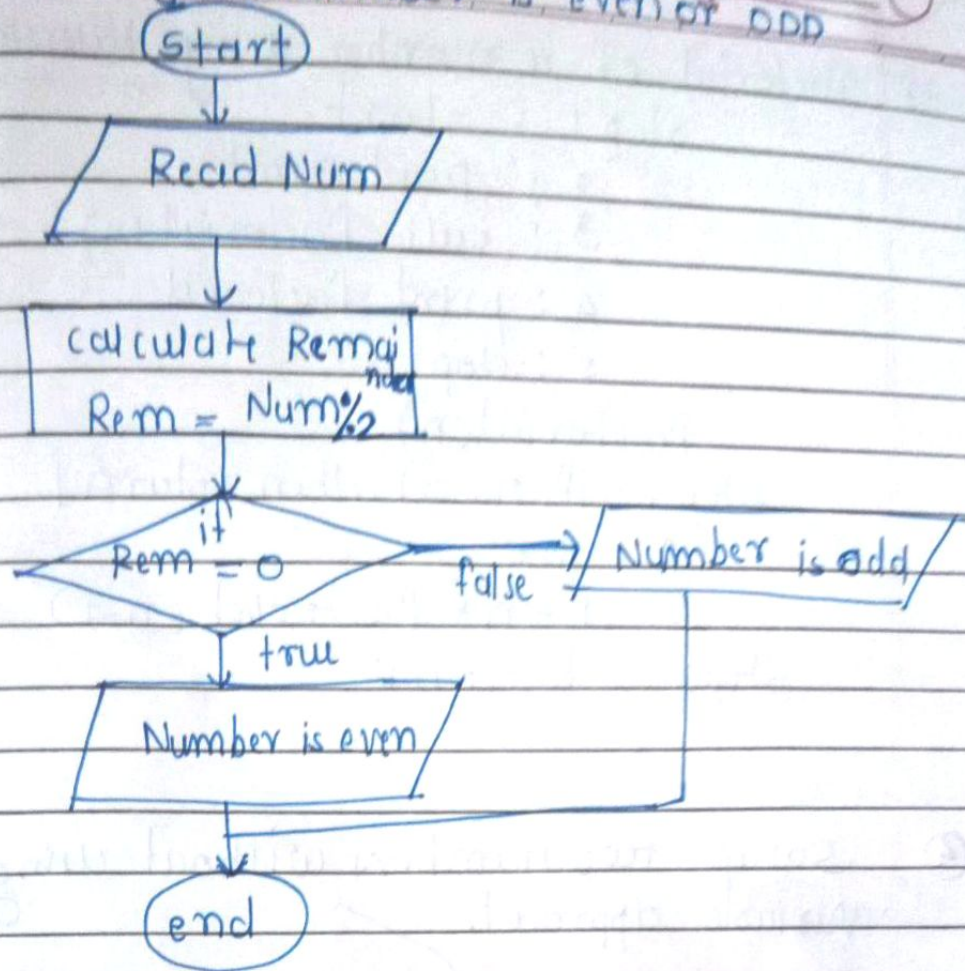
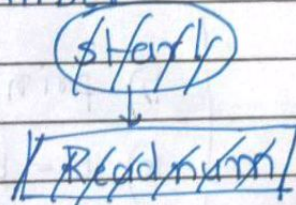


1. check the given number is even or odd



② write a java program to find the factorial of given number



Algorithms

- 1) start
- 2) Read number
- 3) from value of number, up to 1, Means $N-1, N-2, N-3, \dots, 1$
multiply each digit & update the final value
- 4) End program

3 factorial of a number using recursion

step 1 : start

2 : Read number

3 : call factorial(n)

4 : print factorial

5 : stop

factorial(n) -

step 1 : if $n == 1$, then return 1

2 : else

$f = n * \text{factorial}(n-1)$

step 3 : Return f

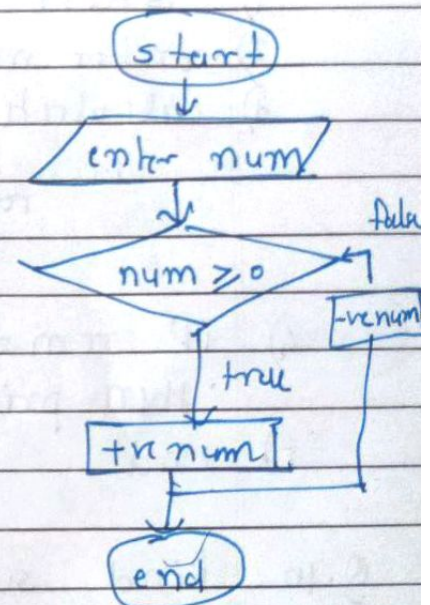
4. swap two ~~vari~~ numbers without using third ~~variable~~ ~~approach~~ variable

Algorithm

- 1) start
- 2) Enter two numbers n_1, n_2
- 3) print number n_1 & n_2
- 4) $n_1 = n_1 + n_2$
 $n_2 = n_1 - n_2$
 $n_1 = n_1 - n_2$
- 5) print n_1 & n_2
- 6) End

5. How to check the given num is +ve or -ve in java?

- 1) start
- 2) enter the number
- 3) check number ≥ 0
- 4) if true then +ve number
 otherwise number is -ve



6. Find whether given number is leap year or NOT

- 1) start
- 2) enter the number
- 3) check whether the number is ~~at~~ leap year or not by dividing the number by 4, 100 & 400
- 4) if num divisible by 4 but not 100 then it is a leap year
- 5) Also if number divisible by 4 & 100 & 400 then it is a leap year
- 6) else not a leap year

⑦ print 1 to 10 without using loop

1) start

2) initialize $k = 1$

3) print k

4) $k = k + 1$

5) print k

6) $k = k + 1$

7) print k

8) $k = k + 1$

9) print k

10) $k = k + 1$

11) print k

12) $k = k + 1$

13) print k

14) $k = k + 1$

15) print k

16) $k = k + 1$

17) print k

18) $k = k + 1$

19) print k

20) $k = k + 1$

21) print k

22) end

NOT

que 8 print digit of given number

- 1) start
- 2) enter the given number
- 3) calculate remainder
 $rem = num \% 10$
- 4) then divide number by 10
 $num = num / 10$
- 5) then print rem
- 6) repeat step 3 to 5 until $n = 0$

que 9 print all factor of given number

- 1) start
- 2) enter number
- 3) calculate remainder from $i = 1$ to $i = num$
i.e.
 $rem = num / i$
 $i++$
- 4) if $rem = 0$
then print i
- 5) end

Q. 10 find sum of digit of given number

- 1) start
- 2) enter the number
- 3) calculate remainder
 $rem = num \% 10$
- 4) initialize $sum = 0$ then
 $sum = sum + rem$
- 5) $num = num / 10$
- 6) repeat 3 & 4 step till $num = 0$
- 7) print sum
- 8) end

11 Find smallest of 3 number (a, b, c)

- 1) start
- 2) enter number a, b, c
- 3) check if $a \leq b$ & $a \leq c$
- 4) if condition is true, a is smallest, if condition is false the
- 5) check if b is less than c & b is less than a then b is smallest
else c is smallest
- 6) stop

13 add two number without using the arithmetic operators in java.

- 1) start
- 2) enter number
- 3) calculate $n = n/10$ remainder
print ~~n~~
 $rem = n \% 10$
print rem
then calculate $n = n/10$ repeat until $n = 0$
- 4) end

14) Find GCD of two given number

- 1) start
- 2) Declare two variable n_1, n_2
 $gcd = 1, i = 1$
- 3) Input n_1, n_2
 - i) check if $i \leq n_1$ & $i \leq n_2$
if Yes
if $n_1 \% i == 0$ & $n_2 \% i == 0$
if Yes $gcd = i$
~~print gcd~~
 $i = i + 1$
 - if No
 $i = i + 1$

if No
printed
end

15) Algorithm to calculate LCM of two number

1) start

2) initialize two variable num_1 & num_2

3) find & store maximum of num_1 & num_2
to separate variable consider max

4) if max is divisible by both number then
max is LCM

5) print it

if max not divisible by num_1 & num_2
then increase max by 1 unit max is
divisible by n_1 & n_2

6) end

16 calculate LCM of two number using prime factor
method - 1) start 2) Input n_1 & n_2

3 - find prime factorization of each number

4 - write each number as a product of primes

5 - common factor take only one

6 - LCM is multiplication of prime factor

7 - end

17) Algorithm to check number is ~~palindrome~~ or not
palindrome

1. start

2. Input number x

3 reverse the number

4. if reverse number = number

then number is ~~palindrome~~ palindrome

~~otherwise~~ else not palindrome

5. end

- 18) 1) start 2) Input number N
3) check if the number N has 2 as a prime factor
Do this by continuously dividing N by 2 & checking if the remainder is 0
4) check for odd prime factors of N Do this by continuously dividing N ~~from 2~~ by 2
5) If rem. is not equal to 0 then ~~increase~~ increase 2 by 1 i.e. 3 then check if it is factor of num or not Do this process again again if it is factor then check if it is prime or not
6) print value of i if it is prime & a factor of num.
7) end

19) algorithm to print series of even numbers
2, 4, 6, 8, 10, 12, 14, 16, 18,

- 1) start
- 2) initialize variable i to 1 & also initialize upto what number series is printed suppose n
- 3) while $i \leq n$
- 4) if $i \% 2 == 0$
- 5) print number i
- 6) increment value of i
- 7) stop

20 Algorithm to find odd number series 1, 3, 5, 7, 9, 11, 13,

1) start

2) initialize $i = 0$

3) check whether $i = 100$ reached

if Yes \rightarrow stop the program

if No $\rightarrow i \% 2 \neq 0$

if true print i

else compute $i = i + 1$

4) stop