ASSIGNMENT-1

(10)

10/08/2021

NAME:- N.VISHNUSAI SRI

REGD NO:- 192372039

COURSE CODE :- CSA0985

COURSE :- JAVA PROGRAMMING FOR WEB APPLICATIONS.

Sum of natural numbers.

```
public class SumofNaturalNumbers {
public static void main (String[] args) {
int n=100;
int sum=0;
for (int i=1 ;i<=n ;i++) {
        sum+=i;
 }
System.out.println ("The sum of "+n+" natural numbers is: "+sum);
}
}
```

Output:- 5050.

(2) Check whether given number is prime or not

```
public class primechecker {
public static void main (String[] args) {
int n=25;
Boolean isprime=true;
if (n<=1) {
   isprime=false;
} else {
for (int i=2; i*i<=n; i++){
   if (n%i==0) {
        isprime=false;
        break;
   }
}
}
```

Output:- 25 is not a prime number

```
System.out.println(n + " is " + (isPrime ? "prime" : "not prime"));
    }
}
```

Output: 13, 5, 7, 89

(3) Factorial of a number.

```
class FactorialExample {
    public static void main(String args[]) {
        int i, fact = 1;
        int number = 5;
        for (i = 1; i <= number; i++) {
            fact = fact * i;
        }
        System.out.println("Factorial of " + number + " is: " + fact);
    }
}
```

output: 371 is an Armstrong number

(4) Reverse Number.

```
public class ReverseNumber
{
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);
```

Strong Number

```
public class Armstrong
public class static void
int n = 371;
int temp = n;
int sum = 0;
while (temp
int digit
sum =
ter
}
if (
```

Armstrong Number

```
public class ArmstrongNumber {
public class static void main (String [ ] args ) {
   int n = 371;
    int temp=n;
    int sum=0;
   while (temp≠0) {
    int digit = temp%10;
   sum = sum +digit*digit*digit;
   temp = temp/10;
  }
if (sum==n) {
   System.out.println (num +"is an Armstrong number");
   } else {
   System.out.println (num+"is not an Armstrong number");
  }
 }
}
```

(6) Happy number

```
public class HappyNumber {
 public static int isHappyNumber (int n) {
   int r=0, sum=0;
 While (n>0){
   rem=n%10;
 sum = sum+(rem*rem);
 n=n/10;
 }
 Return sum;
 }
```

output:- 82 is a happy number.

```
public static void main (String[] args) {
    int n= 82
    int result = n;
while (result!=1 && result!=4) {
 result= isHappyNumber (result);
}
if (result==1)
System.out.println (n+" is a happy number");
else if (result==4)
   System.out.println (n+" is not a happy number");
}
}
```

(7) Palindrome.

```
Class palindrome sample{
public static void main (String args[]) {
  int r, sum=0, temp;
int n=454;
temp=n;
while (n>0) {
r=n%10;
sum=(sum*10)+r;
   n=n/10;
}
if (temp==sum)
   System.out.println ("palindrome number");
else
   System.out.println ("not palindrome");
}
}
```

output:
palindrome

Sum of digits

```
public static void main (String args[])
 {
int number, digit, sum=0.
Scanner sc =new Scanner (System.in);
System.out.print ("enter thenumber:");
number = sc.nextInt()
While (number > 0)
{
digits = number %.10
sum = sum + digit;
 number= number /10;
 }
system.out.println ("sum of digits: "+sum);
}
```

output:-
Sum of digits
= 15.

(9) Numbers 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+" ");
 }
 return n;
}
public static void main (String args[])
{
int N=50;
   NumGen(N):
}
```

output:- 35 is divisible by 35 and 7.

(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.next long();
int i= 1;
While (i<=n/2)
{
if (n%i==0)
}
sum = sum+i;
}
i++;
}
if (sum==n)
System.out.print (n+"is a perfect number.")
}
else.
System.out.print (n+"is not a perfect number:");
```

output :-
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 gcd value)
{
```

```
return Math.abs(a*b)/gcd value);
}
public static void main (String [] args) {
    int a= 20 ; b=30, gcd value;
gcd value = get value (a,b)
    System.out.println ("Gcd = "+gcd value)
    System.out.println ("LCM = "+lcm (a,b,gcd value));
}
```

Output:- GCD of 72 and 120 is :24.
LCM of 72 and 120 is : 360.

(12) Decimal to Binary

```
static void dec to binary (int n)
{
int [ ] binary Num= new int [1000];
  int i = 0;
while (n>0)
 {
 binary Num [i]=n%2;
    n:n/2;
    i++;
 }
for(int j = i-1 ;j>= 0;j--)
System.out.print (binary Num[j]);
  }
public static void main (String() args)

{
int n=17;
System.out.println("Decimal-"+n);

System.out.print ("Binary -");

   decToBinary (n);

}
```

Output:-
42 is 101010.

(13) Binary to decimal

```
Class Q1{
 Static int binary to decimal (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.print in (binary to decimal);
  }
```

output:-
42_101010.

Output:
169.

(14) Celsius to farenheit.

```
public class temperature.
{public static void main (String args( ))
{ float farenheit, celsius;
  celsius = 13;
  farenheit = ((celsius*9)15)+32;
System.out.print in ("Temperature in farenheit is =" + fahrenheit)
}}.
```

output:-
oc in F is 55.4

sum of odd numbers and even numbers.

```
import java.io.*;
public class EFG{
public static void main (String [] args)
{
int n=8;
int even sum=0;
int odd sum=0;
for (int i=1; i<=2 ; n=i+1)
    evensum+=i;
    else
    odd sum+=i;
}
System.out.println ("sum of first "+n+" even num: "+ evensum);
System.out.println ("sum of first "+n+" odd numbers ="+odd sum);
}
}
```

Output: Sum of first 8 even numbers = 32
Sum of first 8 odd num = 64

(16) even or odd.

```
class even_odd {
public static void main( String [] args)
{int n=5 ; even sum = 0 ; odd sum=0 ;
for(int i=1 ;i<=2*n;i++) {
IF((iF1==0)
evensum+=i;
else
oddsum+=i.
}
System.out.println ("sum of first "+n+"even numbers = "+even sum);
system.out println ("sum of first "+n+"odd numbers = "+ odd sum);
}}
```

(17) voting:

```
class voting age
  {
  public static void main (String [] args)
    {
    int age;
    if (age >= 18)
  {
  System.out.println ("you are eligible to vote");
  }
  else
    {
    System.out.println ("you are not eligible to vote");
    }
```

output:-
age: 19
"eligible to vote".

(18) vowels and consonants

```
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.print (ch+" "+"is vowel");
else
System.out.print (ch+"is "+"consonant");
  }
  }
```

output : i is vowel.

strong number.

```
class test {
public static void main (String [] args)
 {
int sum = 0; i,r, fact, number;
Scanner (System.in);
System.out.println ("enter a number:");
 number = in.nextInt();
 int original = number;
   while (num > 0)
    {
    r = num % 10;
   fact = 1;
   for (i = 1; i<r; 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");
```

(80) Cube root and square root

```
public class roots calculator{
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 cubic Root=Math.cbrt(number);
System.out.println("Squareroot of"+number+"is:"+squareroot);
System.out.println("cubicroot of"+number+"is:"+cobicroot);
```

output:

square root of number [3] is 9.

cubic root of 3 is 27.