;  
        System.out.println("Before swapping : a, b = "+a+", "+ b);  
        temp = a;  
        a = b;  
        b = temp;  
        System.out.println("After swapping : a, b = "+a+", "+ b);  
    }  
}
```

16. Write a Java program to print a face.

Expected Output

```
+*****+
[ |  o  o  | ]
|    ^    |
|   '-'   |
+-----+
```

Solution:

```
public class Exercise16 {
    public static void main(String[] args)
    {
        System.out.println(" +\"\\\"\\\"\\\"\\\"");
        System.out.println("[ |  o  o  | ]");
        System.out.println(" |    ^    | ");
        System.out.println(" |   '-'   | ");
        System.out.println(" +-----+");
    }
}
```

17. Write a Java program to add two binary numbers.

Input Data:

Input first binary number: 10

Input second binary number: 11

Expected Output

Sum of two binary numbers: 101

Solution:

```
import java.util.Scanner;
public class Exercise17 {
    public static void main(String[] args)
    {
        long binary1, binary2;
        int i = 0, remainder = 0;
        int[] sum = new int[20];
        Scanner in = new Scanner(System.in);

        System.out.print("Input first binary number: ");
        binary1 = in.nextLong();
        System.out.print("Input second binary number: ");
        binary2 = in.nextLong();

        while (binary1 != 0 || binary2 != 0)
        {
            sum[i++] = (int)((binary1 % 10 + binary2 % 10 + remainder) % 2);
            remainder = (int)((binary1 % 10 + binary2 % 10 + remainder) / 2);
            binary1 = binary1 / 10;
            binary2 = binary2 / 10;
        }
        if (remainder != 0) {
            sum[i++] = remainder;
        }
        --i;
        System.out.print("Sum of two binary numbers: ");
        while (i >= 0) {
            System.out.print(sum[i--]);
        }
        System.out.print("\n");
    }
}
```

18. Write a Java program to multiply two binary numbers.

Input Data:

Input the first binary number: 10

Input the second binary number: 11

Expected Output

Product of two binary numbers: 110

Solution:

```
import java.util.Scanner;
public class Exercise18 {
    public static void main(String[] args)
    {
        long binary1, binary2, multiply = 0;
        int digit, factor = 1;
        Scanner in = new Scanner(System.in);
        System.out.print("Input the first binary number: ");
        binary1 = in.nextLong();
        System.out.print("Input the second binary number: ");
        binary2 = in.nextLong();
        while (binary2 != 0)
        {
            digit = (int)(binary2 % 10);
            if (digit == 1)
            {
                binary1 = binary1 * factor;
                multiply = binaryproduct((int) binary1, (int) multiply);
            }
            else
            {
                binary1 = binary1 * factor;
            }
            binary2 = binary2 / 10;
            factor = 10;
        }
        System.out.print("Product of two binary numbers: " + multiply + "\n");
    }
    static int binaryproduct(int binary1, int binary2)
    {
        int i = 0, remainder = 0;
        int[] sum = new int[20];
```

```
int binary_prod_result = 0;

while (binary1 != 0 || binary2 != 0)
{
    sum[i++] = (binary1 % 10 + binary2 % 10 + remainder) % 2;
    remainder = (binary1 % 10 + binary2 % 10 + remainder) / 2;
    binary1 = binary1 / 10;
    binary2 = binary2 / 10;
}
if (remainder != 0)
{
    sum[i++] = remainder;
}
--i;
while (i >= 0)
{
    binary_prod_result = binary_prod_result * 10 + sum[i--];
}
return binary_prod_result;
}
}
```

19. Write a Java program to convert a decimal number to binary number.

Input Data:

Input a Decimal Number : 5

Expected Output

Binary number is: 101

Solution:

```
import java.util.Scanner;
public class Exercise19 {
    public static void main(String args[])
    {
        int dec_num, quot, i=1, j;
        int bin_num[] = new int[100];
        Scanner scan = new Scanner(System.in);

        System.out.print("Input a Decimal Number : ");
        dec_num = scan.nextInt();

        quot = dec_num;

        while(quot != 0)
        {
            bin_num[i++] = quot%2;
            quot = quot/2;
        }

        System.out.print("Binary number is: ");
        for(j=i-1; j>0; j--)
        {
            System.out.print(bin_num[j]);
        }
        System.out.print("\n");
    }
}
```

20. Write a Java program to convert a decimal number to hexadecimal number.

Input Data:

Input a decimal number: 15

Expected Output

Hexadecimal number is : F

```
import java.util.Scanner;
public class Exercise20 {
    public static void main(String args[])
    {
        int dec_num, rem;
        String hexdec_num="";
        /* hexadecimal number digits */

        char
hex[]={'0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'};
        Scanner in = new Scanner(System.in);

        System.out.print("Input a decimal number: ");
        dec_num = in.nextInt();

        while(dec_num>0)
        {
            rem = dec_num%16;
            hexdec_num = hex[rem] + hexdec_num;
            dec_num = dec_num/16;
        }
        System.out.print("Hexadecimal number is : "+hexdec_num+"\n");
    }
}
```

21. Write a Java program to convert a decimal number to octal number.

Input Data:

Input a Decimal Number: 15

Expected Output

Octal number is: 17

Solution:

```
import java.util.Scanner;
public class Exercise21 {
    public static void main(String args[])
    {
        int dec_num, rem, quot, i=1, j;
        int oct_num[] = new int[100];
        Scanner scan = new Scanner(System.in);

        System.out.print("Input a Decimal Number: ");
        dec_num = scan.nextInt();

        quot = dec_num;

        while(quot != 0)
        {
            oct_num[i++] = quot%8;
            quot = quot/8;
        }

        System.out.print("Octal number is: ");
        for(j=i-1; j>0; j--)
        {
            System.out.print(oct_num[j]);
        }
        System.out.print("\n");
    }
}
```

22. Write a Java program to convert a binary number to decimal number

Input Data:

Input a binary number: 100

Expected Output

Decimal Number: 4

Solution:

```
import java.util.Scanner;
public class Exercise22 {
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        long binaryNumber, decimalNumber = 0, j = 1, remainder;
        System.out.print("Input a binary number: ");
        binaryNumber = sc.nextLong();

        while (binaryNumber != 0)
        {
            remainder = binaryNumber % 10;
            decimalNumber = decimalNumber + remainder * j;
            j = j * 2;
            binaryNumber = binaryNumber / 10;
        }
        System.out.println("Decimal Number: " + decimalNumber);
    }
}
```

23. Write a Java program to convert a binary number to hexadecimal number.

Input Data:

Input a Binary Number: 1101

Expected Output

HexaDecimal value: D

Solution:

```
import java.util.Scanner;
public class Exercise23 {
    public static void main(String[] args)
    {
        int[] hex = new int[1000];
        int i = 1, j = 0, rem, dec = 0, bin;
        Scanner in = new Scanner(System.in);
        System.out.print("Input a Binary Number: ");
        bin = in.nextInt();
        while (bin > 0) {
            rem = bin % 2;
            dec = dec + rem * i;
            i = i * 2;
            bin = bin / 10;
        }
        i = 0;
        while (dec != 0) {
            hex[i] = dec % 16;
            dec = dec / 16;
            i++;
        }
        System.out.print("HexaDecimal value: ");
        for (j = i - 1; j >= 0; j--)
        {
            if (hex[j] > 9)
            {
                System.out.print((char)(hex[j] + 55)+"\n");
            } else
            {
                System.out.print(hex[j]+"\n");
            }
        }
    }
}
```

24. Write a Java program to convert a binary number to a Octal number.

Input Data:

Input a Binary Number: 111

Expected Output

Octal number: 7

Solution:

```
import java.util.*;
public class Exercise24 {
public static void main(String[] args)
{
    int binnum, binnum1,rem, decnum=0, quot, i=1, j;
    int octnum[] = new int[100];
    Scanner scan = new Scanner(System.in);
    System.out.print("Input a Binary Number : ");
    binnum = scan.nextInt();
    binnum1=binnum;

    while(binnum > 0)
    {
        rem = binnum % 10;
        decnum = decnum + rem*i;
        //System.out.println(rem);
        i = i*2;
        binnum = binnum/10;
    }

    i=1;
    quot = decnum;

    while(quot > 0)
    {
        octnum[i++] = quot % 8;
        quot = quot / 8;
    }

    System.out.print("Equivalent Octal Value of " +binnum1+ " is
:");
    for(j=i-1; j>0; j--)
    {
        System.out.print(octnum[j]);
    }
}
```

```
System.out.print("\n");  
}  
}
```

25. Write a Java program to convert a octal number to a decimal number.

Input Data:

Input any octal number: 10

Expected Output

Equivalent decimal number: 8

Solution:

```
import java.util.Scanner;

public class Exercise25 {

    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        long octal_num, decimal_num = 0;
        int i = 0;
        System.out.print("Input any octal number: ");
        octal_num = in.nextLong();
        while (octal_num != 0)
        {
            decimal_num = (long)(decimal_num + (octal_num % 10) *
Math.pow(8, i++));
            octal_num = octal_num / 10;
        }
        System.out.print("Equivalent decimal number: " +
decimal_num+"\n");
    }
}
```

26. Write a Java program to convert a octal number to a binary number.

Input Data:

Input any octal number: 7

Expected Output

Equivalent binary number: 111

Solution:

```
import java.util.Scanner;

public class Exercise26 {
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        int[] octal_numvalues = {0, 1, 10, 11, 100, 101, 110, 111};
        long octal_num, tempoctal_num, binary_num, place;
        int rem;
        System.out.print("Input any octal number: ");
        octal_num = in.nextLong();
        tempoctal_num = octal_num;
        binary_num = 0;
        place = 1;
        while (tempoctal_num != 0)
        {
            rem = (int)(tempoctal_num % 10);
            binary_num = octal_numvalues[rem] * place + binary_num;
            tempoctal_num /= 10;
            place *= 1000;
        }
        System.out.print("Equivalent binary number: " + binary_num + "\n");
    }
}
```

27. Write a Java program to convert a octal number to a hexadecimal number.

Input Data:

Input a octal number : 100

Expected Output

Equivalent hexadecimal number: 40

Solution:

```
import java.util.Scanner;

public class Exercise27 {
    public static void main(String args[])
    {
        String octal_num, hex_num;
        int decnum;
        Scanner in = new Scanner(System.in);

        System.out.print("Input a octal number : ");
        octal_num = in.nextLine();

        decnum = Integer.parseInt(octal_num, 8);
        hex_num = Integer.toHexString(decnum);

        System.out.print("Equivalent hexadecimal number: "+
hex_num+"\n");
    }
}
```

28. Write a Java program to convert a hexadecimal to a decimal number.

Input Data:

Input a hexadecimal number: 25

Expected Output

Equivalent decimal number is: 37

Solution:

```
import java.util.Scanner;

public class Exercise28 {
    public static int hex_to_decimal(String s)
    {
        String digits = "0123456789ABCDEF";
        s = s.toUpperCase();
        int val = 0;
        for (int i = 0; i < s.length(); i++)
        {
            char c = s.charAt(i);
            int d = digits.indexOf(c);
            val = 16*val + d;
        }
        return val;
    }
    public static void main(String args[])
    {
        String hexdec_num;
        int dec_num;
```

```
Scanner scan = new Scanner(System.in);

System.out.print("Input a hexadecimal number: ");
hexdec_num = scan.nextLine();

dec_num = hex_to_decimal(hexdec_num);

System.out.print("Equivalent decimal number is: " + dec_num + "\n");
}

}
```

29. Write a Java program to convert a hexadecimal to a binary number.

Input Data:

Enter Hexadecimal Number : 37

Expected Output

Equivalent Binary Number is: 110111

Solution:

```
import java.util.Scanner;
public class Exercise29 {
    public static int hex_to_decimal(String s)
    {
        String digits = "0123456789ABCDEF";
        s = s.toUpperCase();
        int val = 0;
        for (int i = 0; i < s.length(); i++)
        {
            char c = s.charAt(i);
            int d = digits.indexOf(c);
            val = 16*val + d;
        }
        return val;
    }
    public static void main(String args[])
    {
        String hexdec_num;
        int dec_num, i=1, j;
        int bin_num[] = new int[100];
        Scanner scan = new Scanner(System.in);

        System.out.print("Enter Hexadecimal Number : ");
        hexdec_num = scan.nextLine();

        /* convert hexadecimal to decimal */
        dec_num = hex_to_decimal(hexdec_num);

        /* convert decimal to binary */
        while(dec_num != 0)
        {
            bin_num[i++] = dec_num%2;
            dec_num = dec_num/2;
        }
    }
}
```

```
System.out.print("Equivalent Binary Number is: ");
for(j=i-1; j>0; j--)
{
    System.out.print(bin_num[j]);
}
System.out.print("\n");
}
```

30. Write a Java program to convert a hexadecimal to a octal number.

Input Data:

Input a hexadecimal number: 40

Expected Output

Equivalent of octal number is: 100

Solution:

```
import java.util.Scanner;

public class Exercise30 {
    public static int hex_to_decimal(String s)
    {
        String digits = "0123456789ABCDEF";
        s = s.toUpperCase();
        int val = 0;
        for (int i = 0; i < s.length(); i++)
        {
            char c = s.charAt(i);
            int d = digits.indexOf(c);
            val = 16*val + d;
        }
        return val;
    }
    public static void main(String args[])
    {
        String hexdec_num;
        int dec_num, i=1, j;
        int octal_num[] = new int[100];
        Scanner in = new Scanner(System.in);

        System.out.print("Input a hexadecimal number: ");
        hexdec_num = in.nextLine();

        // Convert hexadecimal to decimal

        dec_num = hex_to_decimal(hexdec_num);

        //Convert decimal to octal

        while(dec_num != 0)
        {
```

```
    octal_num[i++] = dec_num%8;
    dec_num = dec_num/8;
}

System.out.print("Equivalent of octal number is: ");
for(j=i-1; j>0; j--)
{
    System.out.print(octal_num[j]);
}
System.out.print("\n");
}
}
```

31. Write a Java program to check whether Java is installed on your computer.

Expected Output

```
Java Version: 1.8.0_71
Java Runtime Version: 1.8.0_71-b15
Java Home: /opt/jdk/jdk1.8.0_71/jre
Java Vendor: Oracle Corporation
Java Vendor URL: http://Java.oracle.com/
Java Class Path: .
```

Solution:

```
public class Exercise31 {
    public static void main(String[] args) {
        System.out.println("\nJava Version:
"+System.getProperty("java.version"));
        System.out.println("Java Runtime Version:
"+System.getProperty("java.runtime.version"));
        System.out.println("Java Home: "+System.getProperty("java.home"));
        System.out.println("Java Vendor:
"+System.getProperty("java.vendor"));
        System.out.println("Java Vendor URL:
"+System.getProperty("java.vendor.url"));
        System.out.println("Java Class Path:
"+System.getProperty("java.class.path")+"\n");
    }
}
```

32. Write a Java program to compare two numbers.

Input Data:

Input first integer: 25

Input second integer: 39

Expected Output

25 != 39

25 < 39

25 <= 39

Solution:

```
import java.util.Scanner;
public class Exercise32 {
    public static void main( String args[] )
    {
        // create Scanner to obtain input from command window
        Scanner input = new Scanner(System.in);
        int number1; // first number to compare
        int number2; // second number to compare

        System.out.print( "Input first integer: " ); // prompt
        number1 = input.nextInt(); // read first number from user

        System.out.print( "Input second integer: " ); // prompt
        number2 = input.nextInt(); // read second number from user

        if ( number1 == number2 )
            System.out.printf( "%d == %d\n", number1, number2 );
        if ( number1 != number2 )
            System.out.printf( "%d != %d\n", number1, number2 );
        if ( number1 < number2 )
            System.out.printf( "%d < %d\n", number1, number2 );
        if ( number1 > number2 )
            System.out.printf( "%d > %d\n", number1, number2 );
        if ( number1 <= number2 )
            System.out.printf( "%d <= %d\n", number1, number2 );
        if ( number1 >= number2 )
            System.out.printf( "%d >= %d\n", number1, number2 );
    }
}
```

33. Write a Java program and compute the sum of the digits of an integer.

Input Data:

Input an integer: 25

Expected Output

The sum of the digits is: 7

Solution:

```
import java.util.Scanner;
public class Exercise33 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Input an integer: ");
        long n = input.nextLong();
        System.out.println("The sum of the digits is: " +
sumDigits(n));

    }

    public static int sumDigits(long n) {
        int sum = 0;
        while (n != 0) {
            sum += n % 10;
            n /= 10;
        }
        return sum;
    }
}
```

34. Write a Java program to compute the area of a hexagon.

Area of a hexagon = $(6 * s^2)/(4 * \tan(\pi/6))$
where s is the length of a side

Input Data:

Input the length of a side of the hexagon: 6

Expected Output

The area of the hexagon is: 93.53074360871938

Solution:

```
import java.util.Scanner;
public class Exercise34 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Input the length of a side of the hexagon:");
    };
    double s = input.nextDouble();
    System.out.print("The area of the hexagon is: " +
hexagonArea(s)+"\n");
}
public static double hexagonArea(double s) {
    return (6*(s*s))/(4*Math.tan(Math.PI/6));
}
}
```

35. Write a Java program to compute the area of a polygon.

Area of a polygon = $(n*s^2)/(4*tan(\pi/n))$

where n is n-sided polygon and s is the length of a side

Input Data:

Input the number of sides on the polygon: 7

Input the length of one of the sides: 6

Expected Output

The area is: 130.82084798405722

Solution:

```
import java.util.Scanner;

public class Exercise35 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);
        System.out.print("Input the number of sides on the polygon:");
        int ns = input.nextInt();
        System.out.print("Input the length of one of the sides: ");
        double side = input.nextDouble();
        System.out.print("The area is: " + polygonArea(ns,
side)+"\n");
    }

    public static double polygonArea(int ns, double side) {
        return (ns * (side * side)) / (4.0 * Math.tan((Math.PI /
ns)));
    }
}
```

36. Write a Java program to compute the distance between two points on the surface of earth.

Distance between the two points [(x1,y1) & (x2,y2)]

$d = \text{radius} * \arccos(\sin(x1) * \sin(x2) + \cos(x1) * \cos(x2) * \cos(y1 - y2))$

Radius of the earth r = 6371.01 Kilometers

Input Data:

Input the latitude of coordinate 1: 25

Input the longitude of coordinate 1: 35

Input the latitude of coordinate 2: 35.5

Input the longitude of coordinate 2: 25.5

Expected Output

The distance between those points is: 1480.0848451069087 km

Solution:

```
import java.util.Scanner;

public class Exercise36 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);
        System.out.print("Input the latitude of coordinate 1: ");
        double lat1 = input.nextDouble();
        System.out.print("Input the longitude of coordinate 1: ");
        double lon1 = input.nextDouble();
        System.out.print("Input the latitude of coordinate 2: ");
        double lat2 = input.nextDouble();
        System.out.print("Input the longitude of coordinate 2: ");
        double lon2 = input.nextDouble();

        System.out.print("The distance between those points is: " +
distance_Between_LatLong(lat1, lon1, lat2, lon2) + " km\n");
    }

    // Points will be converted to radians before calculation
}
```

```
public static double distance_Between_LatLong(double lat1, double
lon1, double lat2, double lon2) {
    lat1 = Math.toRadians(lat1);
    lon1 = Math.toRadians(lon1);
    lat2 = Math.toRadians(lat2);
    lon2 = Math.toRadians(lon2);

    double earthRadius = 6371.01; //Kilometers
    return earthRadius * Math.acos(Math.sin(lat1)*Math.sin(lat2) +
Math.cos(lat1)*Math.cos(lat2)*Math.cos(lon1 - lon2));
}
```

37. Write a Java program to reverse a string.

Input Data:

Input a string: The quick brown fox

Expected Output

Reverse string: xof nworb kciuq ehT

Solution:

```
import java.util.Scanner;
public class Exercise37 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Input a string: ");
        char[] letters = scanner.nextLine().toCharArray();
        System.out.print("Reverse string: ");
        for (int i = letters.length - 1; i >= 0; i--) {
            System.out.print(letters[i]);
        }
        System.out.print("\n");
    }
}
```

- 38.** Write a Java program to count the letters, spaces, numbers and other characters of an input string.

Expected Output

```
The string is : Aa kiu, I swd skiego 236587. GH kiu: sieo??  
25.33  
letter: 23  
space: 9  
number: 10  
other: 6
```

Solution:

```
import java.util.Scanner;  
public class Exercise38 {  
  
    public static void main(String[] args) {  
        String test = "Aa kiu, I swd skiego 236587. GH kiu: sieo??  
25.33";  
        count(test);  
  
    }  
    public static void count(String x){  
        char[] ch = x.toCharArray();  
        int letter = 0;  
        int space = 0;  
        int num = 0;  
        int other = 0;  
        for(int i = 0; i < x.length(); i++){  
            if(Character.isLetter(ch[i])){  
                letter++;  
            }  
            else if(Character.isDigit(ch[i])){  
                num++;  
            }  
            else if(Character.isSpaceChar(ch[i])){  
                space++;  
            }  
            else{  
                other++;  
            }  
        }  
        System.out.println("The string is : Aa kiu, I swd skiego  
236587. GH kiu: sieo?? 25.33");  
        System.out.println("letter: " + letter);  
    }  
}
```

```
        System.out.println("space: " + space);
        System.out.println("number: " + num);
        System.out.println("other: " + other);
    }
}
```

39. Write a Java program to create and display unique three-digit number using 1, 2, 3, 4. Also count how many three-digit numbers are there.

Expected Output

```
123  
124  
...  
  
431  
432  
Total number of the three-digit-number is 24.
```

Solution:

```
import java.util.Scanner;  
  
public class Exercise39 {  
  
    public static void main(String[] args) {  
        int amount = 0;  
        for(int i = 1; i <= 4; i++){  
            for(int j = 1; j <= 4; j++){  
                for(int k = 1; k <= 4; k++){  
                    if(k != i && k != j && i != j){  
                        amount++;  
                        System.out.println(i + " " + j + " " +  
k);  
                    }  
                }  
            }  
        }  
        System.out.println("Total number of the three-digit-number  
is " + amount);  
    }  
}
```

40. Write a Java program to list the available character sets in charset objects.

Expected Output

List of available character sets:

Big5
Big5-HKSCS
CESU-8
EUC-JP
EUC-KR
GB18030
GB2312
GBK

...

x-SJIS_0213
x-UTF-16LE-BOM
X-UTF-32BE-BOM
X-UTF-32LE-BOM
x-windows-50220
x-windows-50221
x-windows-874
x-windows-949
x-windows-950
x-windows-iso2022jp

Solution:

```
import java.nio.charset.Charset;
public class Exercise40 {
    public static void main(String[] args) {
        System.out.println("List of available character sets: ");
        for (String str : Charset.availableCharsets().keySet()) {
            System.out.println(str);
        }
    }
}
```

41. Write a Java program to print the ascii value of a given character.

Expected Output

The ASCII value of Z is :90

Solution:

```
public class Exercise41 {  
  
    public static void main(String[] String) {  
        int chr = 'Z';  
        System.out.println("The ASCII value of Z is :" + chr);  
    }  
}
```

42. Write a Java program to input and display your password.

Expected Output

Input your Password:
Your password was: abc@123

Solution:

```
import java.io.Console;
public class Example42 {
    public static void main(String[] args) {
        Console cons;
        if ((cons = System.console()) != null) {
            char[] pass_ward = null;
            try {
                pass_ward = cons.readPassword("Input your
Password:");
                System.out.println("Your password was: " + new
String(pass_ward));
            } finally {
                if (pass_ward != null) {
                    java.util.Arrays.fill(pass_ward, ' ');
                }
            }
        } else {
            throw new RuntimeException("Can't get password...No
console");
        }
    }
}
```

43. Write a Java program to print the following string in a specific format (see the output).

Sample Output

```
Twinkle, twinkle, little star,  
    How I wonder what you are!  
        Up above the world so high,  
            Like a diamond in the sky.  
Twinkle, twinkle, little star,  
    How I wonder what you are
```

Solution:

```
public class Example43 {  
    public static void main(String[] args) {  
  
        System.out.println("\nTwinkle, twinkle, little star, \n\tHow I  
wonder what you are! \n\t\tUp above the world so high, \n\t\t\tLike a  
diamond in the sky. \nTwinkle, twinkle, little star, \n\tHow I wonder  
what you are!\n\n");  
    }  
}
```

44. Write a Java program that accepts an integer (n) and computes the value of $n+nn+nnn$.

Sample Output:

```
Input number: 5
5 + 55 + 555
```

Solution:

```
import java.util.Scanner;
public class Exercise44 {
    public static void main(String[] args) {

        int n;
        char s1, s2, s3;
        Scanner in = new Scanner(System.in);
        System.out.print("Input number: ");
        n = in.nextInt();
        System.out.printf("%d + %d%d + %d%d%d\n", n, n, n, n, n, n);
    }
}
```

45. Write a Java program to find the size of a specified file.

Sample Output:

```
/home/students/abc.txt  : 0 bytes  
/home/students/test.txt : 0 bytes
```

Solution:

```
import java.util.Scanner;  
import java.io.File;  
public class Exercise45 {  
  
    public static void main(String[] args) {  
        System.out.println("/home/students/abc.txt  : " + new  
File("abc.txt").length() + " bytes");  
        System.out.println("/home/students/test.txt : " + new  
File("test.txt").length() + " bytes");  
    }  
}
```

46. Write a Java program to display the system time.

Sample Output:

Current Date time: Fri Jun 16 14:17:40 IST 2017

Solution:

```
public class Exercise46 {  
  
    public static void main(String[] args){  
        System.out.format("\nCurrent Date time: %tc%n\n",  
System.currentTimeMillis());  
    }  
}
```

47. Write a Java program to display the current date time in specific format.

Sample Output:

Now: 2017/06/16 08:52:03.066

Solution:

```
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.TimeZone;

public class Exercise47 {

    public static void main(String args[]) {
        SimpleDateFormat cdt = new SimpleDateFormat("yyyy/MM/dd
HH:mm:ss.SSS");

        cdt.setCalendar(Calendar.getInstance(TimeZone.getTimeZone("GMT")));
    }

    System.out.println("\nNow:
"+cdt.format(System.currentTimeMillis()));
}
}
```

48. Write a Java program to print the odd numbers from 1 to 99. Prints one number per line.

Sample Output:

```
1  
3  
5  
7  
9  
11  
....  
  
91  
93  
95  
97  
99
```

Solution:

```
import java.util.*;  
public class Exercise48 {  
    public static void main(String[] args){  
        for (int i = 1; i < 100; i++) {  
            if (i % 2 != 0) {  
                System.out.println(i);  
            }  
        }  
    }  
}
```

- 49.** Write a Java program to accept a number and check the number is even or not. Prints 1 if the number is even or 0 if the number is odd.

Sample Output:

```
Input a number: 20  
1
```

Solution:

```
import java.util.*;  
public class Exercise49 {  
    public static void main(String[] args){  
        Scanner in = new Scanner(System.in);  
        System.out.print("Input a number: ");  
        int n = in.nextInt();  
        if (n % 2 == 0) {  
            System.out.println(1);  
        }  
        else {  
            System.out.println(0);  
        }  
    }  
}
```

50. Write a Java program to print numbers between 1 to 100 which are divisible by 3, 5 and by both.

Sample Output:

```
Divided by 3:  
3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51,  
54, 57  
, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96, 99,  
  
Divided by 5:  
5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80,  
85, 90,  
95,  
  
Divided by 3 & 5:  
15, 30, 45, 60, 75, 90,
```

Solution:

```
public class Exercise50 {  
    public static void main(String args[]) {  
        System.out.println("\nDivided by 3: ");  
        for (int i=1; i<100; i++) {  
            if (i%3==0)  
                System.out.print(i +", ");  
        }  
  
        System.out.println("\n\nDivided by 5: ");  
        for (int i=1; i<100; i++) {  
            if (i%5==0) System.out.print(i +", ");  
        }  
  
        System.out.println("\n\nDivided by 3 & 5: ");  
        for (int i=1; i<100; i++) {  
            if (i%3==0 && i%5==0) System.out.print(i +", ");  
        }  
        System.out.println("\n");  
    }  
}
```

51. Write a Java program to convert a string to an integer in Java.

Sample Output:

```
Input a number(string): 25
The integer value is: 25
```

Solution:

```
import java.util.*;
public class Exercise51 {

    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        System.out.print("Input a number(string): ");
        String str1 = in.nextLine();
        int result = Integer.parseInt(str1);
        System.out.printf("The integer value is: %d",result);
        System.out.printf("\n");
    }
}
```

52. Write a Java program to calculate the sum of two integers and return true if the sum is equal to a third integer.

Sample Output:

```
Input the first number : 5
Input the second number: 10
Input the third number : 15
The result is: true
```

Solution:

```
import java.util.*;
public class Exercise52 {
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        System.out.print("Input the first number : ");
        int x = in.nextInt();
        System.out.print("Input the second number: ");
        int y = in.nextInt();
        System.out.print("Input the third number : ");
        int z = in.nextInt();
        System.out.print("The result is: "+sumoftwo(x, y, z));
        System.out.print("\n");
    }

    public static boolean sumoftwo(int p, int q, int r)
    {
        return ((p + q) == r || (q + r) == p || (r + p) == q);
    }
}
```

53. Write a Java program that accepts three integers from the user and return true if the second number is greater than first number and third number is greater than second number. If "abc" is true second number does not need to be greater than first number.

Sample Output:

```
Input the first number : 5
Input the second number: 10
Input the third number : 15
The result is: true
```

Solution:

```
import java.util.*;
public class Exercise53 {
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        System.out.print("Input the first number : ");
        int x = in.nextInt();
        System.out.print("Input the second number: ");
        int y = in.nextInt();
        System.out.print("Input the third number : ");
        int z = in.nextInt();
        System.out.print("The result is: "+test(x, y, z,true));
        System.out.print("\n");
    }

    public static boolean test(int p, int q, int r, boolean xyz)
    {
        if(xyz)
            return (r > q);
        return (q > p && r > q);
    }
}
```

54. Write a Java program that accepts three integers from the user and return true if two or more of them (integers) have the same rightmost digit. The integers are non-negative.

Sample Output:

```
Input the first number : 5
Input the second number: 10
Input the third number : 15
The result is: true
```

Solution:

```
import java.util.*;
public class Exercise54 {
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        System.out.print("Input the first number : ");
        int x = in.nextInt();
        System.out.print("Input the second number: ");
        int y = in.nextInt();
        System.out.print("Input the third number : ");
        int z = in.nextInt();
        System.out.print("The result is: "+test_last_digit(x, y,
z,true));
        System.out.print("\n");
    }

    public static boolean test_last_digit(int p, int q, int r, boolean
xyz)
    {
        return (p % 10 == q % 10) || (p % 10 == r % 10) || (q % 10
== r % 10);
    }
}
```

55. Write a Java program to convert seconds to hour, minute and seconds.

Sample Output:

```
Input seconds: 86399  
23:59:59
```

Solution:

```
import java.util.*;  
public class Exercise55 {  
    public static void main(String[] args)  
    {  
        Scanner in = new Scanner(System.in);  
        System.out.print("Input seconds: ");  
        int seconds = in.nextInt();  
        int p1 = seconds % 60;  
        int p2 = seconds / 60;  
        int p3 = p2 % 60;  
        p2 = p2 / 60;  
        System.out.print( p2 + ":" + p3 + ":" + p1);  
        System.out.print("\n");  
    }  
}
```

56. Write a Java program to find the number of integers within the range of two specified numbers and that are divisible by another number.

For example x = 5, y=20 and p =3, find the number of integers within the range x..y and that are divisible by p i.e. { i : $x \leq i \leq y, i \bmod p = 0$ }

Sample Output:

5

Solution:

```
import java.util.*;
public class Exercise56 {
    public static void main(String[] args){
        int x = 5;
        int y = 20;
        int p = 3;
        System.out.println(result(x,y,p));
    }
    public static int result(int x, int y, int p) {
        if (x%p == 0)
            return( y/p - x/p + 1);
        return(y/p - x/p);
    }
}
```

57. Write a Java program to accepts an integer and count the factors of the number.

Sample Output:

```
Input an integer: 25
3
```

Solution:

```
import java.util.*;
public class Exercise57 {
    public static void main(String[] args){
        Scanner in = new Scanner(System.in);
        System.out.print("Input an integer: ");
        int x = in.nextInt();

        System.out.println(result(x));
    }
    public static int result(int num) {
        int ctr = 0;
        for(int i=1; i<=(int)Math.sqrt(num); i++) {
            if(num%i==0 && i*i!=num) {
                ctr+=2;
            } else if (i*i==num) {
                ctr++;
            }
        }
        return ctr;
    }
}
```

58. Write a Java program to capitalize the first letter of each word in a sentence.

Sample Output:

Input a Sentence: the quick brown fox jumps over the lazy dog.
The Quick Brown Fox Jumps Over The Lazy Dog.

Solution:

```
import java.util.*;
public class Exercise58 {
    public static void main(String[] args){
        Scanner in = new Scanner(System.in);
        System.out.print("Input a Sentence: ");
        String line = in.nextLine();
        String upper_case_line = "";
        Scanner lineScan = new Scanner(line);
        while(lineScan.hasNext()) {
            String word = lineScan.next();
            upper_case_line += Character.toUpperCase(word.charAt(0))
+ word.substring(1) + " ";
        }
        System.out.println(upper_case_line.trim());
    }
}
```

59. Write a Java program to convert a given string into lowercase.

Sample Output:

Input a String: THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG.
the quick brown fox jumps over the lazy dog.

Solution:

```
import java.util.*;  
  
public class Exercise59 {  
    public static void main(String[] args){  
        Scanner in = new Scanner(System.in);  
        System.out.print("Input a String: ");  
        String line = in.nextLine();  
        line = line.toLowerCase();  
        System.out.println(line);  
    }  
}
```

60. Write a Java program to find the penultimate (next to last) word of a sentence.

Sample Output:

Input a String: The quick brown fox jumps over the lazy dog.
Penultimate word: lazy.

Solution:

```
import java.util.*;
public class Exercise60 {
    public static void main(String[] args){
        Scanner in = new Scanner(System.in);
        System.out.print("Input a Sentence: ");
        String line = in.nextLine();
        String[] words = line.split("[ ]+");
        System.out.println("Penultimate word: "+words[words.length - 2]);
    }
}
```

61. Write a Java program to reverse a word

Sample Output:

```
Input a word: dsaf
Reverse word: fasd
```

Solution:

```
import java.util.*;

public class Exercise61 {

    public static void main(String[] args){
        Scanner in = new Scanner(System.in);
        System.out.print("\nInput a word: ");

        String word = in.nextLine();
        word = word.trim();
        String result = "";

        char[] ch=word.toCharArray();
        for (int i = ch.length - 1; i >= 0; i--) {
            result += ch[i];
        }

        System.out.println("Reverse word: "+result.trim());
    }
}
```

62. Write a Java program that accepts three integer values and return true if one of them is 20 or more and less than the subtractions of others.

Sample Output:

```
Input the first number : 15
Input the second number: 20
Input the third number : 25
False
```

Solution:

```
import java.util.*;
public class Exercise62 {
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        System.out.print("Input the first number : ");
        int x = in.nextInt();
        System.out.print("Input the second number: ");
        int y = in.nextInt();
        System.out.print("Input the third number : ");
        int z = in.nextInt();
        System.out.println((Math.abs(x - y) >= 20 || Math.abs(y - z)
            >= 20 || Math.abs(z - x) >= 20));
    }
}
```

63. Write a Java program that accepts two integer values from the user and return the larger values. However if the two values are the same, return 0 and return the smaller value if the two values have the same remainder when divided by 6.

Sample Output:

Test Data:

```
Input the first number : 12
Input the second number: 13
Result: 13
Input the first number : 12
Input the second number: 12
Result: 0
Input the first number : 6
Input the second number: 18
Result: 6
```

Solution:

```
import java.util.*;
public class Exercise63 {
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        System.out.print("Input the first number : ");
        int a = in.nextInt();
        System.out.print("Input the second number: ");
        int b = in.nextInt();
        System.out.println("Result: "+result(a, b));
    }
    public static int result(int x, int y)
    {
        if(x == y)
            return 0;
        if(x % 6 == y % 6)
            return (x < y) ? x : y;
        return (x > y) ? x : y;
    }
}
```

64. Write a Java program that accepts two integer values between 25 to 75 and return true if there is a common digit in both numbers.

Sample Output:

```
Input the first number : 35
Input the second number: 45
Result: true
```

Solution:

```
import java.util.*;
public class Exercise64 {
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        System.out.print("Input the first number : ");
        int a = in.nextInt();
        System.out.print("Input the second number: ");
        int b = in.nextInt();
        System.out.println("Result: "+common_digit(a, b));
    }

    public static boolean common_digit(int p, int q)
    {
        if (p<25 || q>75)
            return false;
        int x = p % 10;
        int y = q % 10;
        p /= 10;
        q /= 10;
        return (p == q || p == y || x == q || x == y);
    }
}
```

65. Write a Java program to calculate the modules of two numbers without using any inbuilt modulus operator.

Sample Output:

```
Input the first number : 19
Input the second number: 7
5
```

Solution:

```
import java.util.*;
public class Exercise65 {
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        System.out.print("Input the first number : ");
        int a = in.nextInt();
        System.out.print("Input the second number: ");
        int b = in.nextInt();
        int divided = a / b;
        int result = a - (divided * b);
        System.out.println(result);
    }
}
```

66. Write a Java program to compute the sum of the first 100 prime numbers.

Sample Output:

Sum of the first 100 prime numbers: 24133

Solution:

```
import java.util.*;
public class Exercise66 {
    public static void main(String[] args)
    {
        int sum = 1;
        int ctr = 0;
        int n = 0;

        while (ctr < 100) {
            n++;
            if (n % 2 != 0) {
                // check if the number is even
                if (is_Prime(n)) {
                    sum += n;
                }
            }
            ctr++;
        }
        System.out.println("\nSum of the prime numbers till 100:
"+sum);
    }

    public static boolean is_Prime(int n) {
        for (int i = 3; i * i <= n; i+= 2) {
            if (n % i == 0) {
                return false;
            }
        }
        return true;
    }
}
```

67. Write a Java program to insert a word in the middle of the another string.

Insert "Tutorial" in the middle of "Python 3.0", so result will be Python Tutorial 3.0

Sample Output:

Python Tutorial 3.0

Solution:

```
import java.lang.*;
public class Exercise67 {
    public static void main(String[] args)
    {
        String main_string = "Python 3.0";
        String word = "Tutorial";
        System.out.println(main_string.substring(0, 7) + word +
main_string.substring(6));
    }
}
```

68. Write a Java program to create a new string of 4 copies of the last 3 characters of the original string. The length of the original string must be 3 and above.

Sample Output:

3.03.03.03.0

Solution:

```
import java.lang.*;
public class Exercise68 {
public static void main(String[] args)
{
    String main_string = "Python 3.0";

    String last_three_chars =
main_string.substring(main_string.length() - 3);
    System.out.println(last_three_chars + last_three_chars +
last_three_chars + last_three_chars);
}
```

69. Write a Java program to extract the first half of a string of even length.

Test Data: Python

Sample Output:

Pyt

Solution:

```
import java.lang.*;
public class Exercise69 {
public static void main(String[] args)
{
    String main_string = "Python";
    System.out.println(main_string.substring(0,
main_string.length()/2));
}
```

70. Write a Java program to create a string in the form short_string + long_string + short_string from two strings. The strings must not have the same length.

Test Data: Str1 = Python

Str2 = Tutorial

Sample Output:

PythonTutorialPython

Solution:

```
import java.lang.*;
public class Exercise70 {
public static void main(String[] args)
{
    String str1 = "Python";
    String str2 = "Tutorial";

    if(str1.length() >= str2.length())
        System.out.println( str2+str1+str2);
    else
        System.out.println(str1+str2+str1);
}
}
```

71. Write a Java program to create the concatenation of the two strings except removing the first character of each string. The length of the strings must be 1 and above.

Test Data: Str1 = Python

Str2 = Tutorial

Sample Output:

Ythonutorial

Solution:

```
import java.lang.*;
public class Exercise71 {
    public static void main(String[] args)
    {
        String str1 = "Python";
        String str2 = "Tutorial";

        System.out.println(str1.substring(1) + str2.substring(1));
    }
}
```


72. Write a Java program to create a new string taking first three characters from a given string. If the length of the given string is less than 3 use "#" as substitute characters.

Test Data: Str1 = " "

Sample Output:

#

Solution:

```
import java.lang.*;
public class Exercise72 {
public static void main(String[] args)
{
    String str1 = "";
    int len = str1.length();
    if(len >= 3)
        System.out.println( str1.substring(0, 3));
    else if(len == 1)
        System.out.println( (str1.charAt(0)+"##"));
    else
        System.out.println("###");
}
}
```

73. Write a Java program to create a new string taking first and last characters from two given strings. If the length of either string is 0 use "#" for missing character.

Test Data: str1 = "Python"

str2 = " "

Sample Output:

P#

Solution:

```
import java.lang.*;
public class Exercise73 {
public static void main(String[] args)
{
    String str1 = "Python";
    String str2 = "";

    int length2 = str2.length();
    String result = "";
    result += (str1.length() >= 1) ? str1.charAt(0) : '#';
    result += (length2 >= 1) ? str2.charAt(length2-1) : '#';
    System.out.println(result);
}
```

74. Write a Java program to test if 10 appears as either the first or last element of an array of integers. The length of the array must be greater than or equal to 2.

Sample Output:

Test Data: array = 10, -20, 0, 30, 40, 60, 10

true

Solution:

```
import java.lang.*;
public class Exercise74 {
    public static void main(String[] args)
    {
        int[] num_array = {10, -20, 0, 30, 40, 60, 10};
        System.out.println((num_array[0] == 10 ||
num_array[num_array.length-1] == 10));
    }
}
```

75. Write a Java program to test if the first and the last element of an array of integers are same. The length of the array must be greater than or equal to 2.

Test Data: array = 50, -20, 0, 30, 40, 60, 10

Sample Output:

false

Solution:

```
import java.lang.*;
public class Exercise75 {
public static void main(String[] args)
{
    int[] num_array = {50, -20, 0, 30, 40, 60, 10};

    System.out.println (num_array.length >= 2 && num_array[0] ==
num_array[num_array.length-1]);
}
}
```

76. Write a Java program to test if the first and the last element of two array of integers are same. The length of the array must be greater than or equal to 2.

Test Data: array1 = 50, -20, 0, 30, 40, 60, 12

array2 = 45, 20, 10, 20, 30, 50, 11

Sample Output:

false

Solution:

```
public class Exercise76 {
    public static void main(String[] args)
    {
        int[] num_array1 = {50, -20, 0, 30, 40, 60, 12};
        int[] num_array2 = {45, 20, 10, 20, 30, 50, 11};

        System.out.println(num_array1[0] == num_array2[0] ||
num_array1[num_array1.length-1] == num_array2[num_array2.length-1]);
    }
}
```

77. Write a Java program to create a new array of length 2 from two arrays of integers with three elements and the new array will contain the first and last elements from the two arrays.

Test Data: array1 = 50, -20, 0

array2 = 5, -50, 10

Sample Output:

Array1: [50, -20, 0]

Array2: [5, -50, 10]

New Array: [50, 10]

Solution:

```
import java.util.Arrays;
public class Exercise77 {
    public static void main(String[] args)
    {
        int[] array1 = {50, -20, 0};
        int[] array2 = {5, -50, 10};
        System.out.println("Array1: "+Arrays.toString(array1));
        System.out.println("Array2: "+Arrays.toString(array2));
        int[] array_new = {array1[0], array2[2]};
        System.out.println("New Array: "+Arrays.toString(array_new));
    }
}
```

78. Write a Java program to test that a given array of integers of length 2 contains a 4 or a 7.

Sample Output:

```
Original Array: [5, 7]
true
```

Solution:

```
import java.util.Arrays;
public class Exercise78 {
    public static void main(String[] args)
    {
        int[] array_nums = {5, 7};
        System.out.println("Original Array:
"+Arrays.toString(array_nums));
        if(array_nums[0] == 4 || array_nums[0] == 7)
            System.out.println("True");
        else
            System.out.println(array_nums[1] == 4 || array_nums[1] == 7);
    }
}
```

79. Write a Java program to rotate an array (length 3) of integers in left direction.

Sample Output:

```
Original Array: [20, 30, 40]
Rotated Array: [30, 40, 20]
```

Solution:

```
import java.util.Arrays;
public class Exercise79 {
    public static void main(String[] args)
    {
        int[] array_nums = {20, 30, 40};
        System.out.println("Original Array:
"+Arrays.toString(array_nums));
        int[] new_array_nums = {array_nums[1], array_nums[2],
array_nums[0]};
        System.out.println("Rotated Array:
"+Arrays.toString(new_array_nums));
    }
}
```

80. Write a Java program to get the larger value between first and last element of an array (length 3) of integers .

Sample Output:

```
Original Array: [20, 30, 40]
Larger value between first and last element: 40
```

Solution:

```
import java.util.Arrays;
public class Exercise80 {
    public static void main(String[] args)
    {
        int[] array_nums = {20, 30, 40};
        System.out.println("Original Array:
"+Arrays.toString(array_nums));
        int max_val = array_nums[0];
        if(array_nums[2] >= max_val)
            max_val = array_nums[2];
        System.out.println("Larger value between first and last element:
"+max_val);
    }
}
```

81. Write a Java program to swap the first and last elements of an array (length must be at least 1) and create a new array.

Sample Output:

```
Original Array: [20, 30, 40]
New array after swaping the first and last elements: [40, 30,
20]
```

Solution:

```
import java.util.Arrays;
public class Exercise81 {
    public static void main(String[] args)
    {
        int[] array_nums = {20, 30, 40};
        System.out.println("Original Array:
"+Arrays.toString(array_nums));
        int x = array_nums[0];
        array_nums[0] = array_nums[array_nums.length-1];
        array_nums[array_nums.length-1] = x;
        System.out.println("New array after swaping the first and last
elements: "+Arrays.toString(array_nums));
    }
}
```

82. Write a Java program to find the largest element between first, last, and middle values from an array of integers (even length).

Sample Output:

```
Original Array: [20, 30, 40, 50, 67]
Largest element between first, last, and middle values: 67
```

Solution:

```
import java.util.Arrays;
public class Exercise82 {
    public static void main(String[] args)
    {
        int[] array_nums = {20, 30, 40, 50, 67};
        System.out.println("Original Array:
"+Arrays.toString(array_nums));
        int max_val = array_nums[0];
        if(max_val <= array_nums[array_nums.length-1])
            max_val = array_nums[array_nums.length-1];
        if(max_val <= array_nums[array_nums.length/2])
            max_val = array_nums[array_nums.length/2];
        System.out.println("Largest element between first, last, and
middle values: "+max_val);
    }
}
```

83. Write a Java program to multiply corresponding elements of two arrays of integers.

Sample Output:

Array1: [1, 3, -5, 4]

Array2: [1, 4, -5, -2]

Result: 1 12 25 -8

Solution:

```
import java.util.*;
public class Exercise83 {
    public static void main(String[] args){
        String result = "";
        int[] left_array = {1, 3, -5, 4};
        int[] right_array = {1, 4, -5, -2};
        System.out.println("\nArray1: "+Arrays.toString(left_array));
        System.out.println("\nArray2: "+Arrays.toString(right_array));
        for (int i = 0; i < left_array.length; i++) {
            int num1 = left_array[i];
            int num2 = right_array[i];
            result += Integer.toString(num1 * num2) + " ";
        }
        System.out.println("\nResult: "+result);
    }
}
```

84. Write a Java program to take the last three characters from a given string and add the three characters at both the front and back of the string. String length must be greater than three and more.

Test data: "Python" will be "honPythonhon"

Sample Output:

honPythonhon

Solution:

```
import java.util.*;
import java.io.*;
public class Exercise84 {
    public static void main(String[] args)
    {
        String string1 = "Python";
        int slength = 3;
        if (slength > string1.length()) {
            slength = string1.length();
        }

        String subpart = string1.substring(string1.length()-3);
        System.out.println(subpart + string1 + subpart);
    }
}
```

85. Write a Java program to check if a string starts with a specified word.

Sample Data: string1 = "Hello how are you?"

Sample Output:

```
true
```

Solution:

```
import java.util.*;
import java.io.*;
public class Exercise85 {
    public static void main(String[] args)
    {
        String string1 = "Hello how are you?";
        System.out.println(string1.startsWith("Hello"));
    }
}
```

86. Write a Java program start with an integer n, divide n by 2 if n is even or multiply by 3 and add 1 if n is odd, repeat the process until n = 1.

Solution:

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        int ctr = 0;
        Scanner in = new Scanner(System.in);
        int n = in.nextInt();
        while (n != 1) {
            System.out.println(n);
            if (n % 2 == 0) {
                n = n / 2;
                ctr += 1;
            }
            else {
                n = (3 * n + 1) / 2;
                ctr += 1;
            }
        }
        System.out.println(ctr);
        in.close();
    }
}
```

87. Write a Java program than read an integer and calculate the sum of its digits and write the number of each digit of the sum in English.

Solution:

```
import java.io.*;

public class Main {
    public static void main(String[] args) {
        BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));

        try {
            int sum = 0;
            String str = br.readLine();
            char[] numStr = str.toCharArray();
            for (int i = 0; i < numStr.length; i++) {
                sum += numStr[i] - '0';
            }
            System.out.println("Original Number: "+str);
            print_number(sum);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public static void print_number(int n) {
        int x; int y; int z;
        String[] number =
{"zero","one","two","three","four","five","six","seven","eight","nine"};
        System.out.println("Sum of the digits of the said number:
"+n);
        if (n < 10) {
            System.out.println(number[n]);
        }
        else if (n < 100) {
            x = n / 10;
            y = n - x *10;
            System.out.println("In English: "+number[x] + " " +
number[y]);
        }
        else {
            x = n / 100;
            y = (n - x * 100) / 10;
        }
    }
}
```

```
        z = n - x * 100 - y * 10;
        System.out.println("In English: "+number[x] + " " +
number[y] + " " + number[z]);
    }
}
```

88. Write a Java program to get the current system environment and system properties.

Solution:

```
import java.lang.*;
public class Exercise88 {
public static void main(String[] args)
{
    System.out.println("\nCurrent system environment:");
    System.out.println(System.getenv());
    System.out.println("\n\nCurrent system properties:");
    System.out.println(System.getProperties());
}
}
```

89. Write a Java program to check whether a security manager has already been established for the current application or not.

Solution:

```
import java.lang.*;
public class Exercise89 {
    public static void main(String[] args)
    {
        System.out.println("System security interface:");
        System.out.println(System.getSecurityManager());
    }
}
```

90. Write a Java program to get the value of the environment variable PATH, TEMP, USERNAME.

Solution:

```
import java.lang.*;

public class Exercise90 {

    public static void main(String[] args)

    {

        // gets the value of the specified environment variable "PATH"

        System.out.println("\nEnvironment variable PATH: ");

        System.out.println(System.getenv("PATH"));



        // gets the value of the specified environment variable "TEMP"

        System.out.println("\nEnvironment variable TEMP: ");

        System.out.println(System.getenv("TEMP"));



        // gets the value of the specified environment variable "USERNAME"

        System.out.println("\nEnvironment variable USERNAME: ");

        System.out.println(System.getenv("USERNAME"));

    }

}
```

91. Write a Java program to measure how long some code takes to execute in nanoseconds.

Sample Output:

```
The first 10 natural numbers are:  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
Estimated time (in nanoseconds) to get the first 10 natural  
numbers: 29  
1045
```

Solution:

```
import java.lang.*;  
public class Exercise91 {  
    public static void main(String[] args)  
    {  
        long startTime = System.nanoTime();  
        // Sample program  
        // Display the first 10 natural numbers.  
        int i;  
        System.out.println ("The first 10 natural numbers are:\n");  
        for (i=1;i<=10;i++)  
        {  
            System.out.println (i);  
        }  
        long estimatedTime = System.nanoTime() - startTime;  
        System.out.println("Estimated time (in nanoseconds) to get the  
first 10 natural numbers: "+estimatedTime);  
    }  
}
```

92. Write a Java program to count the number of even and odd elements in a given array of integers.

Sample Output:

```
Original Array: [5, 7, 2, 4, 9]
Number of even elements in the array: 2
Number of odd elements in the array: 3
```

Solution:

```
import java.util.*;
public class Exercise92 {
    public static void main(String[] args)
    {
        int[] nums = {5, 7, 2, 4, 9};
        int ctr_even = 0, ctr_odd = 0;
        System.out.println("Original Array: "+Arrays.toString(nums));

        for(int i = 0; i < nums.length; i++) {
            if(nums[i] % 2 == 0)
            {
                ctr_even++;
            }
            else
                ctr_odd++;
        }
        System.out.printf("\nNumber of even elements in the array:
%d",ctr_even);
        System.out.printf("\nNumber of odd elements in the array:
%d",ctr_odd);
        System.out.printf("\n");
    }
}
```

93. Write a Java program to test if an array of integers contains an element 10 next to 10 or an element 20 next to 20, but not both.

Sample Output:

```
Original Array: [10, 10, 2, 4, 20, 20]
false
```

Solution:

```
import java.util.*;
public class Exercise93 {
    public static void main(String[] args)
    {
        //int[] nums = {10, 10, 2, 4, 9};
        int[] nums = {10, 10, 2, 4, 20, 20};
        int ctr_even = 0, ctr_odd = 0;
        System.out.println("Original Array: "+Arrays.toString(nums));

        boolean found1010 = false;
        boolean found2020 = false;

        for(int i = 0; i < nums.length - 1; i++)
        {
            if(nums[i] == 10 && nums[i+1] == 10)
                found1010 = true;

            if(nums[i] == 20 && nums[i+1] == 20)
                found2020 = true;
        }

        System.out.printf( String.valueOf(found1010 != found2020));
        System.out.printf("\n");
    }
}
```

94. Write a Java program to rearrange all the elements of an given array of integers so that all the odd numbers come before all the even numbers.

Sample Output:

```
Original Array: [1, 7, 8, 5, 7, 13, 0, 2, 4, 9]
New Array: [7, 5, 7, 13, 9, 1, 0, 2, 4, 8]
```

Solution:

```
import java.util.*;
import java.io.*;
public class Exercise94 {
    public static void main(String[] args)
    {
        int[] array_nums = {1, 7, 8, 5, 7, 13, 0, 2, 4, 9};

        int i = 0;
        System.out.println("Original Array:
"+Arrays.toString(array_nums));
        while(i < array_nums.length && array_nums[i] % 2 == 0)
            i++;

        for(int j = i + 1; j < array_nums.length; j++) {
            if(array_nums[j] % 2 != 0) {
                int temp = array_nums[i];
                array_nums[i] = array_nums[j];
                array_nums[j] = temp;
                i++;
            }
        }

        System.out.println("New Array: "+Arrays.toString(array_nums));
    }
}
```

95. Write a Java program to create an array (length $\neq 0$) of string values. The elements will contain "0", "1", "2" ... through ... n-1.

Sample Output:

```
New Array: [0, 1, 2, 3, 4]
```

Solution:

```
import java.util.*;
import java.io.*;
public class Exercise95 {
    public static void main(String[] args)
    {
        int n= 5;
        String[] arr_string = new String[n];

        for(int i = 0; i < n; i++)
            arr_string[i] = String.valueOf(i);
        System.out.println("New Array: "+Arrays.toString(arr_string));
    }
}
```

96. Write a Java program to check if there is a 10 in a given array of integers with a 20 somewhere later in the array.

Sample Output:

```
true
```

Solution:

```
import java.util.*;
import java.io.*;
public class Exercise96 {
public static void main(String[] args)
{
    int[] array_nums = {10, 70, 80, 50, 20, 13, 50};
    boolean testd = false;
    int result=0;
    int x = 10;
    int y = 20;

    for(int i = 0; i < array_nums.length; i++)
        if(array_nums[i] == x)
            testd = true;

        if(testd && array_nums[i] == y)
        {
            System.out.printf( String.valueOf(true));
            result = 1
        }
    }
    if (result==0)
    {
        System.out.printf( String.valueOf(false));
    }
    System.out.printf("\n");
}
}
```

97. Write a Java program to check if an array of integers contains a specified number next to each other or there are two same specified numbers separated by one element.

Sample Output:

```
true
```

Solution:

```
import java.util.*;
import java.io.*;
public class Exercise97 {
    public static void main(String[] args)
    {
        int[] array_nums = {10, 20, 10, 50, 20, 13, 50};
        //int[] array_nums = {10, 10, 50, 50, 20, 13, 50};
        boolean testd = false;
        int result=0;
        int x = 10;

        for(int i = 0; i < array_nums.length - 1; i++) {
            if(array_nums[i] == x && array_nums[i+1] == x)
            {
                System.out.printf( String.valueOf(true));
                result = 1 ;
            }

            if(i <= array_nums.length - 3 && array_nums[i] == x &&
array_nums[i+2] == x)
            {
                System.out.printf( String.valueOf(true));
                result = 1 ;
            }
        }

        if (result==0)
        {
            System.out.printf( String.valueOf(false));
        }
        System.out.printf("\n");
    }
}
```

98. Write a Java program to check if the value 20 appears three times and no 20's are next to each other in an given array of integers.

Sample Output:

```
true
```

Solution:

```
import java.util.*;
import java.io.*;
public class Exercise98 {
    public static void main(String[] args)
    {
        int[] array_nums = {10, 20, 10, 20, 40, 13, 20};

        int count = 0, result =0;

        if(array_nums.length >= 1 && array_nums[0] == 20)
            count++;

        for(int i = 1; i < array_nums.length; i++) {
            if(array_nums[i - 1] == 20 && array_nums[i] == 20)
            {
                System.out.printf( String.valueOf(false));
                result = 1;
            }
            if(array_nums[i] == 20)
                count++;
        }

        if (result==0)
        {
            System.out.printf( String.valueOf(count == 3));
        }
        System.out.printf("\n");
    }
}
```

99. Write a Java program to check if a specified number appears in every pair of adjacent element of a given array of integers.

Sample Output:

```
true
```

Solution:

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

public class Exercise99 {
    public static void main(String[] args) {
        int[] array_nums = {10, 20, 10, 20, 40, 20, 50};
        int result = 0;
        int x = 20;
        for(int i = 0; i < array_nums.length - 1; i++) {
            if(array_nums[i] != x && array_nums[i + 1] != x)
                {
                    result = 1;
                }
        }
        if (result==0)
        {
            System.out.printf( String.valueOf(true));
        }
        else
        {
            System.out.printf( String.valueOf(false));
        }
    }
}
```

```
 }  
 }
```

100. Write a Java program to count the two elements of two given arrays of integers with same length, differ by 1 or less.

Sample Output:

```
Array1: [10, 11, 10, 20, 43, 20, 50]
Array2: [10, 13, 11, 20, 44, 30, 50]
Number of elements: 2
```

Solution:

```
import java.util.*;
import java.io.*;
public class Exercise100 {
    public static void main(String[] args)
    {
        int[] array_nums1 = {10, 11, 10, 20, 43, 20, 50};
        int[] array_nums2 = {10, 13, 11, 20, 44, 30, 50};
        System.out.println("Array1: "+Arrays.toString(array_nums1));
        System.out.println("Array2: "+Arrays.toString(array_nums2));
        int ctr = 0;

        for(int i = 0; i < array_nums1.length; i++) {
            if(Math.abs(array_nums1[i] - array_nums2[i]) <= 1 &&
array_nums1[i] != array_nums2[i])
                ctr++;
        }

        System.out.printf("Number of elements: "+ctr);
        System.out.printf("\n");
    }
}
```

Sum of Two Numbers

Write a method that sums two numbers.

Solution:

```
public Integer sum(Integer i, Integer j) {  
    return i + j;  
}
```

```
public class Sum {  
  
    public static void main(String [] args){  
        Sum obj = new Sum();  
        int answer = obj.sum(1,2);  
        System.out.println("The sum of 2 numbers is " + answer);  
    }  
  
    public Integer sum(Integer i, Integer j) {  
        return i + j;  
    }  
}
```

Sum Multiples of Three and Five

Given a number n, write a method that sums all multiples of three and five up to n (inclusive).

Solution:

```
public Integer sum(Integer n) {  
    Integer sum = 0;  
    for (int i = 1; i <= n; i++) {  
        if (i % 3 == 0 || i % 5 == 0) {  
            sum += i;  
        }  
    }  
    return sum;  
}
```

```
public class Sum {  
  
    public static void main(String [] args){  
        Sum obj = new Sum();  
        int answer = obj.sum(21);  
        System.out.println(answer);  
    }  
  
    public Integer sum(Integer n) {  
        Integer sum = 0;  
        for (int i = 1; i <= n; i++) {  
            if (i % 3 == 0 || i % 5 == 0) {  
                sum += i;  
            }  
        }  
        return sum;  
    }  
}
```

Factorial

Write a method that calculates the factorial of a given number.

Factorial is the product of all positive integers less than or equal to n. For example, factorial(4) = 4x3x2x1 = 24.

TIP: To make it more interesting, try to do it recursively.

Solution:

```
public Integer factorial(Integer n) {  
    int factorial = n;  
    for (int j = n - 1; j > 0; j--) {  
        factorial = factorial * j;  
    }  
    return factorial;  
}
```

Linear Search

Write a method that returns the index of the first occurrence of given integer in a list.

Assume that the index of the first element in the list is zero.

If the number does not exist return -1.

Solution:

```
public int search(Integer n, Integer[] list) {  
    int index = -1;  
    for (int i = 0; i < list.length; i++) {  
        if (list[i].equals(n)) {  
            index = i;  
            break;  
        }  
    }  
    return index;  
}
```

Reverse String

Write a method that reverses a string.

For example, 'java interview' becomes 'weivretni avaj'.

Solution:

```
public String reverse(String s) {  
    String result = "";  
    for (int i = 0; i < s.length(); i++) {  
        result += s.charAt(s.length() - i - 1);  
    }  
    return result;  
}
```

Find Maximum

Write a method that returns the largest integer in the list.

You can assume that the list has at least one element.

Solution:

```
public Integer maximum(Integer[] list) {  
    int maximum = list[0];  
    for (int i : list) {  
        if (i > maximum) {  
            maximum = i;  
        }  
    }  
    return maximum;  
}
```

Average Value

Write a method that returns the average of a list of integers.

Solution:

```
public Double average(List<Integer> list) {  
    return list.stream()  
        .mapToInt(i -> i)  
        .average()  
        .getAsDouble();  
}
```

Convert to Upper Case

Write a method that converts all strings in a list to their upper case.

Solution:

```
public List<String> upperCase(List<String> list) {  
    return list.stream()  
        .map(String::toUpperCase)  
        .collect(Collectors.toList());  
}
```

Nth Odd Element

Write a method that returns the nth element of a list. If the index of the element exceeds the list size, then return -1.

Solution:

```
public Integer getElement(List<Integer> list, Integer n) {  
    int elementIndex = 2 * (n-1);  
    return elementIndex > list.size() - 1 ? -1 : list.get(elementIndex);  
}
```

Number Of Tree Nodes

Write a method that returns the number of nodes in a binary tree.
TreeNode API methods: node.left() and node.right().

Solution:

```
public Integer getNumberOfNodes(TreeNode node) {  
    if (node == null) {  
        return 0;  
    }  
  
    return 1 + getNumberOfNodes(node.left()) +  
        getNumberOfNodes(node.right());  
}
```

Count Nodes in List

Count the number of nodes in a list.

ListNode API methods: node.next().

Solution:

```
public Integer count(ListNode node) {  
    int count = 0;  
    ListNode current = node;  
    while (current != null) {  
        count++;  
        current = current.next();  
    }  
    return count;  
}
```

Fizz Buzz

Write a method that returns 'Fizz' for multiples of three and 'Buzz' for the multiples of five.

For numbers which are multiples of both three and five return 'FizzBuzz'.

For numbers that are neither, return the input number.

Solution:

```
public String fizzBuzz(Integer i) {  
    String result = "";  
    if (i % 3 == 0) {  
        result += "Fizz";  
    }  
    if (i % 5 == 0) {  
        result += "Buzz";  
    }  
    if (result.equals("")) {  
        result = i.toString();  
    }  
    return result;  
}
```

Prime Number

A prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself. Write a method that checks if a number is a prime number.

Solution:

```
public Boolean isPrime(Integer n) {  
    boolean isPrime = true;  
    for (int i = n - 1; i > 1; i--) {  
        if (n % i == 0) {  
            isPrime = false;  
            break;  
        }  
    }  
    return isPrime;  
}
```

Fibonacci Number

Write a method that returns the nth element of the Fibonacci Sequence

The Fibonacci Sequence is the series of numbers: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34,...

The next number is found by adding up the two numbers before it.

Assume that indexes start at zero, e.g., fib(0) = 0, fib(1) = 1, ...

Solution:

```
public Integer fibonacci(Integer n) {  
    if (n == 1) {  
        return 1;  
    } else if (n == 0) {  
        return 0;  
    } else {  
        return fibonacci(n - 1) + fibonacci(n - 2);  
    }  
}
```

Palindrome Check

A palindrome is a word which reads the same backward or forward. 'abcba' is a palindrome.

Write a method that detects if a string is a palindrome.

Tip: Use `word.charAt(i)` to get the character at position i.

Solution:

```
public boolean isPalindrome(String word) {  
    int i1 = 0;  
    int i2 = word.length() - 1;  
    while (i2 > i1) {  
        if (word.charAt(i1) != word.charAt(i2)) {  
            return false;  
        }  
        ++i1;  
        --i2;  
    }  
    return true;  
}
```

Even Fibonacci Sum

Write a method that returns the sum of all even Fibonacci numbers. Consider all Fibonacci numbers that are less than or equal to n.

Each new element in the Fibonacci sequence is generated by adding the previous two elements.

1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

Solution:

```
public Integer evenFibonacciSum(Integer n) {  
    int previousFibonacci = 1;  
    int currentFibonacci = 2;  
    int evenFibonacciSum = 0;  
    do {  
        if (currentFibonacci % 2 == 0) {  
            evenFibonacciSum += currentFibonacci;  
        }  
        int newFibonacci = currentFibonacci + previousFibonacci;  
        previousFibonacci = currentFibonacci;  
        currentFibonacci = newFibonacci;  
    } while (currentFibonacci < n);  
    return evenFibonacciSum;  
}
```

Greatest Common Divisor

The Greatest Common Divisor of two positive integers is the largest integer that divides both without remainder. Write a method that returns the Greatest Common Divisor of p and q.

Solution:

```
public Integer gcd(Integer p, Integer q) {  
    if (q == 0) {  
        return p;  
    }  
    return gcd(q, p % q);  
}
```

Package Rice Bags

Provided that you have a given number of small rice bags (1 kilo each) and big rice bags (5 kilos each), write a method that returns true if it is possible to make a package with goal kilos of rice.

Solution:

```
public Boolean packageRice(Integer big, Integer small, Integer goal) {  
    boolean result = false;  
    if (big * 5 >= goal) {  
        if (small >= goal % 5)  
            result = true;  
    } else {  
        if (small >= goal - big * 5) {  
            result = true;  
        }  
    }  
    return result;  
}
```

Filter Strings

Given a list of strings, write a method that returns a list of all strings that start with the letter 'a' (lower case) and have exactly 3 letters. TIP: Use Java 8 Lambdas and Streams API's.

Solution:

```
public List<String> search(List<String> list) {  
    return list.stream()  
        .filter(s -> s.startsWith("a"))  
        .filter(s -> s.length() == 3)  
        .collect(Collectors.toList());  
}
```

Comma Separated

Write a method that returns a comma-separated string based on a given list of integers. Each element should be preceded by the letter 'e' if the number is even, and preceded by the letter 'o' if the number is odd. For example, if the input list is (3,44), the output should be 'o3,e44'.

Solution:

```
public String getString(List<Integer> list) {  
    return list.stream()  
        .map(i -> i % 2 == 0 ? "e" + i : "o" + i)  
        .collect(joining(","));  
}
```

Filter Strings

Given a list of strings, write a method that returns a list of all strings that start with the letter 'a' (lower case) and have exactly 3 letters. TIP: Use Java 8 Lambdas and Streams API's.

Solution:

```
public List<String> search(List<String> list) {  
    return list.stream()  
        .filter(s -> s.startsWith("a"))  
        .filter(s -> s.length() == 3)  
        .collect(Collectors.toList());  
}
```

Ceasar Cipher

The Ceasar cipher is a basic encryption technique used by Julius Ceasar to securely communicate with his generals. Each letter is replaced by another letter N positions down the english alphabet. For example, for a rotation of 5, the letter 'c' would be replaced by an 'h'. In case of a 'z', the alphabet rotates and it is transformed into a 'd'.

Implement a decoder for the Ceasar cipher where N = 5.

TIP: Use code.toCharArray() to get an array of characters.

Solution:

```
public String decode(String code) {  
    String decoded = "";  
    int lastCharValue = 'z';  
    int alphabetLength = 'z' - 'a' + 1;  
    for (char character: code.toCharArray()) {  
        int charNoRotation = character + 5;  
        int charValue = charNoRotation < lastCharValue ? charNoRotation :  
            charNoRotation - alphabetLength;  
        decoded += (char) charValue;  
    }  
    return decoded;  
}
```

Strict Binary Tree Check

A binary tree is strict when all nodes have either two or zero child nodes.

Write a method that checks if a binary tree is strict.

TreeNode API methods: node.left() and node.right().

Solution:

```
public Boolean isStrictTree(TreeNode node) {  
    if (node == null) {  
        return true;  
    }  
    if ((node.left() == null && node.right() != null) || (node.left() != null &&  
        node.right() == null)) {  
        return false;  
    }  
    return isStrictTree(node.left()) && isStrictTree(node.right());  
}
```

Longest Palindrome in Word

A palindrome is a word which reads the same backward or forward.

'abcba' is a palindrome.

Write a method that returns the longest palindrome substring of a given string.

Solution:

```
public String findLongestPalindrome(String word) {  
    String longestPalindrome = "";  
    for (int i = 0; i < word.length(); i++) {  
        String current = findLongestPalindromeInternal(word.substring(i,  
            word.length()));  
        if (longestPalindrome.length() < current.length()) {  
            longestPalindrome = current;  
        }  
    }  
    return longestPalindrome;  
}
```

```
public String findLongestPalindromeInternal(String word) {  
    String longestPalindrome;  
  
    int index = word.length();  
    while (index > 0 && !isPalindrome(word.substring(0, index))) {  
        index--;  
    }  
    longestPalindrome = word.substring(0, index);  
  
    return longestPalindrome;  
}  
  
public boolean isPalindrome(String word) {  
    int i1 = 0;  
    int i2 = word.length() - 1;  
    while (i2 > i1) {  
        if (word.charAt(i1) != word.charAt(i2)) {  
            return false;  
        }  
        ++i1;  
        --i2;  
    }  
    return true;  
}
```

Longest Common Sequence

Given two strings, write a method that finds the longest common sub-sequence.

Solution:

```
public String findLongestCommonSequence(String s1, String s2) {  
    String result = "";  
    for (int length = s1.length(); length > 0; length--) {  
        int startIndex = 0;  
        while (startIndex + length <= s1.length()) {  
            String current = s1.substring(startIndex, startIndex + length);  
            if (s2.contains(current)) {  
                result = current;  
                break;  
            }  
            startIndex++;  
        }  
        if (result.length() != 0) {  
            break;  
        }  
    }  
    return result;  
}
```

Largest Prime Factor

The prime factors of 455 are 5, 7 and 13.

Write a method that calculates the largest prime factor of a given number.

Solution:

```
public Integer largestPrimeFactor(Integer n) {  
    int factor = -1;  
    for (int i = 2; i * i <= n; i++) {  
        if (n == 1) { break; }  
        if (n % i != 0) { continue; }  
        factor = i;  
        while (n % i == 0) {  
            n /= i;  
        }  
    }  
    return n == 1 ? factor : n;  
}
```

Package Rice Bags (part 2)

Provided that you have a given number of small rice bags (1 kilo each) and big rice bags (5 kilos each), write a method that returns the minimum number of small bags necessary to package goal kilos of rice. Return -1 if it is not possible to package the required rice amount with the bags provided.

Solution:

```
public Integer getNumberSmallBags(Integer big, Integer small, Integer goal) {  
    int result = -1;  
    if (big * 5 >= goal) {  
        if (small >= goal % 5)  
            result = goal % 5;  
    } else {  
        if (small >= goal - big * 5) {  
            result = goal - big * 5;  
        }  
    }  
    return result;  
}
```

Perfect Binary Tree Check

A binary tree is perfect when all levels are complete.
Write a method that checks if a binary tree is perfect.
TreeNode API methods: node.left() and node.right().

Solution:

```
public Boolean isPerfectTree(TreeNode node) {  
    int treeDepth = depth(node);  
    return isPerfectTree(node, 1, treeDepth);  
}  
  
private int depth(TreeNode node) {  
    return node.left() != null ? 1 + depth(node.left()) : 1;  
}  
  
private Boolean isPerfectTree(TreeNode node, int depth, int treeDepth) {  
    // check for last level node  
    if (depth == treeDepth && (node.left() == null && node.right() == null)) {  
        return true;  
    }  
    // check for inner levels  
    if ((node.left() != null && node.right() != null)) {  
        return isPerfectTree(node.left(), 1 + depth, treeDepth) &&  
            isPerfectTree(node.right(), 1 + depth, treeDepth);  
    }  
    return false;  
}
```

Complete Binary Tree Check

A binary tree is complete when all levels apart from the last are filled and all leaf nodes in the last level are aligned to the left.

Write a method that checks if a binary tree is complete.

TreeNode API methods: node.left() and node.right().

Solution:

```
public Boolean isCompleteTree(TreeNode node) {  
    int numberNodes = countNodes(node);  
    return isComplete(node, 0, numberNodes);  
}  
  
private Boolean isComplete(TreeNode node, int index, int numberNodes) {  
    if (node == null)  
        return true;  
  
    if (index > numberNodes)  
        return false;  
  
    return isComplete(node.left(), 2 * index + 1, numberNodes) &&  
           isComplete(node.right(), 2 * index + 2, numberNodes);  
}  
  
private Integer countNodes(TreeNode node) {  
    if (node == null) {  
        return 0;  
    }  
    return 1 + countNodes(node.left()) + countNodes(node.right());  
}
```