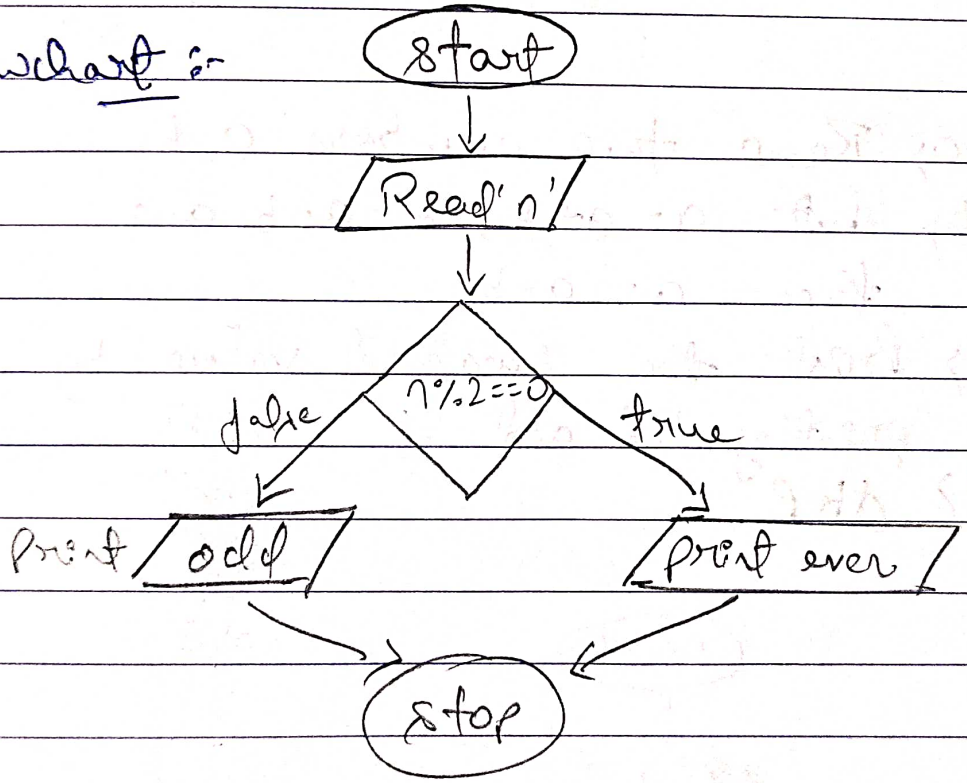
Assignment: 1

Q-1

Algorithm:-

a) Read the input 'n'.
b) Check whether 'n' is completely divisible by 2 or not.
c) If remainder is '0' print "even" else print "odd".
d) Stop

flowchart:-

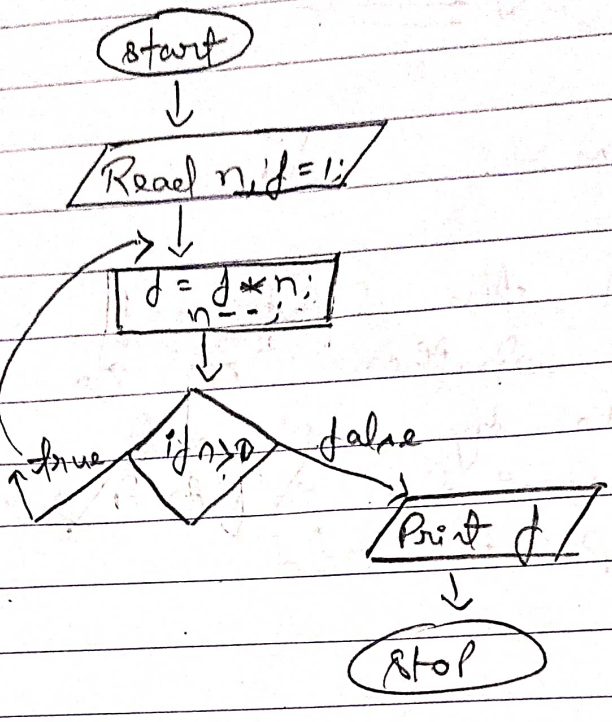


Q-2

Algo:-

a) Read the input 'n'.
b) Define f = 0;
c) factorial : f = (n) * (n-1) * (n-2) ... 1
d) Print the value of 'f'.
e) Stop

flowchart:

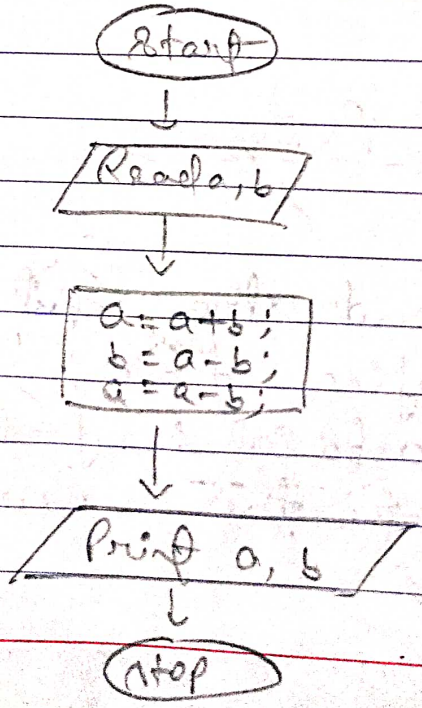


Q-4

Algo:

a) Read two numbers a, b.
b) Put a = a+b; b = a-b and then a = a-b
c) Print the swapped value by printing 'a' and 'b'.
d) Stop

flowchart:



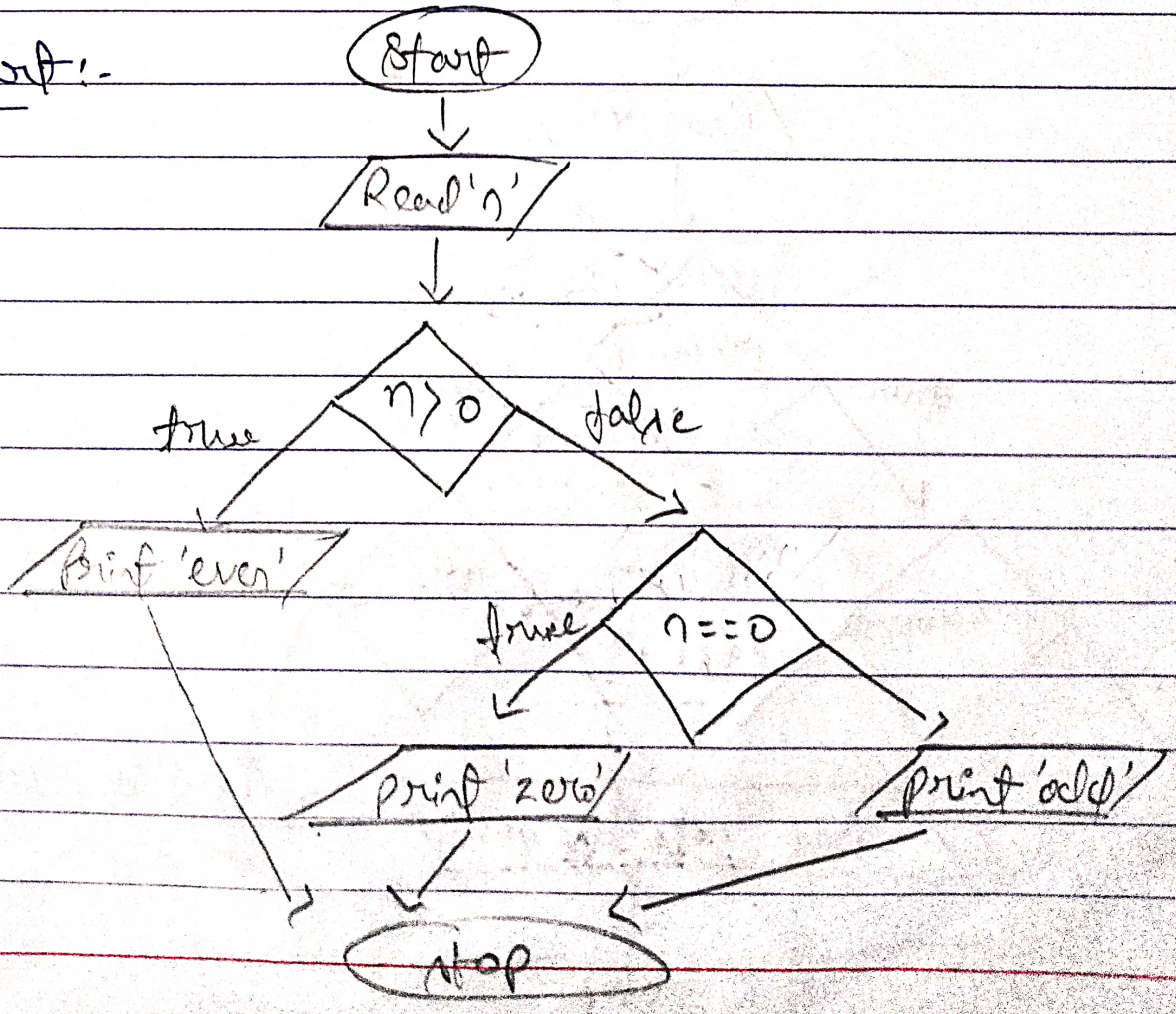
Q-4 Algo:

a) Read the number 'n'.
b) Divide the number by 2.
c) If remainder is 0 print even

Q-5 Algo:

a) Read the number 'n'.
b) Check whether no is greater than zero or less than zero.
c) If n>0 print 'positive' else if n<0 print negative else print zero.
d) Stop

Flowchart:-



Q-6

Algo:-

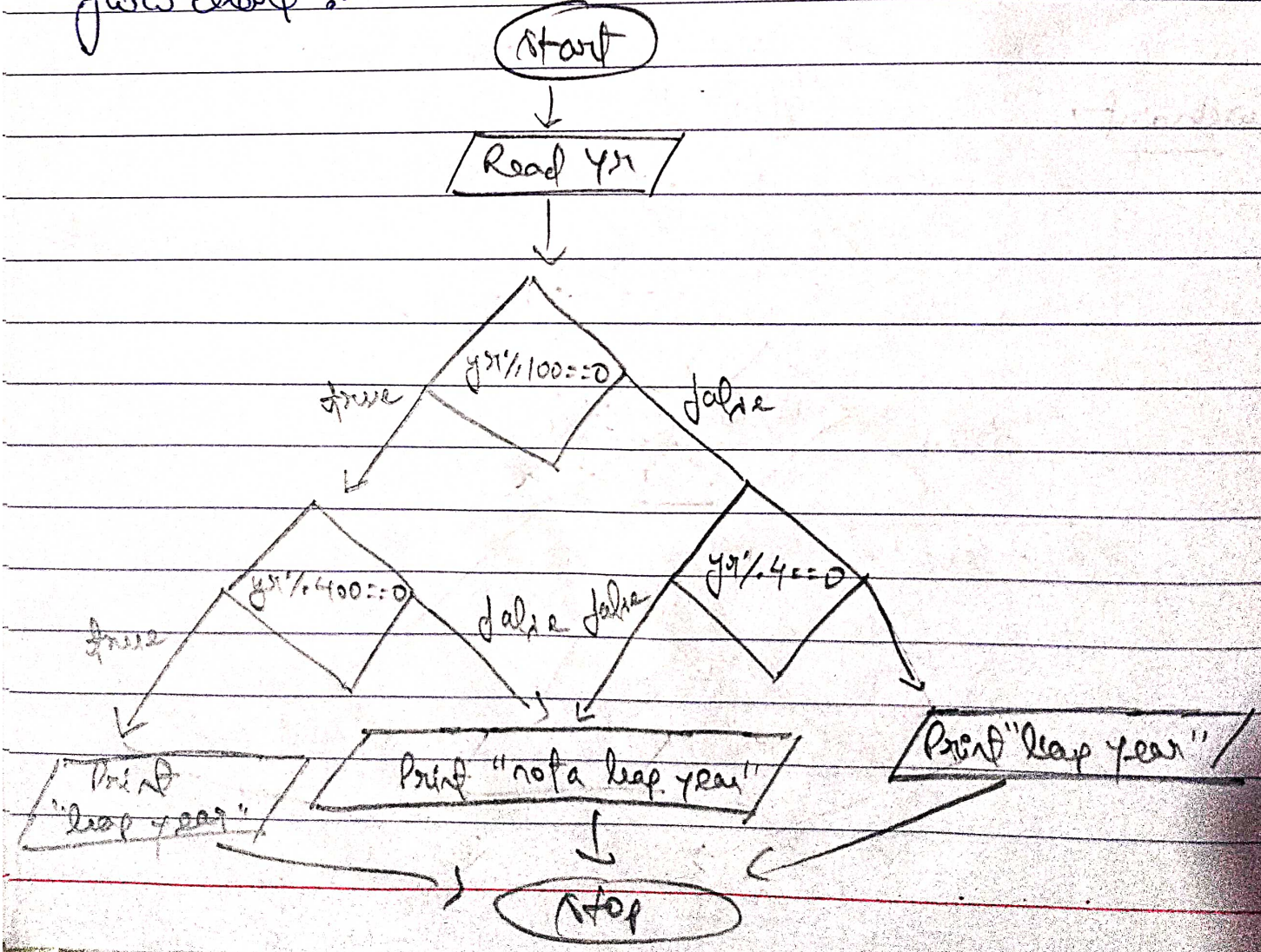
a) Read the year 'yr'.

b) If the number is completely divisible by 100 then divide (yr%100==0) it by '400' if it is completely (yr%400==0) divisible by '400' print "leap year" else "not a leap year".

c) Thereafter if year is not a century year ie (yr%100 != 0) then divide the number by '4' if (yr%4==0) print "leap year" else "not a leap year"

d) stop.

flow chart :-



Q-8

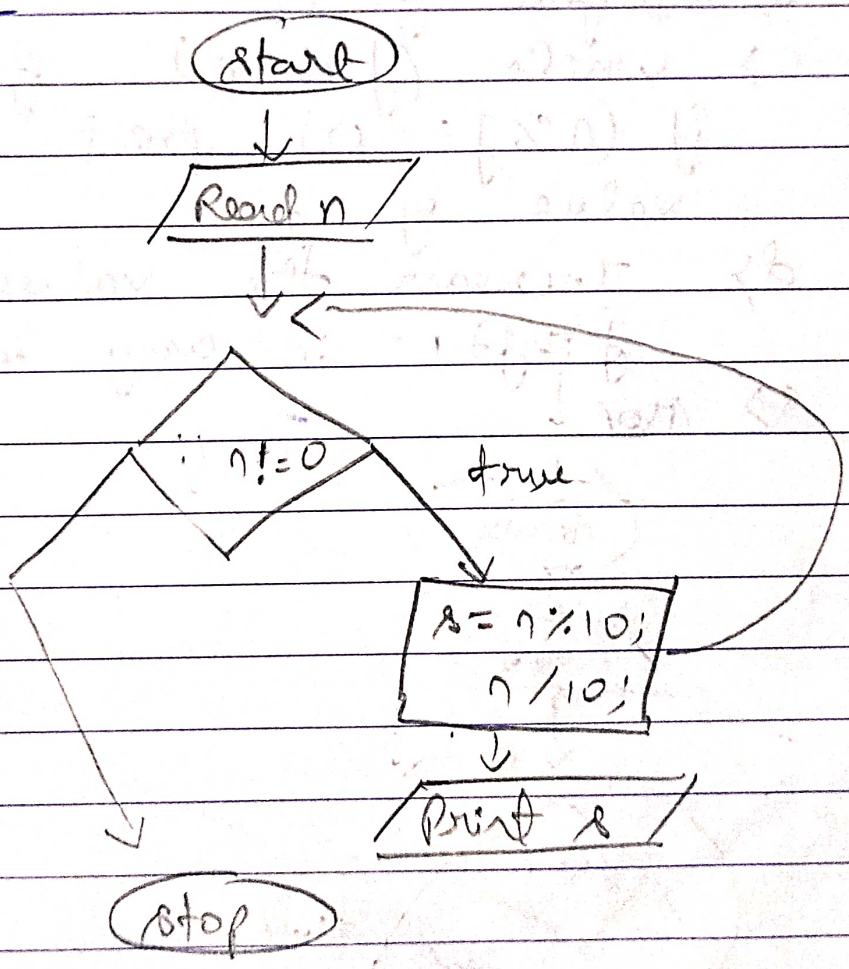
Algo: a) Read the number 'n'.

b) To separate digit: define s=0 s = n%10; repeat this until (n != 0) and make n = n/10; every time

c) Print the value of 's' every time to get each digit separated.

d) stop

flowchart:-



b) ~~from~~ each no if (n % ~~x~~ y == 0

Q-9

Algo:

a) Read the no. 'n'

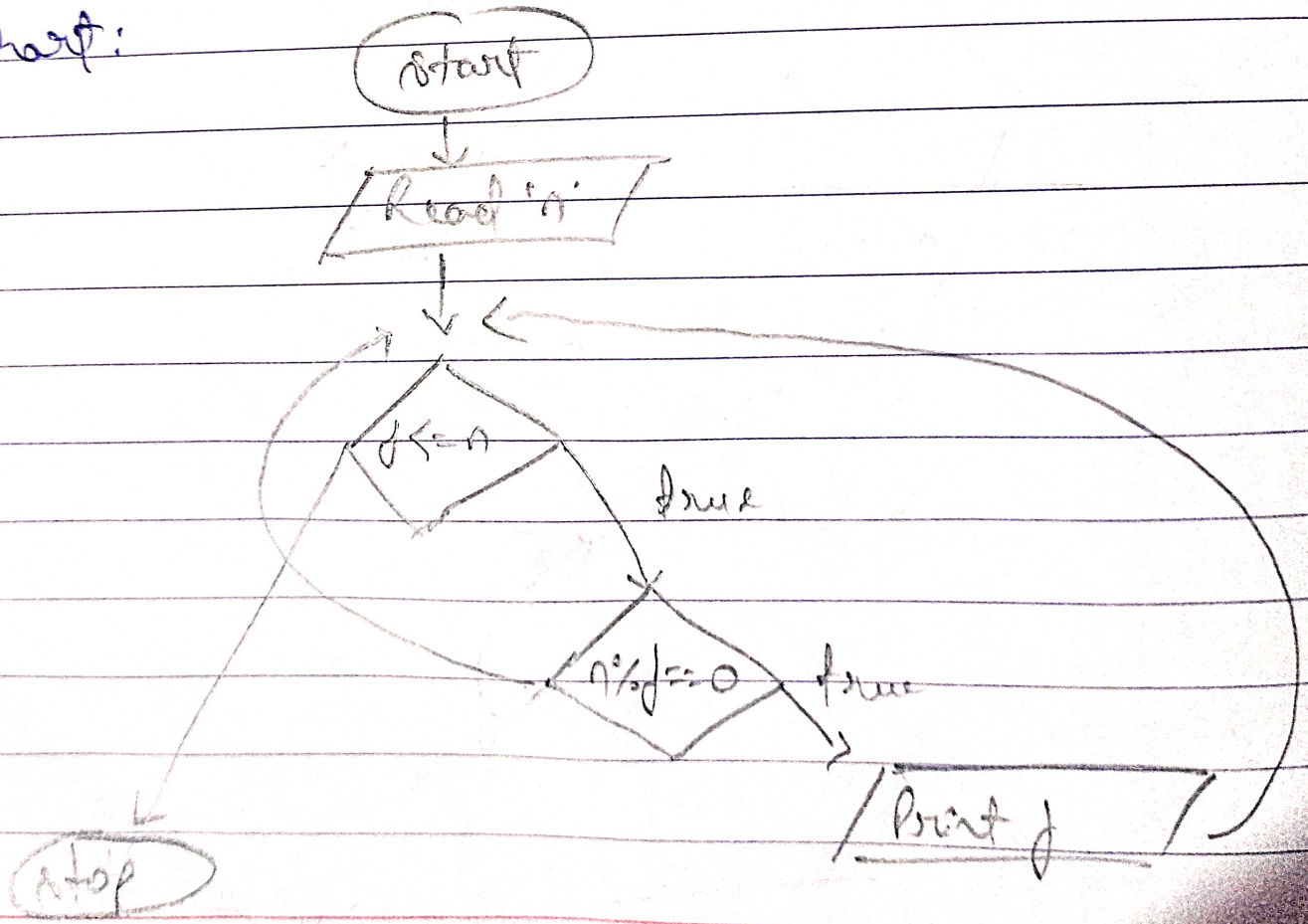
b) Define $f = 1$;

c) while ($f <= n$) ~~if (f n % 10~~ if ($n \% f == 0$) print the value of f.

d) Increase the value of f as $f = f + 1$; in every ~~loop~~ repetition.

e) Stop

Flowchart:



Q 10

Algo:

a) Read number 'n'

b) Define SM = 0, RM = 0.

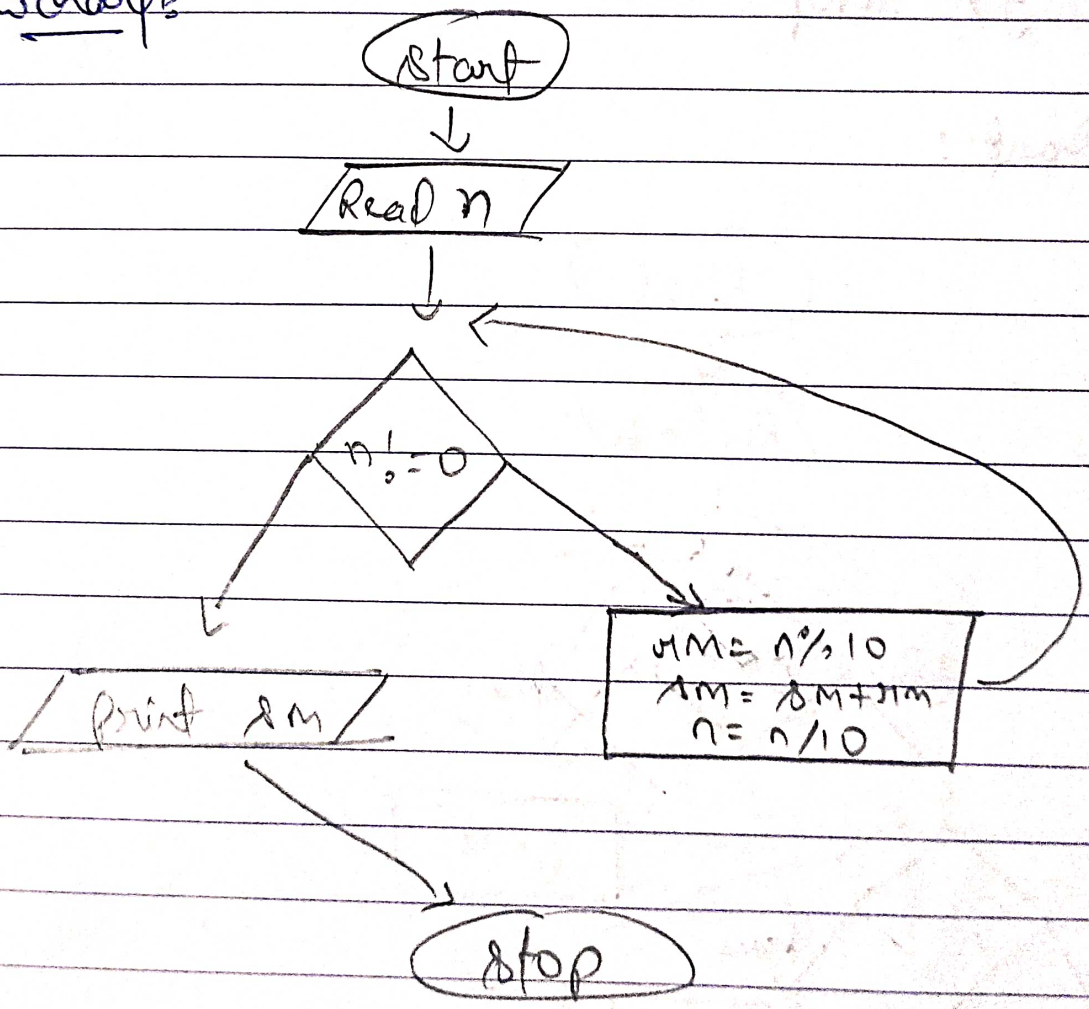
c) while (n != 0) do

RM = n % 10 and everytime SM = SM + RM, n = n / 10'

d) Print the value of 'SM' to get sum of all digit.

e) Stop.

Flowcharts

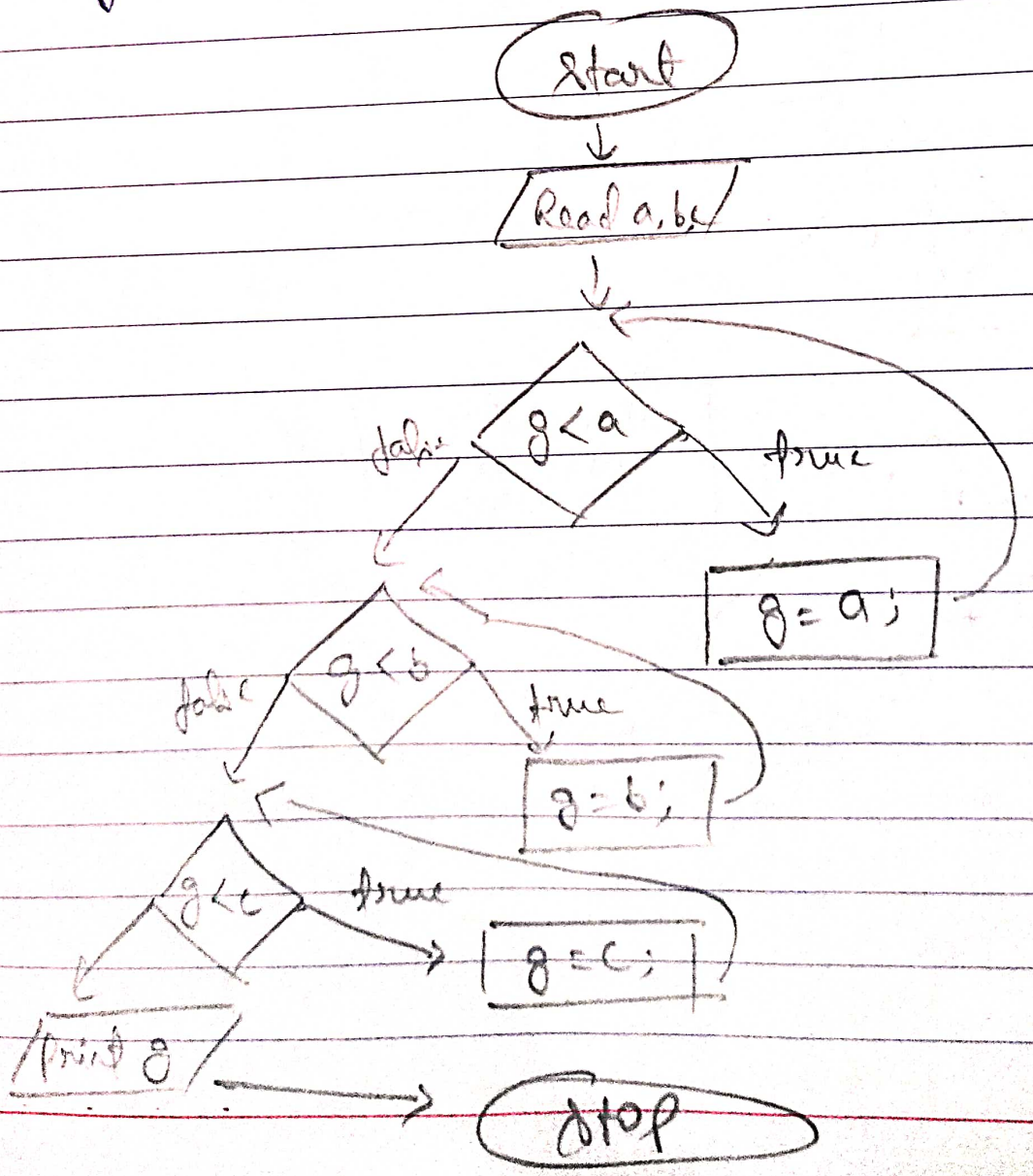


Q.11

Algo:

a) Read numbers a, b, c.
b) Define g = a;
c) Check if (g < a) then g = a; then if (g < b) then g = b, after if (g < c) g = c.
and then
d) Print the value of 'g' to get the smallest no. among the three.
d) Stop

flowchart:



Q-12

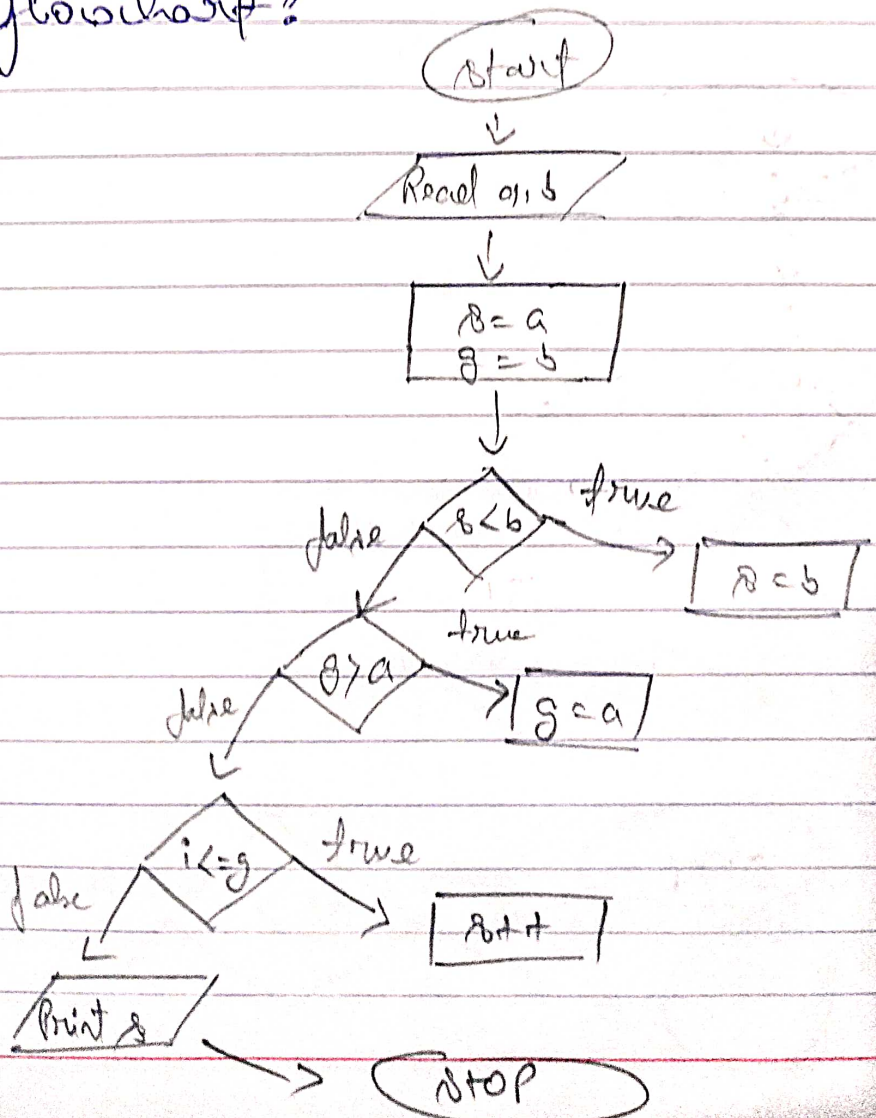
Algo:

a) Read numbers a, b
b) Define x = a, y = b
c) if (x < b) then x = b.
   if (y > a) then y = a.
d) now from 1 to y run a loop and increase x++.

```
for (i = 1 ; i <= y ; i++)
    x++;
```

e) Print the value of x to get sum.
f) stop

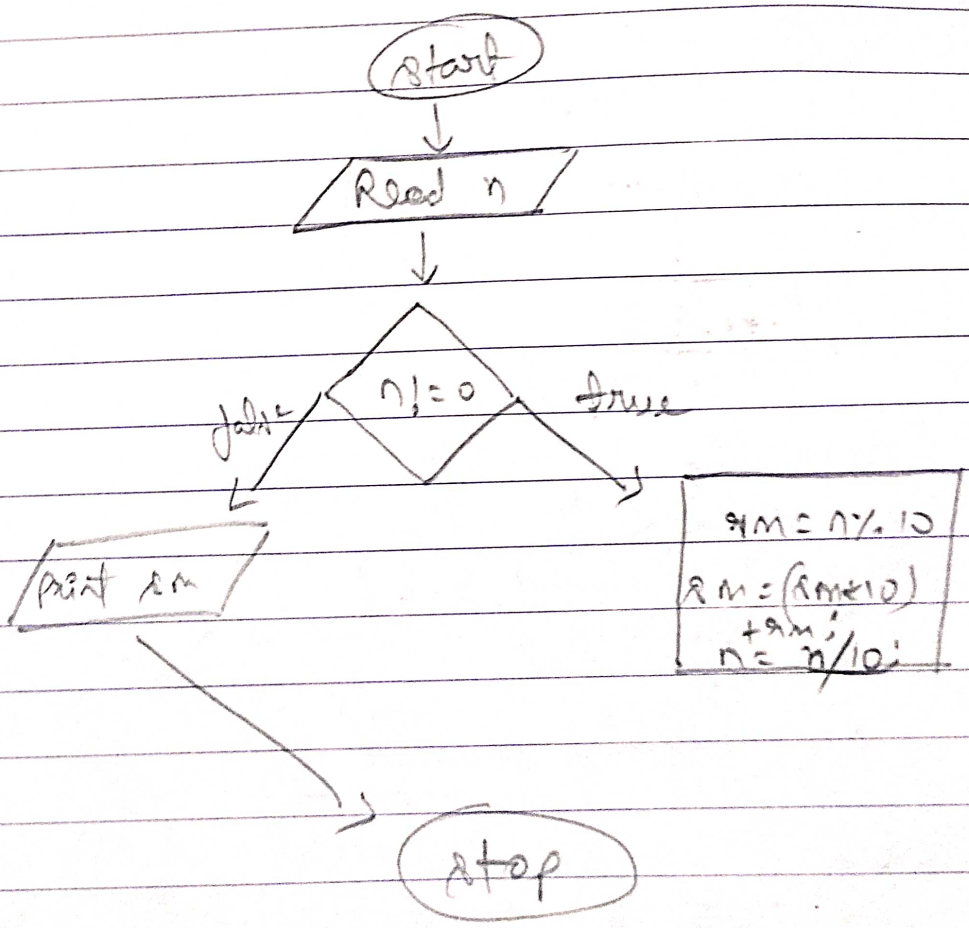
flowchart:



Q-13 Algo:-

a) Read the number n.
b) Define sm = 0 , rm = 0.
c) while (n != 0)
    rm = n % 10;
    sm = (sm * 10) + rm;
    n = n / 10;
d) Print the value of 'sm' to get reversed number.
e) Stop

Flowchart:



Q-14 Algo:

a> Read numbers a, b

b> Define gr = a , g = 0

c> If (gr < b) then gr = b.

d> for (i = 1 ; i < gr ; i++)

{ if (a i == a || i == b)

continue;

if (a % i == 0 && b % i == 0)

{ if (g < i)

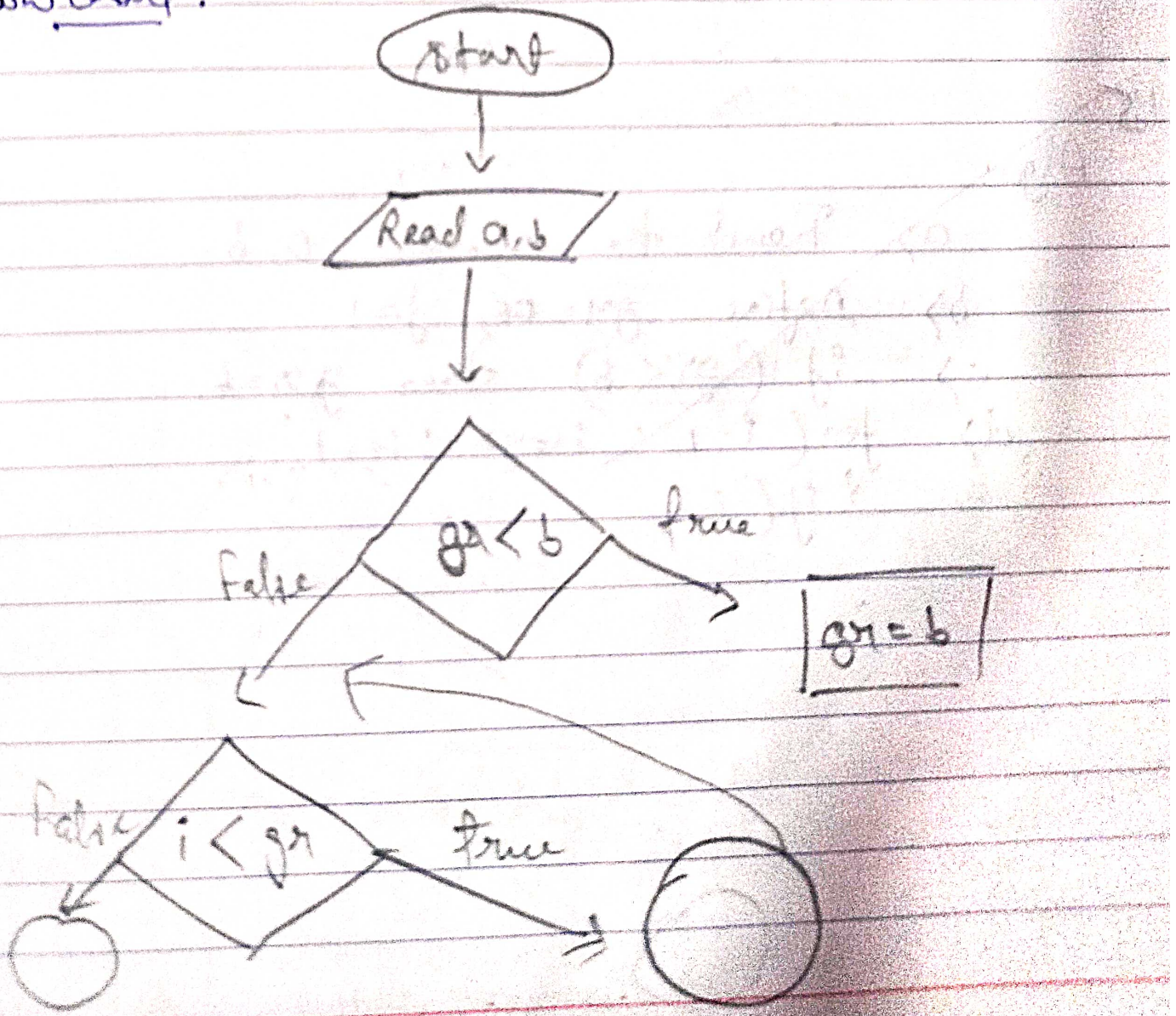
g = i ;

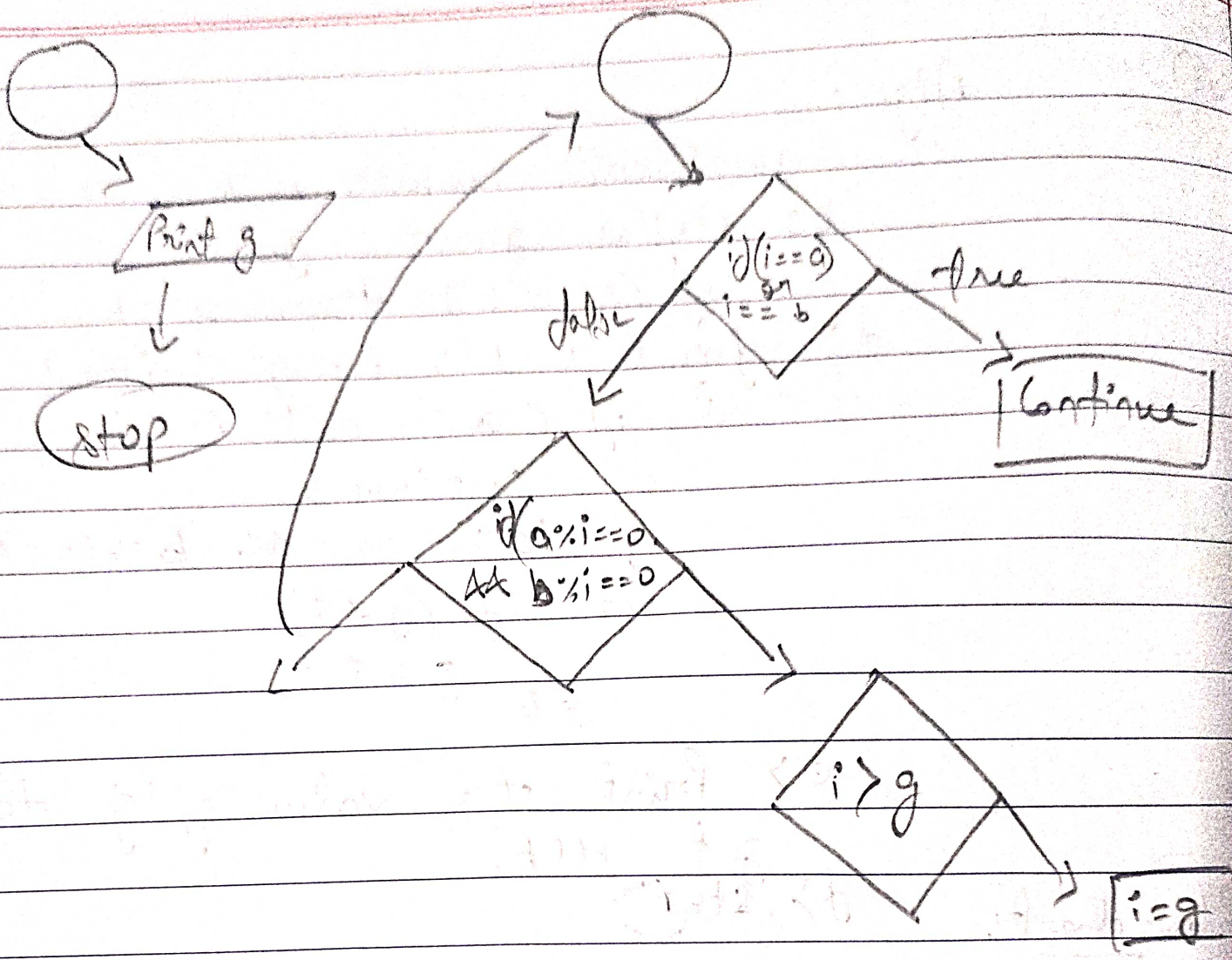
} }

e> Print the value of 'g' to get HCF.

f> Stop

Flowchart:-




Print g
stop
if(i==a
or
i==b
false
true
Continue
if(a%i==0
&& b%i==0
i>g
i=g

Q.17

Algo:

a) Read number 'n'.

b) Define sm = 0, rm = 0, n1 = n

c) while (n != 0)

rm = n % 10;

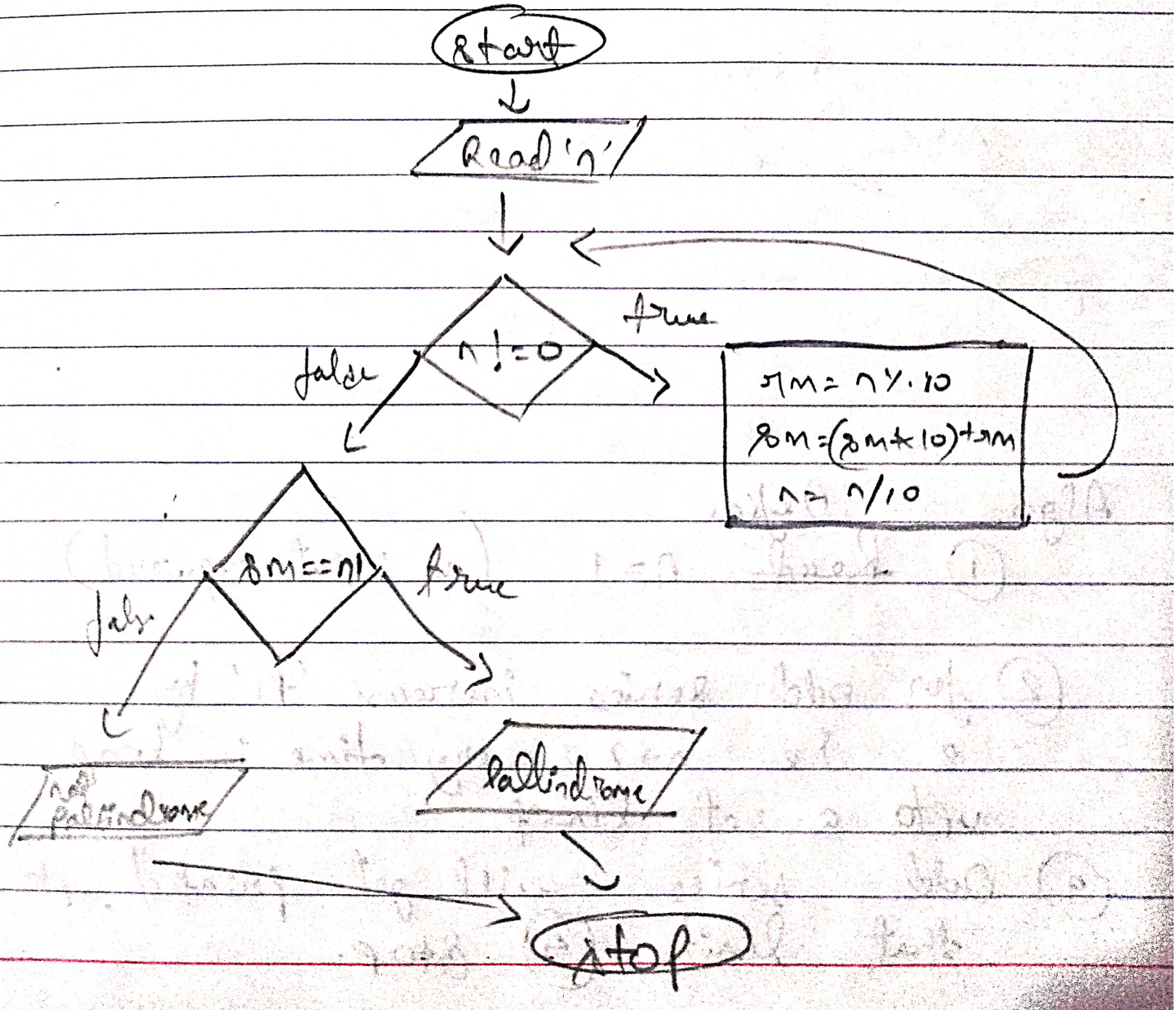
sm = (sm * 10) + rm;

n = n / 10;

d) if (sm == n1) print "palindrome"
else not palindrome.

e) stop.

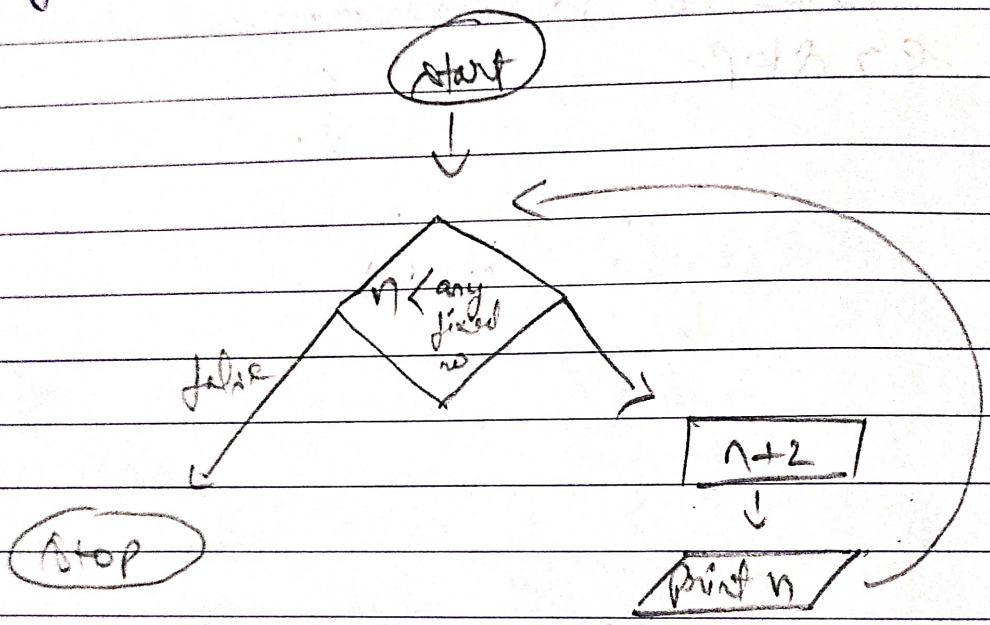
Flowchart



Q-19

Algo: Define
(1) Read n=0 (no input required)
(2) for even series increase n by 2, n+2 every time in loop upto a set limit
(3) Even series gets printed upto that limit (4) Stop.

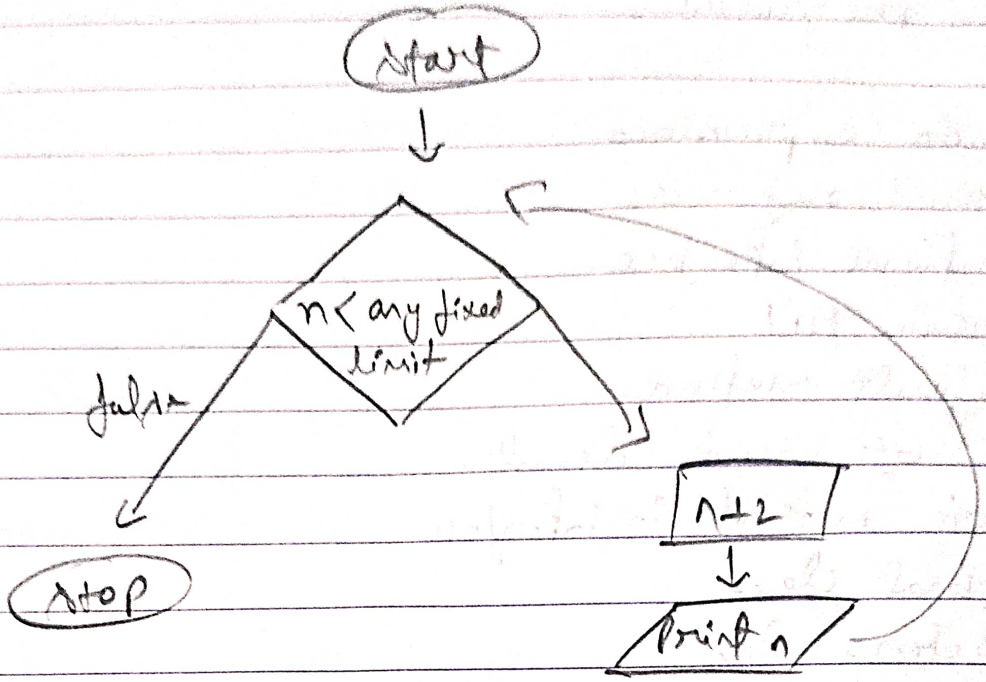
flowchart:



Q-20

Algo. Define
(1) ~~Read~~ n=1 (no input required)
(2) for odd series increase 'n' by 2 i.e. n+2 every time in loop upto a set limit.
(3) Odd series will get printed upto that limit (4) Stop.

Flowchart:-



Q.15

Algo:-

(1) Read the numbers a, b.

(2) Define max = a;

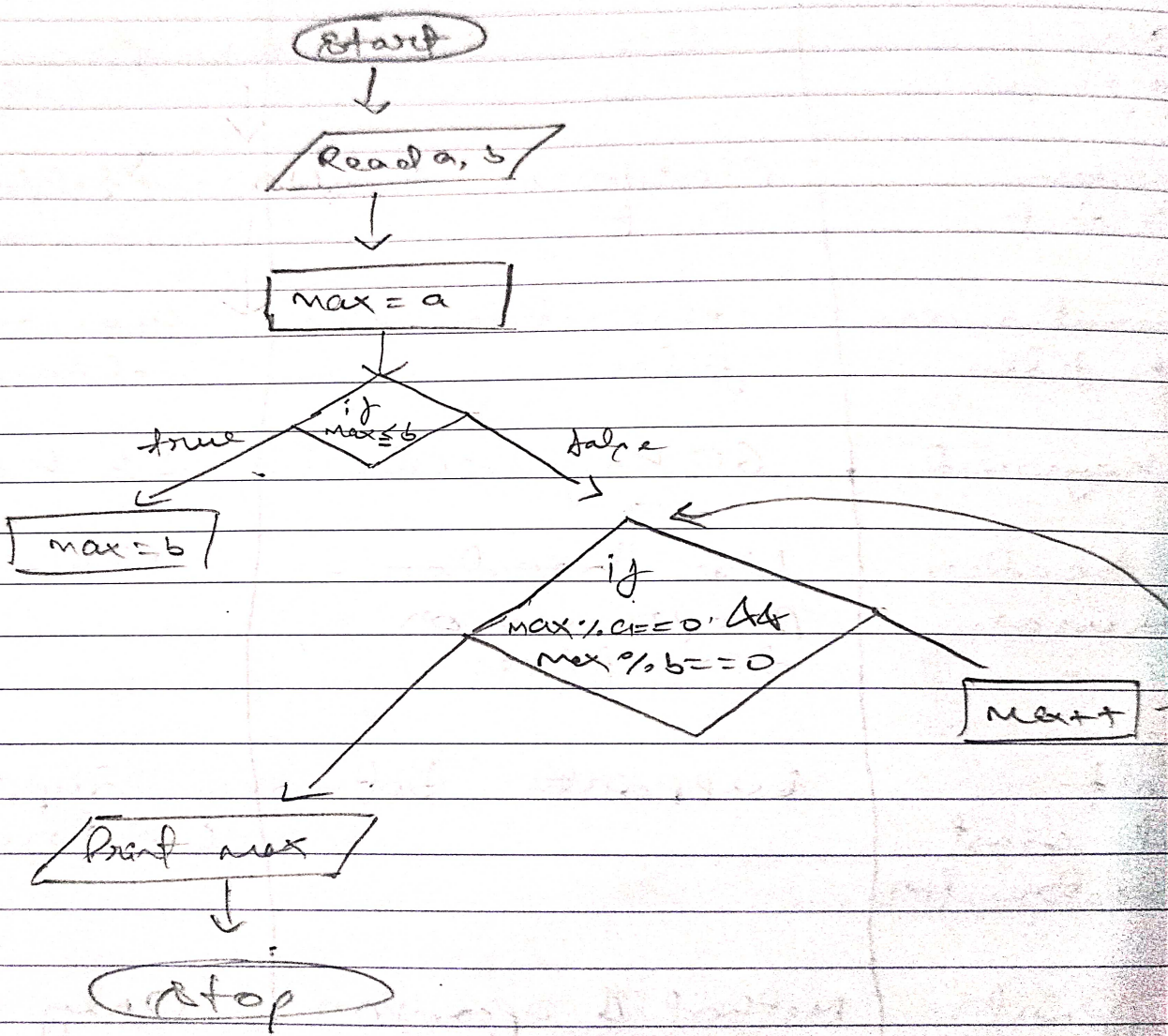
~~(3) If (max % a == 0 && max % b == 0)~~

(3) If (max < b)
max = b;

(4) Run a loop until (max % a == 0 && max % b == 0) increase max by 1 every time.

(5) If (n % a == 0 && max % b == 0) then 'max' will be our LCM.

(6) Stop.


Start
Read a, b
max = a
if max%b
true
false
max = b
if max%a==0 && max%b==0
max++
Print max
Stop