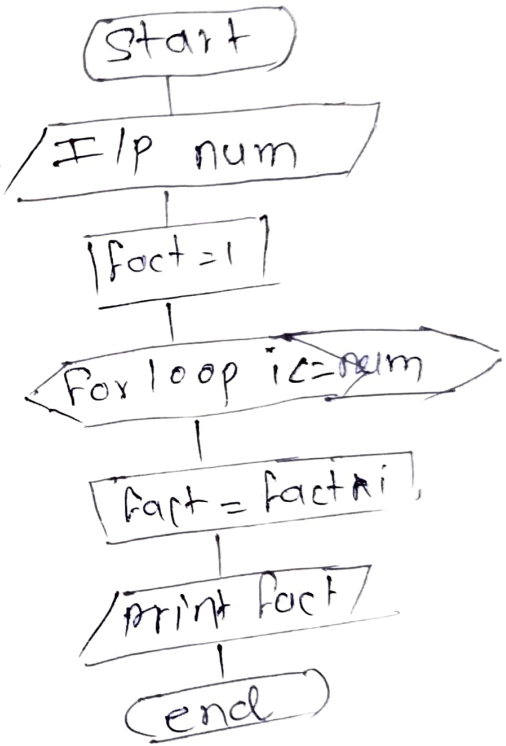
(2) Factorial

1) Start
2) I/p num
3) declared fact = 1;
4) for loop (i <= num)
   i = 1.
5) Inside loop
   fact = fact * i;
6) print fact value.
7) end

Flowchart



(8) (3) Factorial using recursion

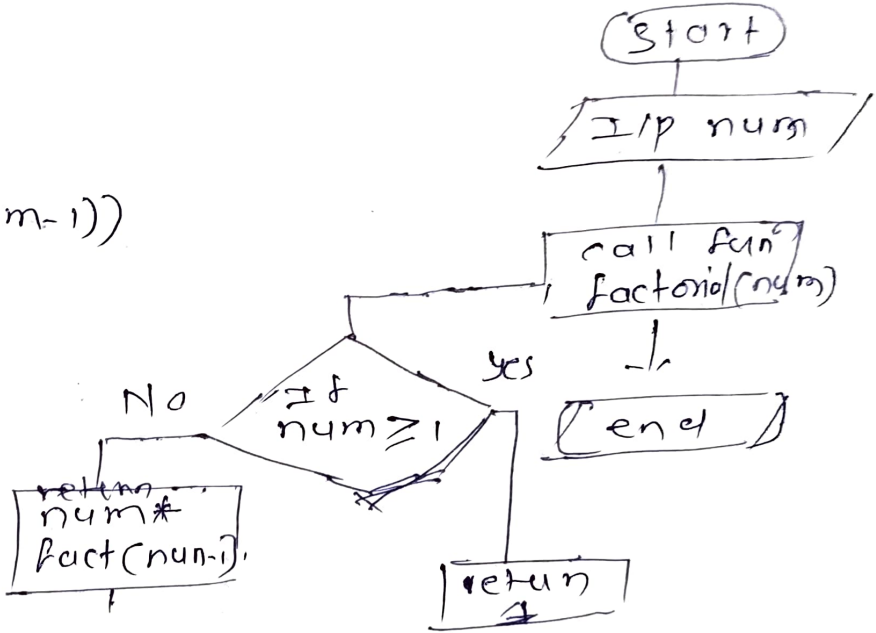
(a) factorial fun^n

(1) check con^n
If (num ≥ 1)
return 1

(2) else
return (num * fact (num-1))
call fun^n itself

main fun^n

(1) start
(2) take i/p num
(3) call fun^n factorial num
(4) print that fun^n.

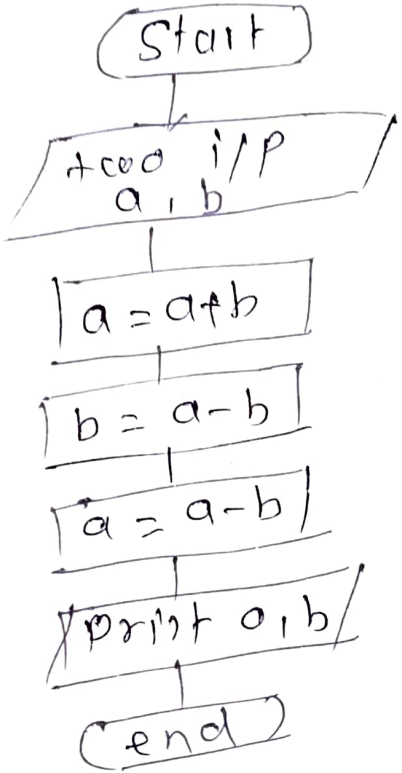


⑤ Swapping num

Algo

1) Start
2) take two num i/p
3) a = a+b
4) b = a-b
5) a = a-b
6) print a, b
7) end

Flowchart

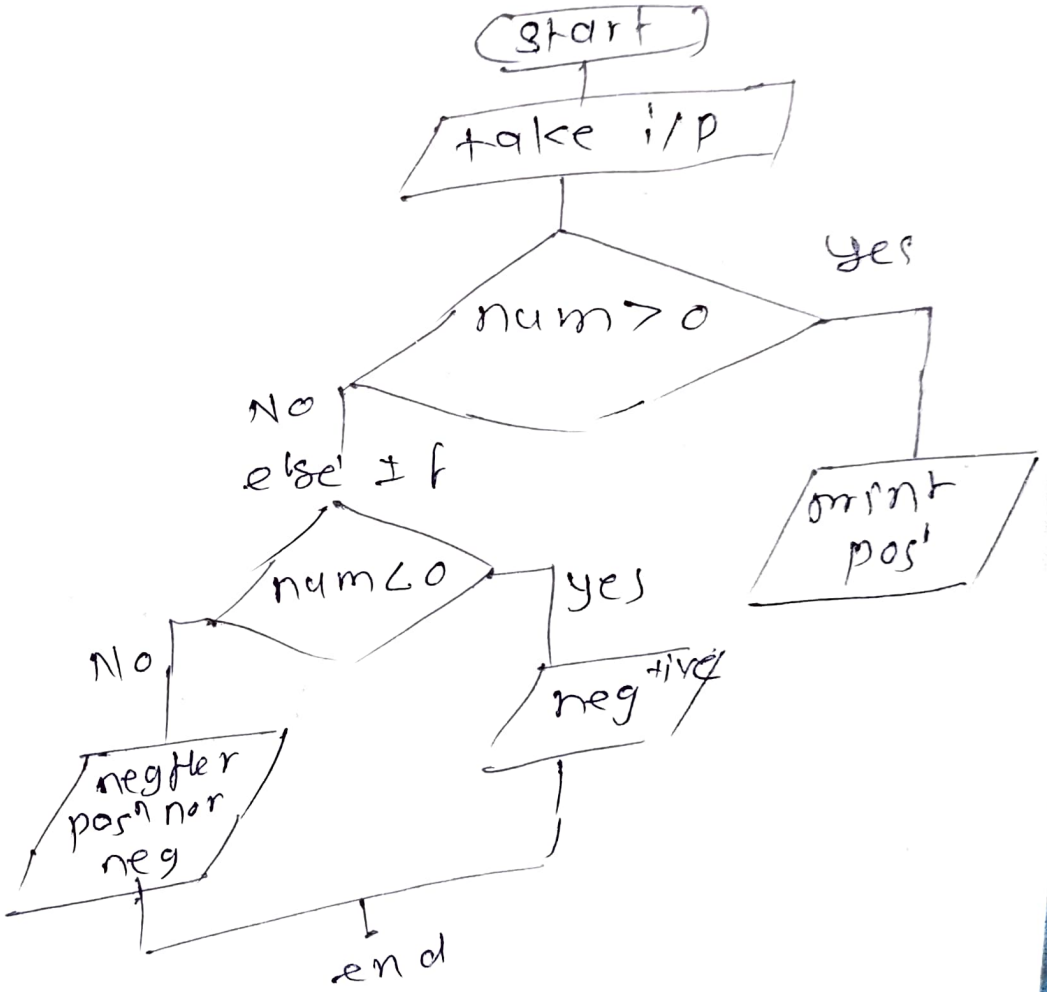


(5) positive or negative

Algo

1) start
2) take i/p num
3) condition
   If num > 0
   positive
4) else if num < 0
   negative
5) else
   neigther pos^n nor negative
6) end.

Flowchart



(6) Leap year

(1) start

(2) I/p year

(3) con^n
If year%==0

(4) check another con^n
If year%.100 == 0
True
If year%.400 == 0

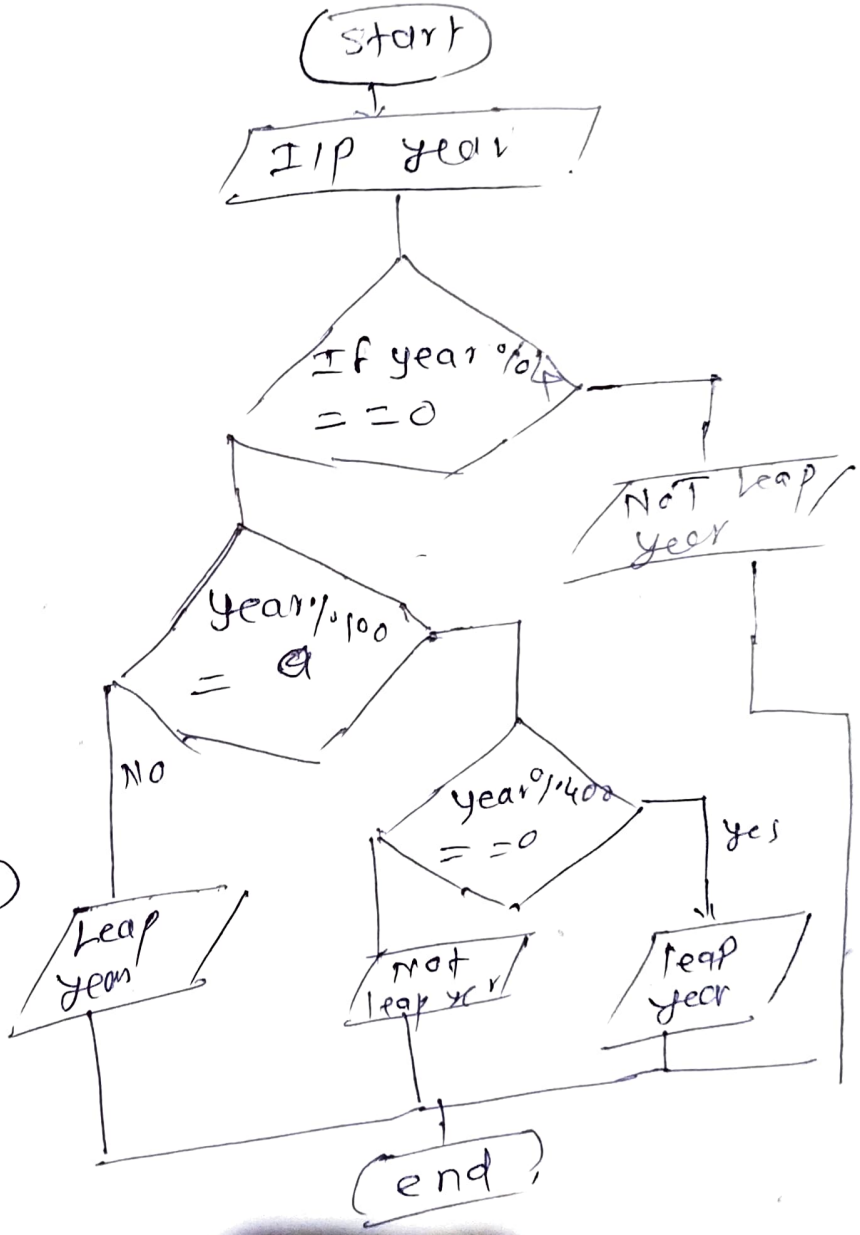
(5) print leap yer

(6) else not leap yeer

(7) else (!100% == 0)
prit leap year

(8) else (! year% 4 == 0)
not leap year

(9) stop.

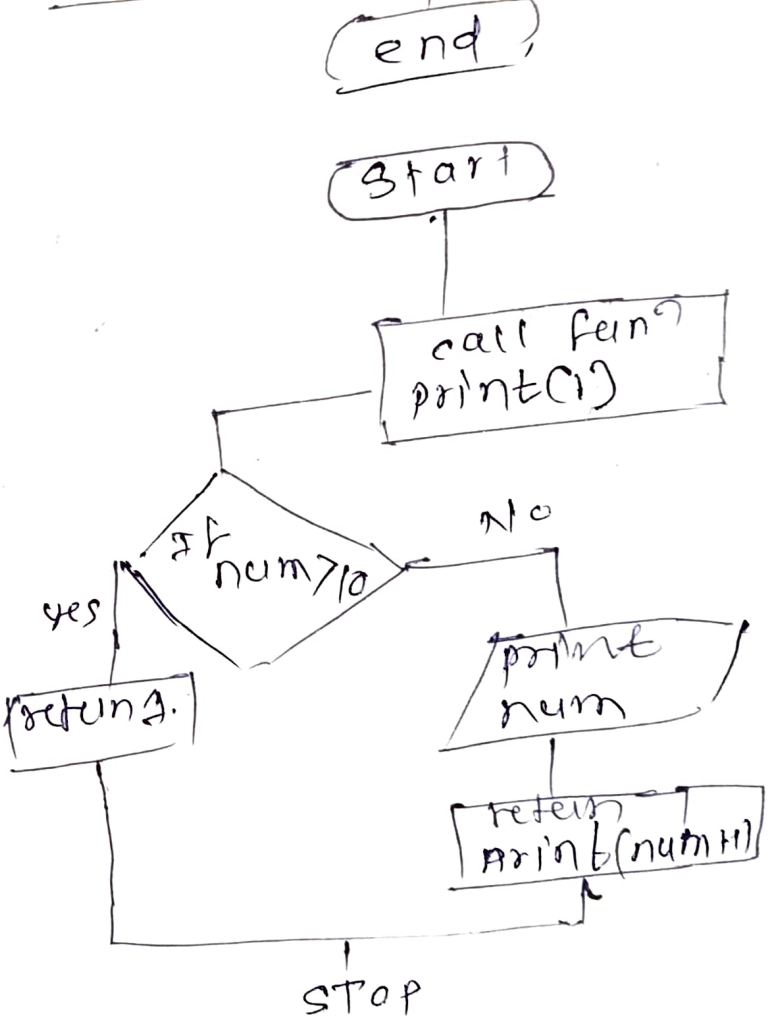


(10) Print 1 to 10 100 P

(7) Print 1 to 10 without loop

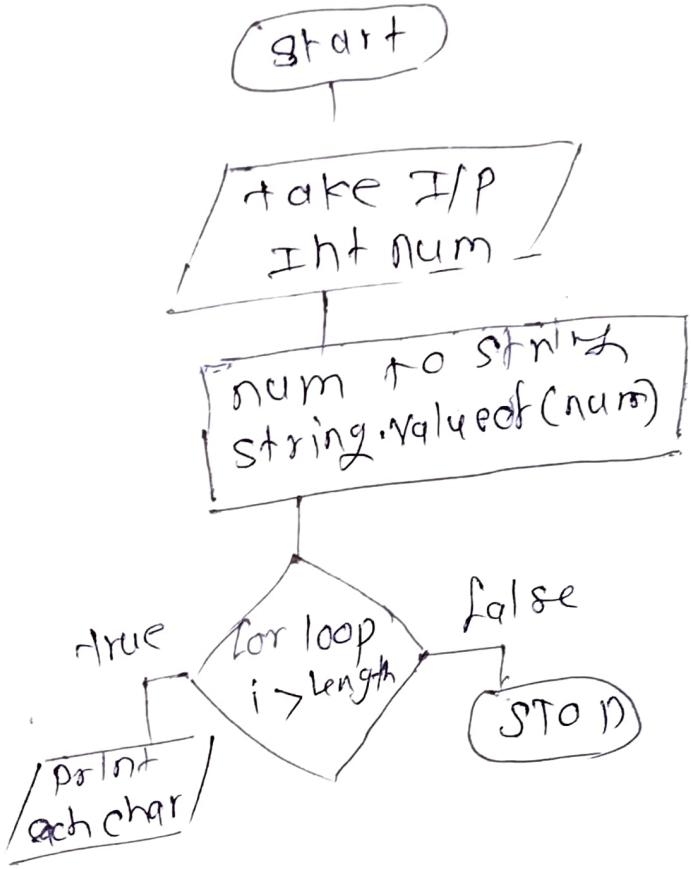
Print fun()
1) If num > 10
   return 1
2) else
   print num
3) return
   print(num+1)
   ↓ call fun() itself

main
1) start
2) call print fun()
3) parameter to fun()
   no. 1.



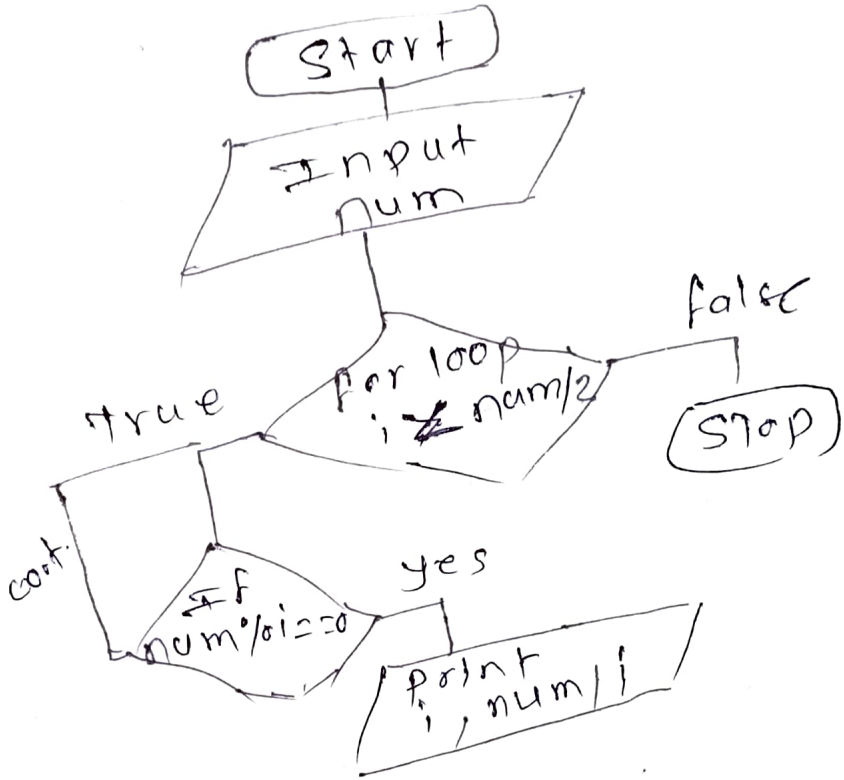
(9) print digit of Num.

1. Start
2. take I/p int
3. int covert into string
4. using string.valueof(i);
5. use for loop until length of num
6. print char at each posn
7. stop.



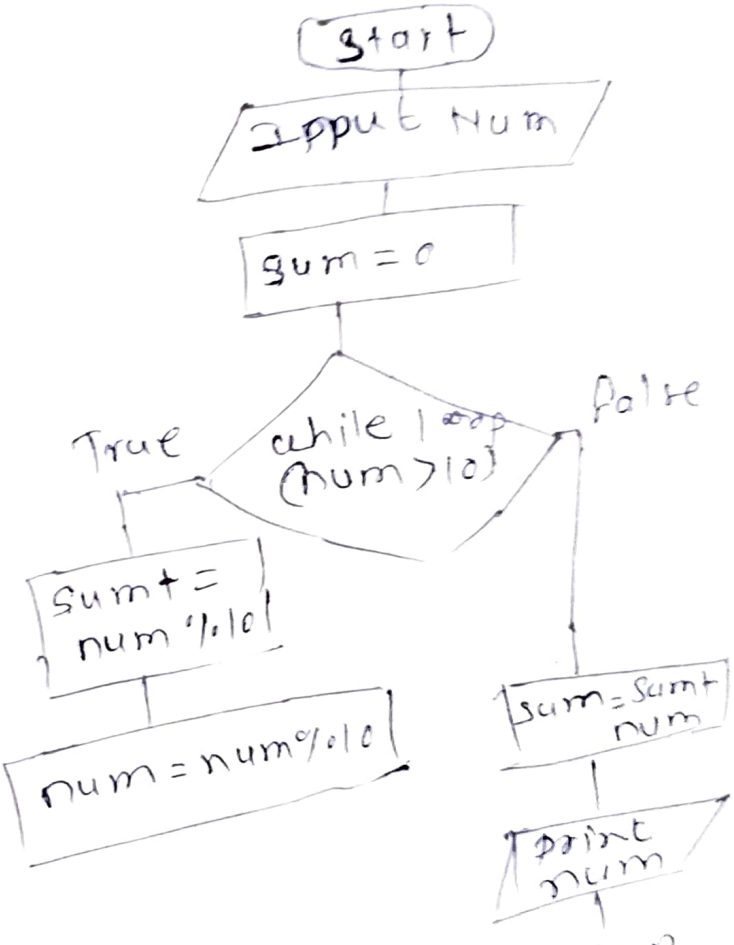
⑨ print factors

1. start
2. Input
3. for loop until num/2
4. con^n If num % i == 0
5. print ;
6. print (num/i)
7. stop



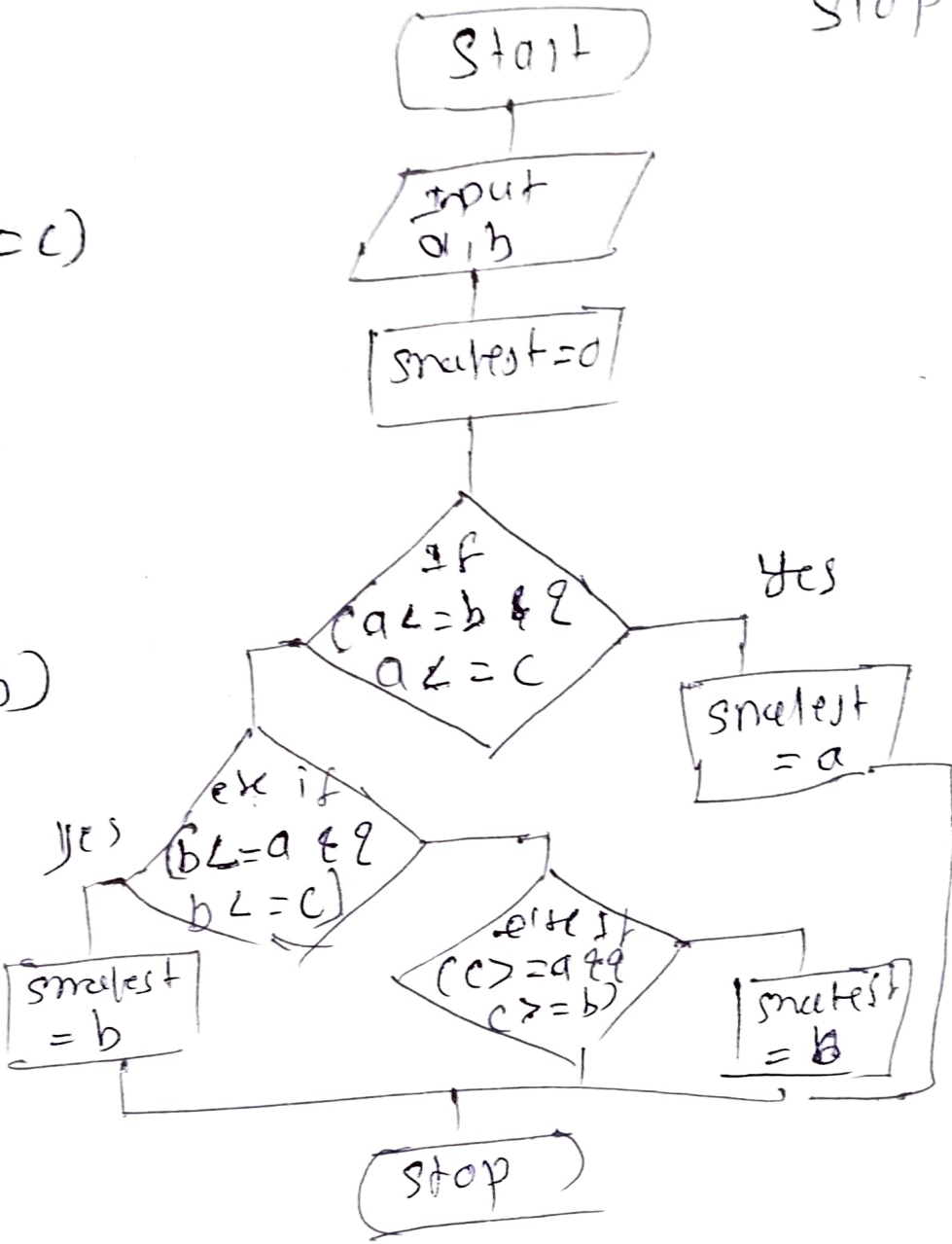
(10) Sum of Digits

1. start
2. Input
3. sum = 0
4. while loop (num > 10)
5. sumt = num % 10;
6. num = num / 10;
7. after loop
   sum = sum + num;
8. print sum
9. stop



(11) Smallest num.

(1) start
(2) Input a, b
(3) int smallest;
(4) If (a<=b && a<=c)
smallest = a;
(5) else if
(b<=a && b<=c)
smallest = b;
(6) else if
(c>=a && c>=b)
smallest = c
(7) print(smallest)

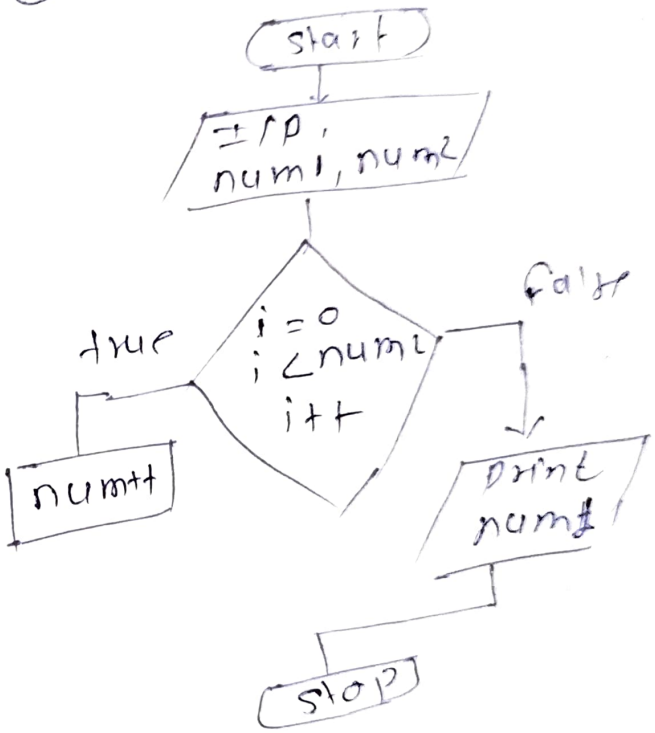


num

stop

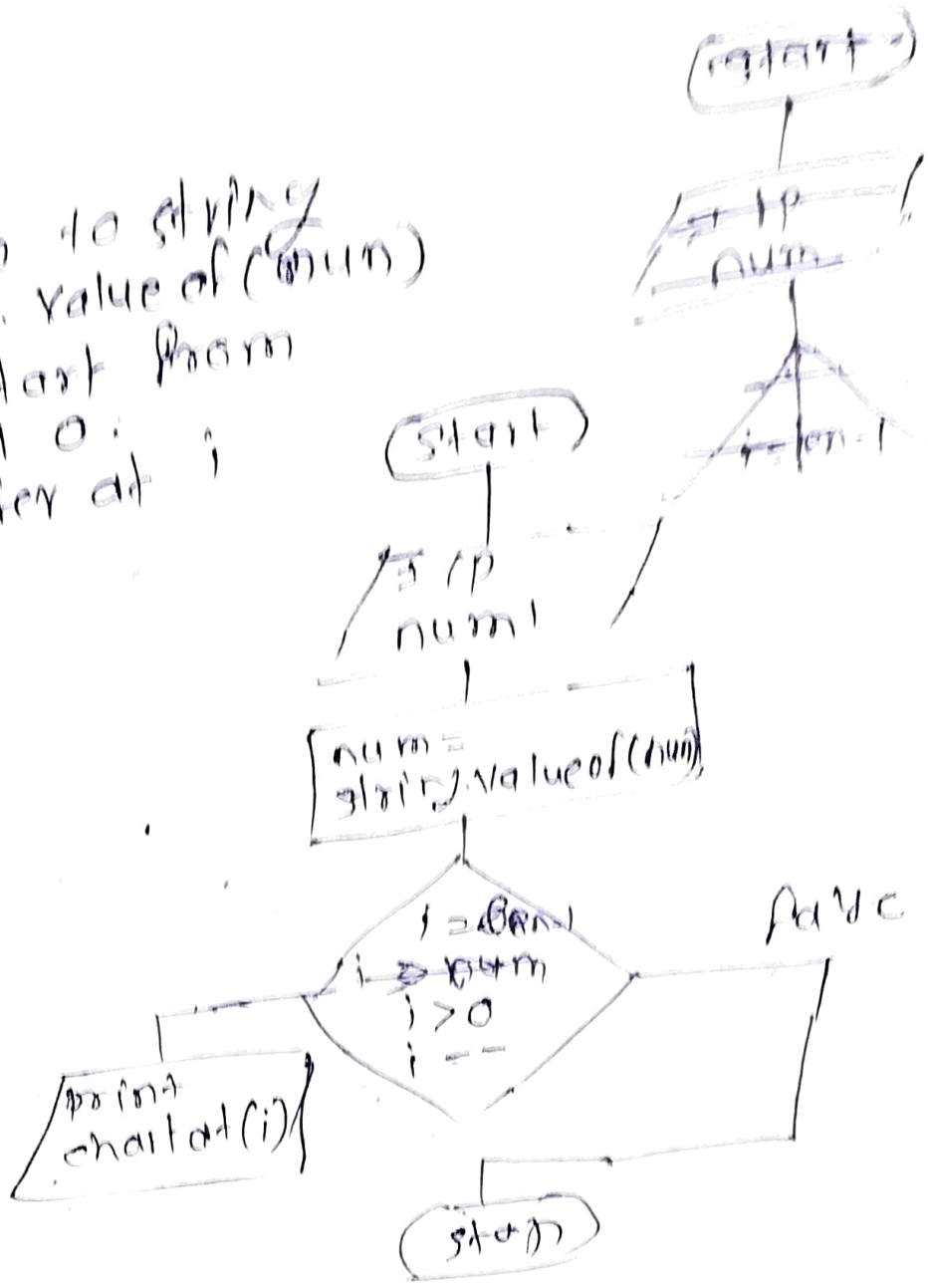
12 Add two no. without using Arithmetic.

1. Start
2. take i/p num1, num2,
3. use for loop
   i=0, i<num2,
   i++.
4. num1++;
5. print num1.
6. Stop



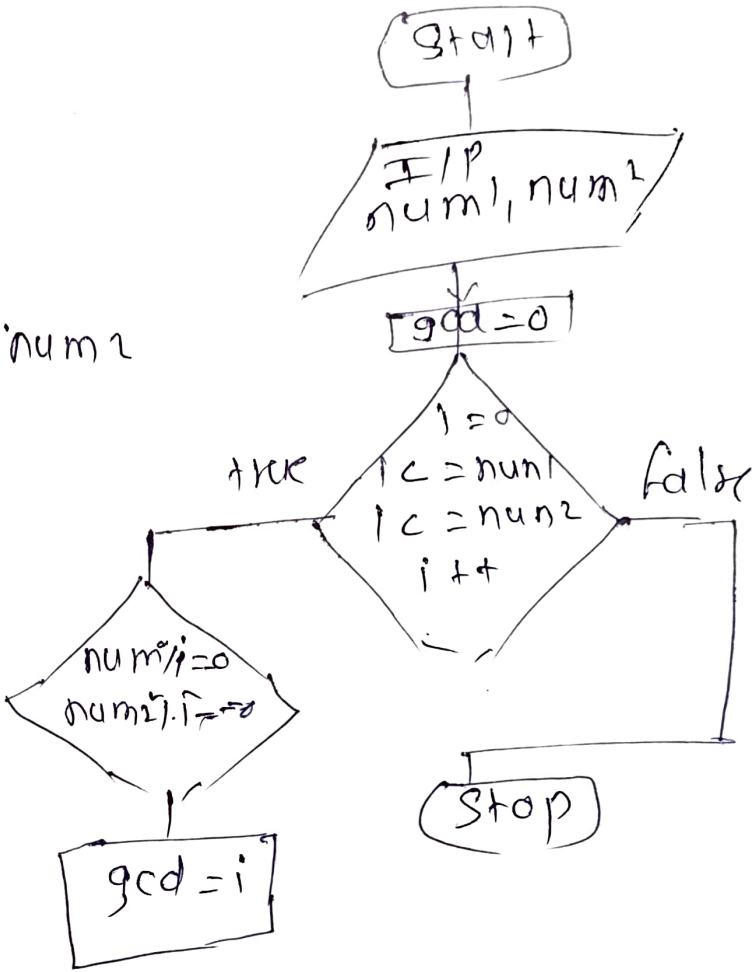
(13) Reverse given number.

(1) start
(2) Input num1.
(3) num convert to string
num = string.valueof(num)
(4) for loop start from length-1 till 0:
(5) print character at i position.
(6) stop.

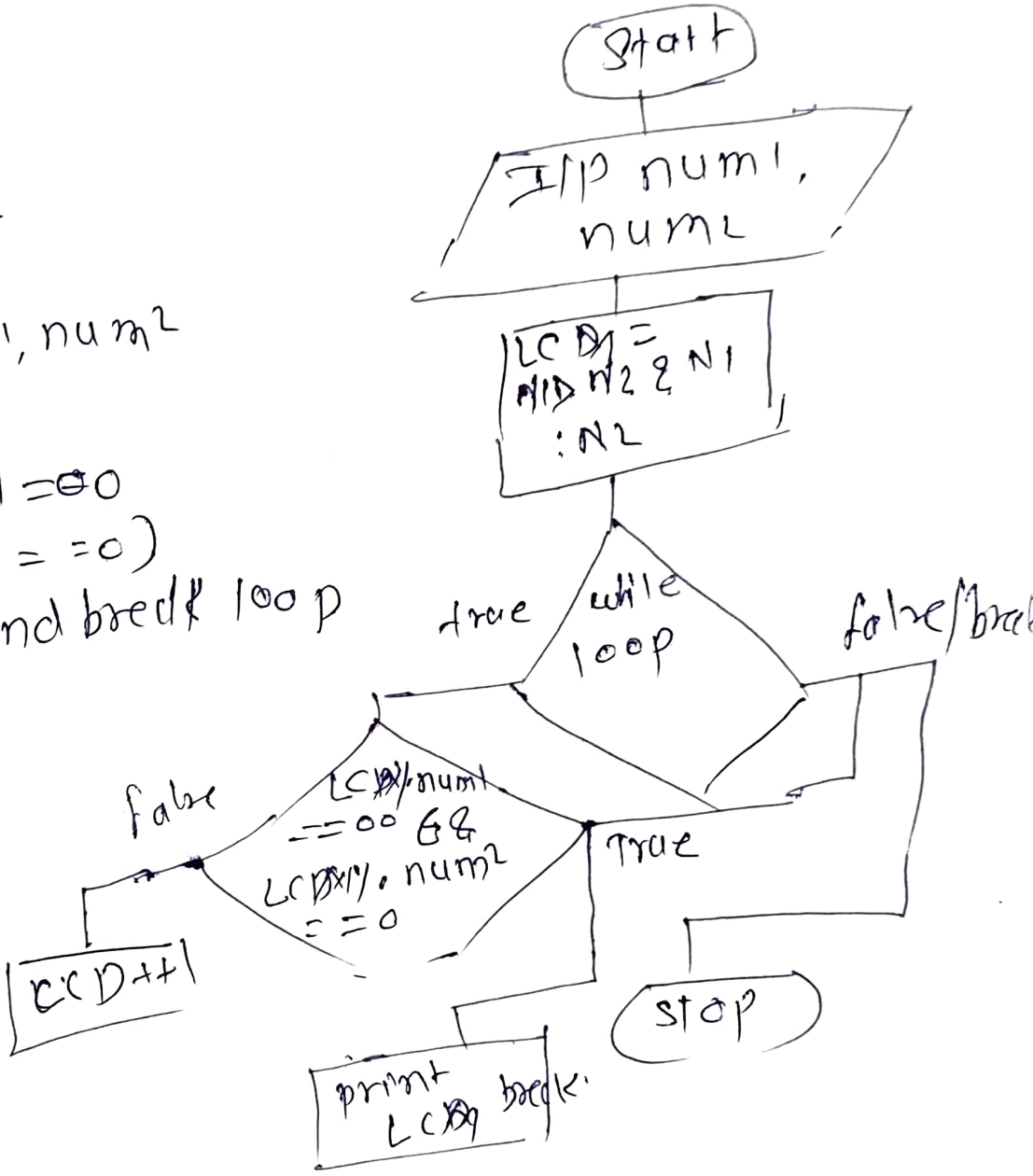


(4) GCD

(1) start
(2) I/p num1, num2
(3) initialize gcd = 0.
(4) for loop
    i = 0, i <= num1 and num2
    i++
(5) If (num1 % i == 0 and
    num % i == 0)
(6) gcd = i
(7) stop.



(1) LCD
(2) start
(3) I/p num1, num2
(4) Initialize LCD = greatest in num1, num2
(5) while loop
(6) If (LCD % num1 == 0 and LCD % num2 == 0)
(7) print LCD and break loop
(8) else LCD++
(9) stop



(17) Palidrome num

(1) start

(2) I/p num

(3) convert into string

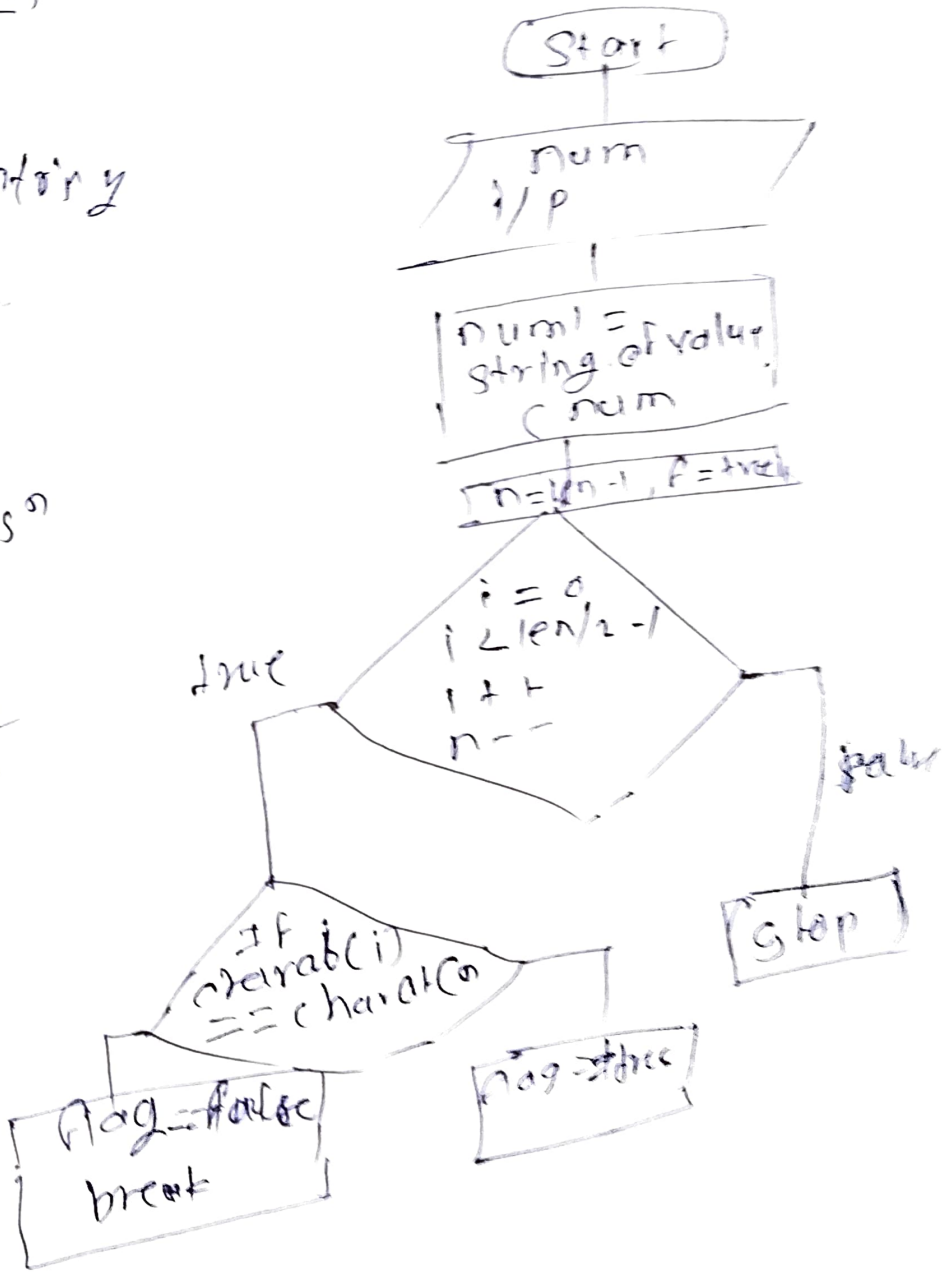
(4) for loop
i < len/2 - 1
i++
Len n--

(5) If con^n check
i posi^n and n pos^n
equal or not

(6) If flag = true
print palidrom

(7) else not palidrom

(8) Stop



(1) prime factor

main function

(1) start
(2) take i/p
(3) for loop
    i = i
    i <= num
(4) If (num % i == 0
    and ~~num %~~
    prime(num / i)
(5) print (num / i)

prime fun^n

prime (num)

(1) boolean
    flag = true
(2) for
    i = 2
    i < num
(5) If num % i == 0
    flag = false
    break
(6) return flag.