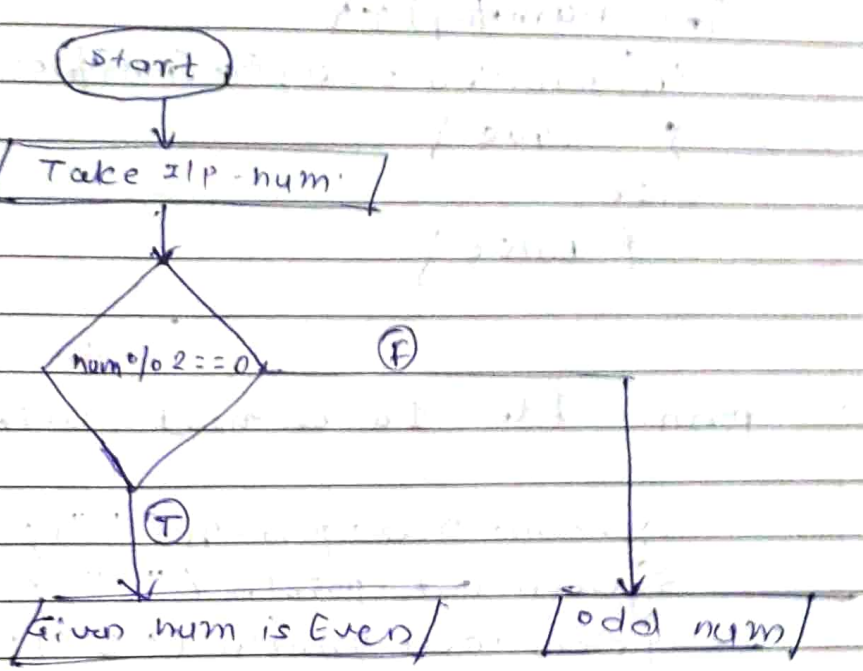
Java - Basic.

T-34/50

Assignment 1.

* Algorithm / Flowchart.

check given no. is Even / odd -



2) Factorial of num -

I/p - int i = 1, F = 1, take num from user.

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

F = (F x i) ; }

// F is factorial of number.

3) Given num is +ve / -ve ?

I/p - num

if (num >= 0) {

True }

else

False.

4) Find yr is leap / not -

Ip - yr.

```
if( (num%4) &&
if((num%4 == 0) && (num%100 != 0) || (num%400==0
{ True }
else
{ False }
```

5. print 1 to 10 without using loop.

```
System.out.println("1");
System.out.println("2");
System.out.println("3");
System.out.println("4");
System.out.println("5");
System.out.println("6");
System.out.println("7");
System.out.println("8");
System.out.println("9");
System.out.println("10")
```

6. write program to print digits of num.
but (10 to 99)

① if(num is two digit)

② a = num % 10 ... unit digit

③ num = num/10

④ b = num

DATE

if (num is 3 digit)
a = num % 10
num = num / 10
b = num % 10
num = num / 10
c = num

while (num != 0) {
int a = num % 10
num = num / 10
print (a);
}

7 print Even number -

- for (i = 2; i <= 100 (n); i = i + 2) {
  ~~if i;~~
  if (i % 2 == 0) {
  print (i); }
} ~~else~~

8 print odd number -

- for (i = 1, i <= n, i++)
- if (i % 2 != 0)
- print (i)

9. print all factors of number.

I/p num
int i = 1
for (i = 1, i <= num, i++) {
if (num % i == 0)
print (i);
}

10. Swap two num without using 3rd no. approach.

```
I/P - x, y;        // x=13, y=12
x = x + y          // x = 13+12 = 25
y = x - y          // y = 25-12 = 13
x = x - y          // x = 25-13 = 12
```

11. Find smallest num of 3 num.

```
I/P - a, b, c;
int min = (a<b) ? (a<c) ? a : c : (b<c) ? b : c;
print(min);
```

12. Write a pro to print sum of digit of numbers.

```
I/P - num - 4994 (4 digit)
int a, b, c, d;
a = num%10;
num = num/10;
b = num%10;
num = num/10;
c = num%10;
d = num/10;
int sum = a+b+c+d;
```

13 Add num without using Arithmetic operator

```
I/P - a, b;
for(i=1; i<b; i++){
a++;
}
print(a);
```

14 I/P - num;
int r = 0;
while (num != 0) {
r = (10 x r) + num % 10
num = num / 10; }
print (r);

15. write pro. to GCD of 2 num.

- I/P - 2 num (a, b). to find GCD
- initialize GCD = 1
- check a or b equal to 0, if zero then store non zero value as GCD ~~bet~~ num (skip loop)
- for (i = 1, i <= min(a, b), i++)
- check both a & b divisible i, if yes then print (i).

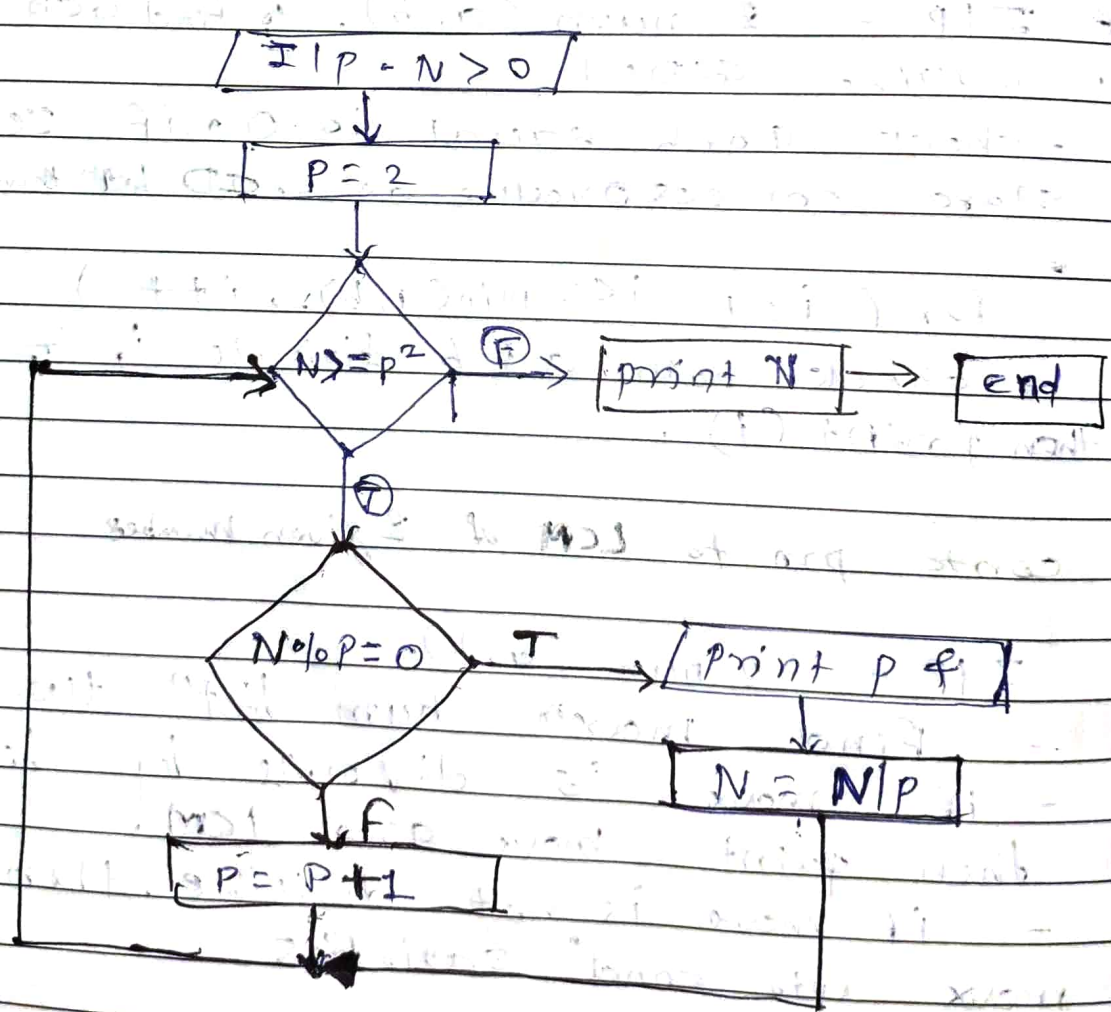
16 write pro to LCM of 2 given numbers

- I/P - 2 num (a, b).
- Find max num bet" two.
- if max is divisible by both a, b then print max as a LCM.
- if max is not divisible, then increment max upto cond" satisfies

17. Check number is palindrome or not.

- I/p - number
- Hold number is temporarily variable.
- Reverse the given number.
- if (temp == Reverse) then number is palindrom print it. else it is not palindrome.

18. Print all Prime factors of number.



19 Factorial of num using recursion.

- I/P : number. n;
- called : factorial(n)
- ~~Print factorial F~~

Factorial (n)

- If n = 1 then return 1

else

F = n X factorial (n-1)

- ~~return F & print~~.

print (F);

20. Write a pro to LCM of two num using prime factor method.