**Assignment 1 ANSWERS**

1. Check the given number is EVEN or ODD

Answer-

START

Input number

Using modulus to determine the remainder.

rem = number % 2

rem == 1

Given number is ODD

Given number is EVEN

END

true

false

**Algorithm:**

Step 1: START

Step 2: Input a number

Step 3: Using modulus to determine the remainder.

rem = number % 2

Step 4: If rem is equal to 1, then display “Given number is ODD”.

Step 5: If rem is equal to 0, then display “Given number is EVEN”.

Step 6: END

1. To find the factorial of the given number.

Flowchart:

START

Input number whose factorial is to calculated; num

Consider fact = num

Using for loop,

for i=num, i > 1, i -- ;

num -= 1

fact = fact\*num

END for loop

Output Factorial; fact

END

Algorithm:

Step 1: START

Step 2: Input the number whose factorial needs to be calculated; num

Step 3: Store the number in another variable; fact = num

Step 4: Using for loop,

For i = num; i >1; i-- ;

num -= 1;

fact = fact \* num;

Step 5: Display the output of the resultant factorial; fact

Step 6: END

­­­­­­­­­­­­­­­­­­­­­­

1. Find the Factorial of a number using Recursion.

Flowchart:

START

Input number whose factorial is to calculated; num

Define a function using recursion to calculate the factorial.

static int factorial(int num){

if (num == 0)

return 1;

else

return (num \* factorial(num-1));

public static void main(String args[]){

int num = 5;

int fact = factorial(num);

//print output fact

}

Print output; fact

END

Algorithm:

Step 1: START

Step 2: Take input number whose factorial needs to be calculated.

Step 3: Define a function using recursion to calculate the factorial.

static int factorial(int num){

if (num == 0)

return 1;

else

return (num \* factorial(num-1));

public static void main(String args[]){

int num = 5;

int fact = factorial(num);

//print output fact

}

Step 4: Print the output of the result; fact

Step 5: END

­­­­­­­­­­­­­­­

1. Swap two numbers without using third variable approach.

Flowchart:

START

Input two numbers to swap; n1, n2.

Display numbers before swapping

n1 = n1 + n2

n2 = n1 – n2

n1 = n1 – n2

Numbers have been swapped now

Display swapped numbers

END

Algorithm:

Step 1: START

Step 2: Take input numbers to be swapped; n1, n2

Step 3: n1 = n1+n2

n2 = n1-n2

n1 = n1-n2

Step 4: Output swapping result

Step 5: END

1. How to check the given number is Positive or Negative in Java?

Flowchart:

START

Input number; n

Compare the number with zero, if its greater than or less than zero.

n > 0

NEGATIVE

POSITIVE

END

Algorithm:

Step 1: START

Step 2: Input number; n

Step 3: Compare the number with zero, if its greater than or less than zero.

Step 4: If n > 0, then its positive

if n<0, then its negative

­­

1. Write a Java Program to find whether given number is Leap year or NOT?

Flowchart:

START

Input year; y

Perform modulus operation of the input year with 4.

y % 4 == 0

Given year is leap year

Given year is not a leap year

END

false

true

Algorithm:

Step 1: START

Step 2: Input year; y

Step 3: Perform modulus operation on y with 4.

Step 4: If year % 4 is zero, then given year is leap year, else it’s not.

Step 5: END

1. Write a Java Program to Print 1 To 10 Without Using Loop.

Flowchart:

START

Input number; n=1

Print n

Increment n

Print n

if n=10

END

Algorithm:

Step 1: Take input n=1;

Step 2: Display output n

Step 3: Increment n and display output n till value of n = 10

Step 4: END