**JAVA 15**

**HACKER RANK SOLUTIONS FOR LOOPING STATEMENTS**

**PROBLEM 1:**

**Harshad number 22**

Write a program to find whether the given number is a Harshad number or not. Note that Harshard number is an integer that is divisible by the sum of its digits.

**Input Format**

Input consists of 1 integer.

**Constraints**

No Constraints

**Output Format**

If the given number is a Harshad Number, display “Harshad Number” or display “Not Harshad Number”.

**Sample Input 0**

11

**Sample Output 0**

Not Harshad Number

**Sample Input 1**

1729

**Sample Output 1**

Harshad Number

**Harshad number 22**

**CODING :**

import java.util.Scanner;

public class HarshadNumber {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int num = sc.nextInt();

int temp = num;

int sum = 0;

while (temp > 0) {

sum += temp % 10;

temp /= 10;

}

if (num % sum == 0) {

System.out.println("Harshad Number");

} else {

System.out.println("Not Harshad Number");

}

sc.close();

}

}

**PROBLEM 2:**

**Abundant number 11**

Write a program to find whether the given number is an Abundant number or not. Note: An abundant number is a number for which the sum of its proper divisors is greater than the number itself. For example, integer 12 is an abundant number. The divisors of 12 are 1, 2, 3, 4 and 6. The sum of divisors of 12 is 16. As 12 < 16, it is an abundant number.

**Input Format**

Input consists of 1 integer.

**Constraints**

No Constraint

**Output Format**

If it is an Abundant number display “Abundant Number” or display “Not Abundant Number”.

**Sample Input 0**

6

**Sample Output 0**

Not Abundant Number

**Sample Input 1**

12

**Sample Output 1**

Abundant Number

**CODING :**

import java.io.\*;

import java.util.\*;

public class Solution {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

int i,sum=0;

int a = s.nextInt();

for(i=1;i<=a/2;i++){

if(a%i==0){

sum+=i;

}

}

if(sum>a){

System.out.println("Abundant Number");

}

else{

System.out.println("Not Abundant Number");

}

}

}

**PROBLEM 3:**

**SUM OF DIGIT 10**

Gunal wants to find the sum of integer digits number.could you please help him to find it and also impelements in program.

**Input Format**

input consists of one integer.

**Constraints**

Given N is more then 2 digit integer.

**Output Format**

Print the sum of digit value.

**Sample Input 0**

1234

**Sample Output 0**

Sum of digit is 10

**Sample Input 1**

12

**Sample Output 1**

Invalid Input

**CODING :**

import java.io.\*;

import java.util.\*;

public class Solution {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

int sum = 0;

int a = s.nextInt();

int temp = a;

if (temp < 100) {

System.out.println("Invalid Input");

} else {

while (temp > 0) {

sum += temp % 10;

temp /= 10;

}

System.out.println("Sum of digit is " + sum);

}

}

}

**PROBLEM 4:**

**Fibonacci series 144**

Write the program to calculate the sum of the Fibonacci series between the range of the two values. The n is the start of the range value and m is the end of the range value. Finally, return the sum of the range value of the Fibonacci Series.

**Input Format**

Input consists of two integers.

**Constraints**

N is greater than equal to one and lesser than equal to 20

M is greater than equal to one and lesser than equal to 20

**Output Format**

The sum of the range value.

If the constraints are more than the input print the statement is "Invalid Input".

**Sample Input 0**

3

5

**Sample Output 0**

The Sum of Fibonacci value is 6.0

**Sample Input 1**

22

23

**Sample Output 1**

Invalid Input

**CODING:**

import java.io.\*;

import java.util.\*;

public class Solution {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

int m = sc.nextInt();

if (n < 1 || n > 20 || m < 1 || m > 20) {

System.out.println("Invalid Input");

return;

}

if (n > m) {

int temp = n;

n = m;

m = temp; }

int sum = 0;

int a = 0, b = 1;

for (int i = 1; i <= m; i++) {

int fib;

if (i == 1) fib = a;

else if (i == 2) fib = b;

else {

fib = a + b;

a = b;

b = fib;

}

if (i >= n) {

sum += fib;

}

}

System.out.println("The Sum of Fibonacci value is " + sum + ".0");

}

}

**PROBLEM 5:**

**Multiplication table 79**

write the program to print the Multiplication table for nth table and till n count.

**Input Format**

consdier one integer input

**Constraints**

1=>n<=9

**Output Format**

Multiplication table

**Sample Input 0**

5

**Sample Output 0**

5 x 1 = 5

5 x 2 = 10

5 x 3 = 15

5 x 4 = 20

5 x 5 = 25

**Sample Input 1**

11

**Sample Output 1**

Invalid Input

**CODING :**

import java.io.\*;

import java.util.\*;

public class Solution {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

int n = s.nextInt();

int i;

if(n>=1&&n<=9){

for(i=1;i<=n;i++){

System.out.println(n+" x "+i+" = "+(i\*n));

}

}

else{

System.out.println("Invalid Input");

}

}

}

**PROBLEM 6:**

**sum of even number 1**

Find the sum of even number with the range

**Input Format**

Consider two integer input n and m

**Constraints**

n and m values greater than 0 and less than 30

**Output Format**

sum of even values

**Sample Input 0**

5

15

**Sample Output 0**

50

**Sample Input 1**

22

10

**Sample Output 1**

Invalid Input

**CODING :**

import java.io.\*;

import java.util.\*;

public class Solution {

public static void main(String[] args) {

Scanner s = new Scanner (System.in);

int a = s.nextInt();

int b = s.nextInt();

int i,sum =0;

if((a>0&&b<30)&&(a<b)){

for(i=a;i<=b;i++){

if(i%2==0){

sum=sum+i;

}

}

System.out.println(sum);

}

else {

System.out.println("Invalid Input");

}

}

}

**PROBLEM 7:**

**Armstrong number or not 13**

write a program to find the given 3- digit number is Armstrong number or not

**Input Format**

Input corresponds to the 3-digit Integer Number

**Constraints**

N>=100 <=999

**Output Format**

Yes or No

**Sample Input 0**

153

**Sample Output 0**

Yes

**Sample Input 1**

1245

**Sample Output 1**

No

**CODING :**

import java.io.\*;

import java.util.\*;

public class Solution {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

int a = s.nextInt();

int sum=0,temp=a,digit=0;

if(a>=100&&a<=999){

while(temp>0){

sum = temp%10;

digit=sum\*sum\*sum;

temp/=10;

}

System.out.println("Yes");

}

else{

System.out.println("No");

}

}

}

**PROBLEM 8:**

**swap 2 digit number**

write a program to read 2 digit number dynamically and swap digits and print the result

**Input Format**

once integer number of 2 digits

**Constraints**

it can be assumed that the input value will be such that there will be no zero as any of the 2 digits

**Output Format**

Calculated numerical output value,not other number message to be printed in the STDOUT.

**Sample Input 0**

21

**Sample Output 0**

12

**Sample Input 1**

123

**Sample Output 1**

21

**Sample Input 2**

14579

**Sample Output 2**

41

**Sample Input 3**

7

**Sample Output 3**

STDOUT

**CODING :**

import java.io.\*;

import java.util.\*;

public class Solution {

public static void main(String[] args) {

Scanner s = new Scanner (System.in);

String a = s.nextLine();

if(a.length()<2){

System.out.println("STDOUT");

}

else{

char c = a.charAt(0);

char b = a.charAt(1);

System.out.print(b);

System.out.print(c);

}

}

}

**PROBLEM 9:**

**Reverse a Number Using Loop 1**

write the program to Reverse a given number Using Looping statement

**Input Format**

get a user input as an integer value

**Constraints**

0< n >1000000000

**Output Format**

Reverse order of a number

**Sample Input 0**

1423

**Sample Output 0**

3241

**Sample Input 1**

-128

**Sample Output 1**

Invalid Input

**Sample Input 2**

987654321

**Sample Output 2**

123456789

CODING :

import java.io.\*;

import java.util.\*;

public class Solution {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

long n = sc.nextLong();

if(n <= 0 || n > 1000000000) {

System.out.println("Invalid Input");

} else {

long rev = 0;

long temp = n;

while(temp > 0) {

long digit = temp % 10;

rev = rev \* 10 + digit;

temp /= 10;

}

System.out.println(rev);

}

}

}

PROBLEM 10:

**Collatz Sequence 7**

The rules for generating Collatz Sequence are:

If n is even: n = n / 2 If n is odd: n = 3n + 1 For example, if the starting number is 5 the sequence is:

5 -> 16 -> 8 -> 4 -> 2 -> 1 It has been proved that for almost all integers, the repeated application of the above rule will result in a sequence that ends at 1.

Given a positive integer, write a program to print this sequence and the number of times this rule needs to be applied in order to reach 1.

**Input Format**

Input consists of a positive integer.

**Constraints**

No constraints

**Output Format**

Print the numbers in the sequence, one per line and finally print the number of times the rule has to be applied in order to reach 1.

**Sample Input 0**

5

**Sample Output 0**

Enter a number

5

16

8

4

2

1

count:5

**Sample Input 1**

1

**Sample Output 1**

Enter a number

1

count:0

CODING :

import java.io.\*;

import java.util.\*;

public class Solution {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

int count = 0;

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

while (n != 1) {

System.out.println(n);

if (n % 2 == 0) {

n = n / 2;

} else {

n = 3 \* n + 1;

}

count++;

}

System.out.println(n);

System.out.println("count:" + count);

}

}