

## Assignment-01

1. sum of natural numbers

```
public class sum of naturalNumbers
```

{

```
public static void main (String [] args)
```

{

```
int i, num=10; sum=0;
```

```
for (i=1; i<=num; ++i)
```

{

```
sum = sum + i;
```

}

```
System.out.println ("sum of first 10 NaturalNumbers is =
```

```
" + sum);
```

}

}

**Output:**

sum of first 10 Natural Numbers is = 55.

2. Given number it is Palindrome number or not.

```
class palindromeExample {
```

```
public static void main (String args[]) {
```

```
int n, sum=0, temp;
```

```
int r=usn;
```

```
temp = n;
```

```
while (n>0) {
```

```
r = n % 10;
```

```
sum = (sum * 10) + r;
```

```
n = n / 10;
```

}

P. Reddy renuka

192311062

```
if (temp == sum)
    system.out.println("palindrome number");
else
    system.out.println("not palindrome");
}
```

Output:

palindrome number.

3. Given number it is prime number or not.

```
public class main{
    public static void main(string[] args) {
        int num = 29;
        boolean flag = false;
        for(int i=2; i<=num/2; ++i){
            if (num % i == 0){
                flag = true;
                break;
            }
        }
        if (!flag)
            system.out.println(num + " is a prime number.");
        else
            system.out.println(num + " is not a prime number.");
    }
}
```

Input:

29 is a prime number

#### 4 N factorial of a number.

```
class factorial example{  
public static void main(String args[]){  
int i, fact=1;  
int number=5;  
for(i=1; i<=number; i++){  
fact=fact*i;  
}  
}  
}
```

#### Output:

factorial of 5 is: 120.

#### 5 Reverse a number.

```
public class reverse number example{  
public static void main(String[] args){  
int number = 987654, reverse = 0;  
while(number != 0)  
{  
int remainder = number % 10;  
reverse = reverse * 10 + remainder;  
number = number / 10;  
}  
}
```

system.out.println ("The reverse of the given number is: " + reverse)

}

Output 456789

## 6. Armstrong number:

```
static boolean isArmstrong(int n)
{
    int temp, digits = 0, last = 0, sum = 0;
    temp = n;
    while (temp > 0)
    {
        temp = temp / 10;
        digits++;
        sum += Math.pow(last, digits);
        last = temp % 10;
    }
    if (n == sum)
        return true;
    else
        return false;
}

public static void main(String args[])
{
    int num;
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter the limit:");
    num = sc.nextInt();
    System.out.println("Armstrong Number up to " + num + " are:");
    for (int i = 0; i < num; i++)
    {
        if (isArmstrong(i))
            System.out.print(i + " ");
    }
}
```

## Output:

Enter: 999

Armstrong number upto 999 are:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 153, 370, 371, 407

## 7 Happy number

```
def isHappyNumber(num):  
    rem = sum = 0;  
    while (num > 0):  
        rem = num % 10;  
        sum = sum + (rem * rem);  
        num = num // 10;  
    return sum;  
  
num = 82;  
result = sum;  
while (result != 1 and result != 4):  
    result = isHappyNumber(result);  
    if (result == 1):  
        print(str(num) + " is a happy number");  
    elif (result == 4):  
        print(str(num) + " is not a happy number");
```

## Output:

82 is a happy number.

## 8. sum of the digits.

```
public static void main(String args[]){  
    int number, digit, sum = 0;  
    Scanner sc = new Scanner(System.in);  
    System.out.print("Enter the number:");  
    number = sc.nextInt();
```

```

while (number > 0)
{
    digit = number % 10;
    sum = sum + digit;
    number = number / 10;
}
System.out.println ("sum of digits!" + sum);

```

**Output:**

Enter the number: 876  
 sum of digits: 21.

### 9. Number divisible by 5 and 7

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

```

class GFG {
    static int NumGen(int n)
    {
        for (int j = 1; j < n + 1; j++)
        {
            if (j % 5 == 0 || j % 7 == 0)
                System.out.print(j + " ");
        }
    }
}
```

```

public static void main (String args[])
{
    int N = 50;
    NumGen(N);
}
```

**Output:**

5 7 10 14 15 20 21 25 28 30 35 40 42 45 49 50

## 10. perfect number.

```
public static void main (String args[])
{
    long n, sum=0;
    scanner sc=new scanner (System.in);
    System.out.print ("Enter the number :");
    n=sc.nextLong();
    int i=1;
    while (i<=n)
    {
        if (n % i == 0)
        {
            sum=sum+i;
        }
        i++;
    }
    if (sum==n)
    {
        System.out.print (n + " is a perfect number.");
    }
    else
    {
        System.out.println (n + " is not a perfect number.");
    }
}
```

### output:

enter: 28

28 is a perfect number

## 11. GCD and LCM

```
import java.io.*;
class GFG {
    static int gcd (int a, int b) {
        if (b == 0)
            return a;
    }
}
```

```

    else
        return gcd(b % a, b);
}

static int Lcm(int a, int b, int gcdValue)
{
    return Math.abs(a * b) / gcdValue;
}

public static void main(String[] args)
{
    int a = 20, b = 30, gcdValue;
    gcdValue = gcd(a, b);
    System.out.println("GCD = " + gcdValue);
    System.out.println("LCM = " + lcm(a, b, gcdValue));
}

```

### Output:

GCD = 10

LCM = 60.

## 12. Decimal to binary

```

static void decimalToBinary(int n)
{
    int[] binaryNum = new int[1000];
    int i = 0;
    while (n > 0)
    {
        binaryNum[i] = n % 2;
        n = n / 2;
        i++;
    }
    for (int j = i - 1, j >= 0; j--)
        System.out.print(binaryNum[j]);
}

```

```
public static void main (String[] args)
{
    int n=17;
    System.out.println("Decimal - "+n);
    System.out.print("Binary - ");
    decToBinary(n);
}
```

Output:

Decimal - 17

Binary - 10001

### 13. Binary to decimal

```
class GFG {
    static int binaryTodecimal (int n)
    {
        int num = n;
        int dec_value = 0;
        int base = 1;
        int temp = num;
        while (temp > 0) {
            int last_digit = temp % 10;
            temp = temp / 10;
            dec_value += last_digit * base;
            base = base * 2;
        }
        return dec_value;
    }

    public static void main (String[] args)
    {
        int num = 10101001;
```

```
    system.out.println(binaryToDecimal(num));
```

```
}
```

Output:

169.

14. Celsius to Fahrenheit.

```
public class Temperature
```

```
{ public static void main (String args [ ])
```

```
float fahrenheit, celsius;
```

```
celsius = 13;
```

```
fahrenheit = ((celsius * 9) / 5) + 32;
```

```
System.out.println ("Temperature in fahrenheit is :" +
```

```
fahrenheit);
```

25

Output:

Temperature in Fahrenheit is : 55.4

15. sum of odd numbers and even numbers.

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

```
public class GFG {
```

```
    public static void main (String [ ] args)
```

```
{
```

```
int n = 8;
```

```
int evenSum = 0;
```

```
int oddSum = 0;
```

```

for (int i=1; i <= 2*n; i++) {
    if ((i&1) == 0)
        evensum += i;
    else
        oddsum += i;
}
System.out.println("sum of first " + n + " even numbers = " +
    evensum);
System.out.println("sum of first " + n + " odd numbers = " + odd
    sum);
}
}

```

**output:**

sum of first 8 even numbers = 72

sum of first 8 odd numbers = 64.

## 16. Even or odd

```

class evenOdd {
    public static void main(String[] args) {
        int n = 5; evenSum = 0, oddSum = 0;
        for (int i=1; i <= 2*n; i++) {
            if ((i&1) == 0)
                evenSum += i;
            else
                oddSum += i;
        }
        System.out.println("sum of first " + n + " even numbers = "
            + evenSum);
        System.out.println("sum of first " + n + " odd numbers = " +
            oddSum);
    }
}

```

Output:

sum of first 8 even numbers = 72  
sum of first 8 odd numbers = 84.

17.

voting

```
class VotingAge
{
    public static void main(String[] args)
    {
        int age;
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter your age");
        age = sc.nextInt();
        if (age >= 18)
        {
            System.out.println("you are eligible for vote.");
        }
        else
        {
            System.out.println("you are not eligible for vote.");
        }
    }
}
```

Output:

Enter: 19

You are eligible for vote

## vowels and consonants

prints class vowel consonant {

```
public static void main (String [] args) {
```

```
    char ch = 'i';
```

```
    if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u')
```

```
        System.out.println(ch + " is vowel");
```

```
    else  
        System.out.println(ch + " is consonant");
```

y.  
y.

output:

y is vowel.

## 19. Strong number

```
class last {
```

```
public static void main (String [] args)
```

```
{
```

```
    int sum = 0, i, n, fact, number;
```

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

```
    System.out.print ("Enter a number: ");
```

```
    number = scanner.nextInt();
```

```
    int original = number;
```

```
    while (number > 0)
```

```
{
```

```
    number = number % 10;
```

```
    fact = 1;
```

```
    for (i = 1; i <= x; i++)
```

```
{
```

```
    fact = fact * i;
```

```

    }  

    sum = sum + fact;  

    number = number / 10;  

    if (original == sum)  

        system.out.println("Strong Number");  

    else  

        system.out.println("Not Strong Number");
}

```

Output: 145

Square root & cube root.

20.

```

public class RootsCalculator {  

    public static void main (String [] args) {  

        Scanner scanner = new Scanner (System.in);  

        System.out.println ("Enter a number.");  

        double number = scanner.nextDouble();  

        double squareRoot = Math.sqrt (number);  

        double cubicRoot = Math.cbrt (number);  

        System.out.println ("Square root of " + number + " is " +  

            squareRoot);  

        System.out.println ("Cubic root of " + number + " is " +  

            cubicRoot);
    }
}

```

Output:

$$2^2 = 4$$

$$2^3 = 8$$