

IS  $f(x) = (2\sqrt{x} + 1)$   
Input

chosen ride  
height  
age

## Processing

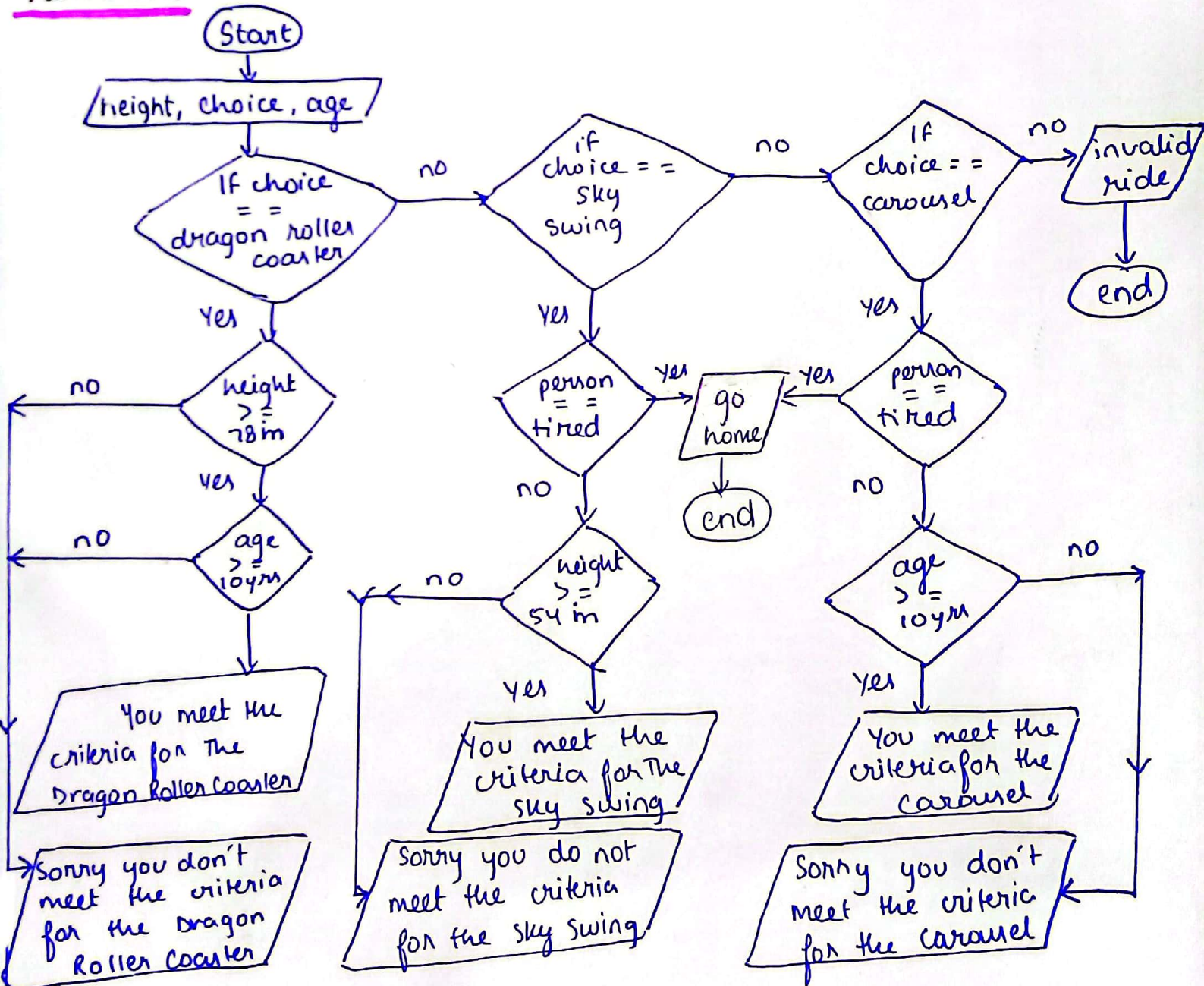
retrieve criteria  
check height against criteria  
check age against criteria  
determine if they meet the requirements. Also check if selected ride is valid & if person is tired.

## Output

eligibility status  
· eligible  
· not eligible

invalid ride

## Flowchart



## My Approach

check the ride first, then check the respective requirements. In 2nd & 3rd ride also check if the person's tired or not. If he doesn't choose one of these three rides, it's obviously an invalid ride.

## Pseudocode

Start

```
Declare ride choice as string
Declare height as float
Declare age as integer
Print("Enter the ride choice (Dragon Roller coaster/ Sky
      swing/ carousel):")
Input ride choice
Print("Enter the height of the person: ")
Input height
Print("Enter the age of the person: ")
Input age
If ride choice == Dragon Roller coaster
    If height >= 78 inches && age >= 10 yrs
        Print("You meet the criteria for Dragon Roller coaster")
    Else
        Print("Sorry you don't meet the criteria for Dragon Roller coaster")
Else If ride choice == sky swing
    If person == tired
        Print("go home")
    Else
        If height >= 54 inches
            Print("You meet the criteria for sky swing")
        Else
            Print("Sorry you don't meet the criteria for sky swing")
Else If ride choice == carousel
    If person == tired
        Print("go home")
    Else
        If age >= 10 yrs
            Print("You meet the criteria for carousel")
        Else
            Print("Sorry you don't meet the criteria for carousel")
Else
    Print("invalid ride choice")
End If
End
```



Input

7-digit binary number

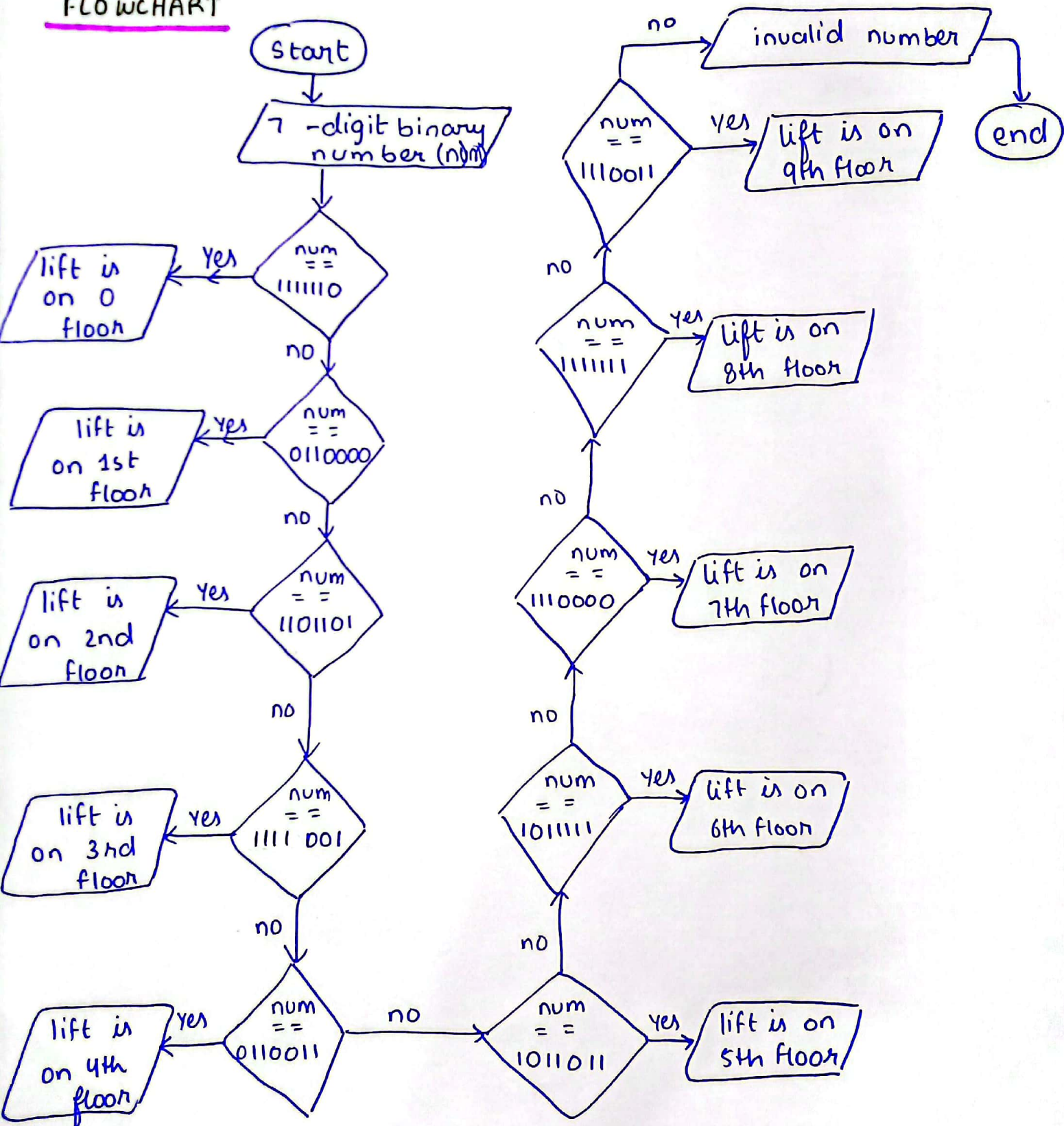
Processing

convert 7-digit binary number input into segments  
compare it with patterns

Output

number corresponding to floor number

FLOWCHART



## Pseudocode

Start

Declare num as integer

Print("Enter a 7 digit binary number:")

Input num

If num == 111110

Print("lift is on 0 floor.")

Else If num == 0110000

Print("lift is on 1st floor")

Else If num == 1101101

Print("lift is on 2nd floor")

Else If num == 1111001

Print("lift is on 3rd floor")

Else If num == 0110011

Print("lift is on 4th floor")

Else If num == 1011011

Print("lift is on 5th floor")

Else If num == 1011111

Print("lift is on 6th floor")

Else If num == 1110000

Print("lift is on 7th floor")

Else If num == 1111111

Print("lift is on 8th floor")

Else If num == 1110011

Print("lift is on 9th floor")

Else

Print("invalid number")

End If

End

## my Approach

first I checked the floor numbers & what sections it needed to light up then I checked if the binary 7 digit number corresponds to that

Input

Processing

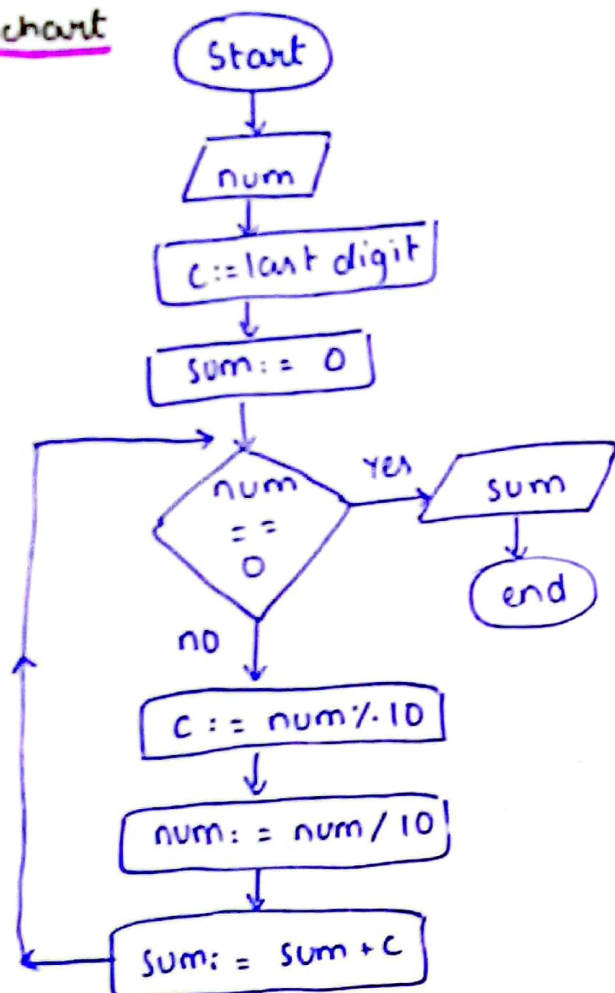
Output

a number

get last digit by using  
 $\%10$  & remaining digits by  
 using  $/10$ .

sum of digits  
 of the number

## Flowchart



## Pseudocode

Start

Declare num as integer  
 Declare sum of digits as integer  
 Set c as last digit  
 Set sum = 0  
 If num == 0  
   Print("sum")

Else

  c = num % 10  
   num =

## Pseudocode

Start

  set last digit to c  
   sum = 0

  while num != 0

    set c to num % 10

    set sum to sum + c

    set num to num / 10

  Print("sum")

End

My Approach

Take a num & last digit equal to c & sum to 0. If num is not equal to 0 set c to num % 10, num to num / 10 & sum to sum + c and continue this until num equals 0, after that print sum & end it.

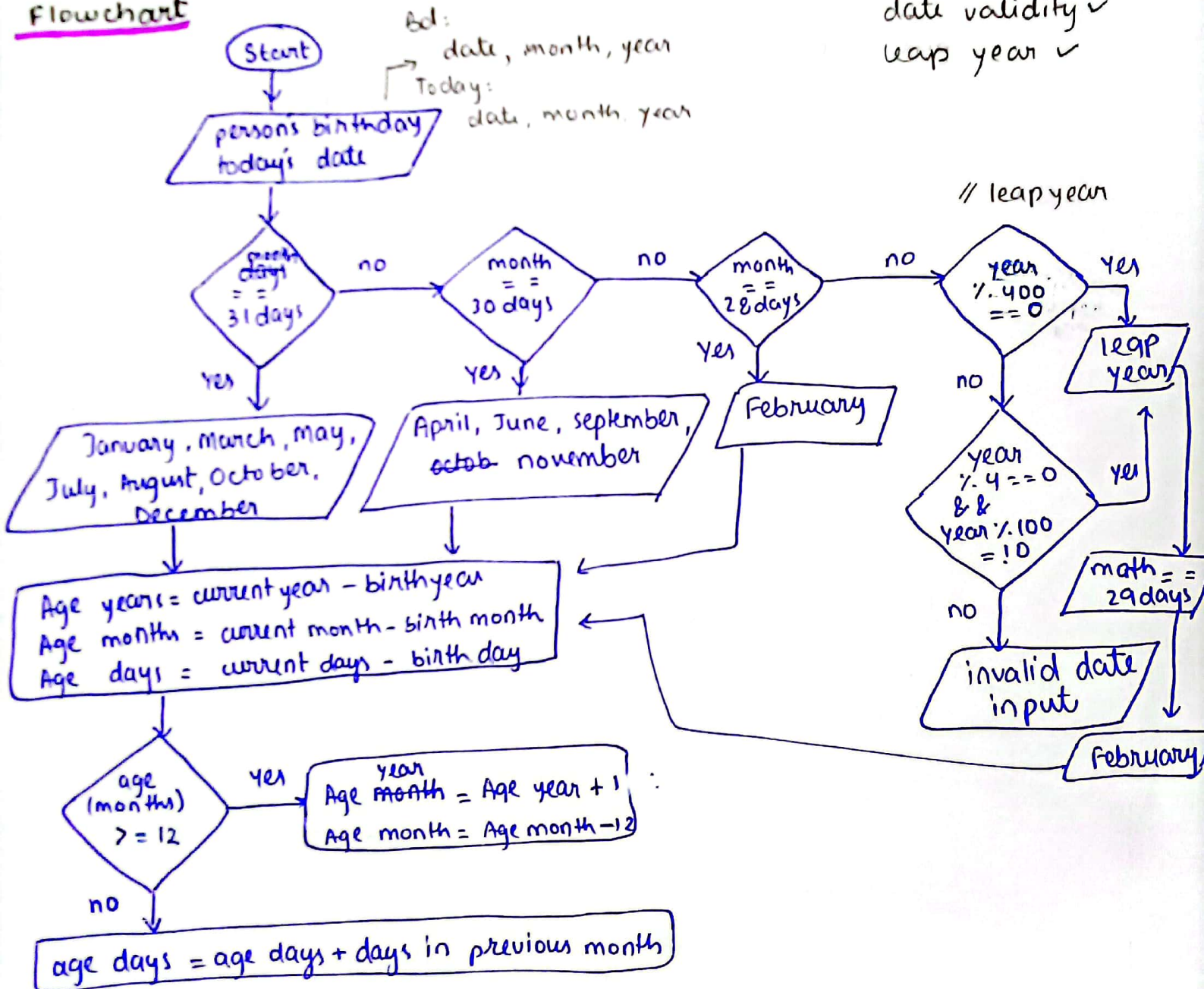


Input

person's birthday  
Today's date

Processing  
check the validity  
of date & then  
 $\text{age} = \text{today's date} - \text{person's birthday}$

Output  
exact age

Flowchart

Pseudocode

Start

Input birth year, birth month, birth day

Input current year, current month, current day

Age years = current year - birth year

Age month = current month - birth month

Age days = current day - birth day

If age (months)  $\geq 12$  then

    age (years) = age (years) + 1

    age month = age month  $\div 12$

Else If age days  $< 0$  then

    age month = age month - 1

Else If current month = 1 || then 3 || 5 || 7 || 8 || 10 || 12

    print (days in previous month = 31)

Else if current month = 2

    If current year % 400 == 0 || current year % 4 == 0 ||  
        current year % 100 != 0 then

        print (days in previous month = 29)

    else print (days in previous month = 28)

Else print (days in previous month = 30)

age days = age days + days in previous month

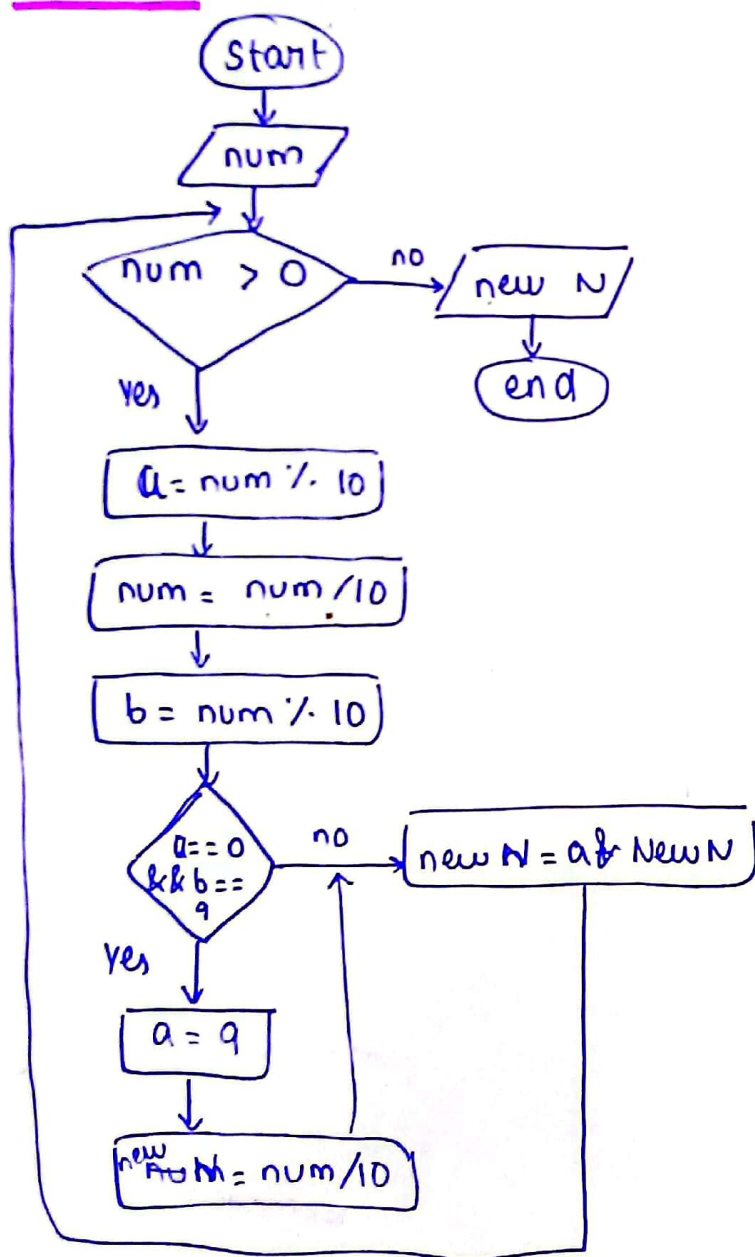
Input  
number

Processing

If  $\text{num} > 0$ , get last digit by  $\text{num} \% 10$  and divide  $\text{num}/10$  do it again to know 2nd last digit. If last digit is 0 & second last is 9 then initialize last digit with 9 to replace 0 add this to new string.

Output  
corrected  
number

Flow chart



Pseudocode

Start

Input num

while num > 0

$a = \text{num} \% 10$

$\text{num} = \text{num} / 10$

$b = \text{num} \% 10$

If  $a == 0 \ \& \ b == 9$

$a = 9$

$\text{new N} = \text{num} / 10$

End If

$\text{New N} = \text{new N} \ \& \ A$

End while

Output new N

End

My Approach

First I'll check if last digit is 0 and second last is 9, if that's the case I'll divide it by 10 to remove 0 so that the number can be corrected.

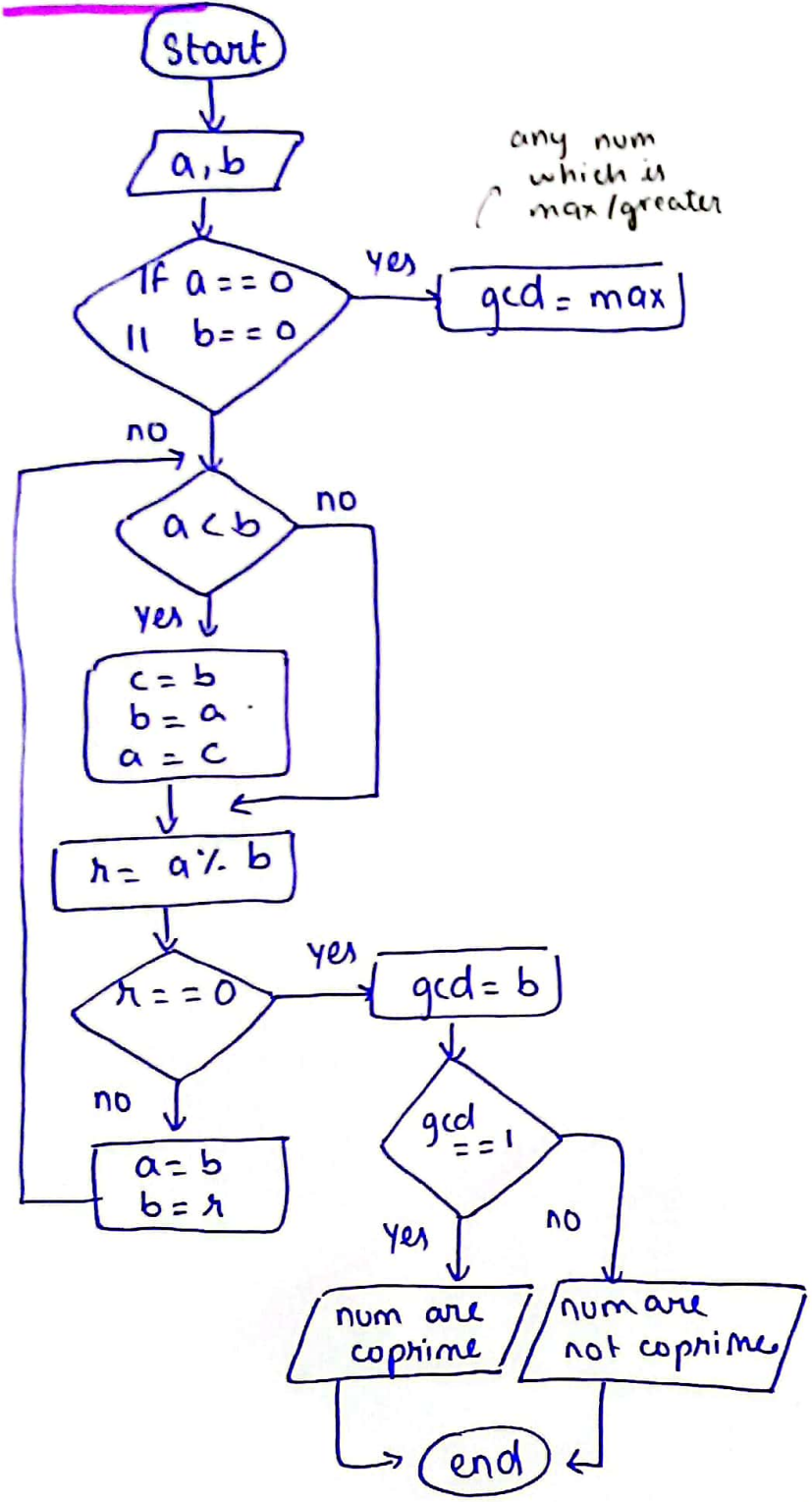


**Input**  
2 integers  
greater than 0

**Processing**  
calculate gcd of 2 num.  
If its 1, they're coprime

**Output**  
coprime?  
(yes/no)

Flowchart



Pseudocode

Start  
Input a  
Input b  
If  $a == 0$  ||  $b == 0$  then  
    gcd = max  
Else  
    while  $a < b$  then  
        c = b  
        b = a  
        a = c  
        r =  $a \% b$   
        If  $r == 0$  then  
            gcd = b  
            If gcd == 1  
                print("coprime")  
            Else  
                print("not coprime")  
        Else  
            a = b  
            b = r  
    End while  
End

**My Approach:-**  
2 num are coprime when their gcd = 1, we just need to take 2 numbers, find their gcd if  $a < b$  so we need to swap it to make a the greater num & if one of them is zero, gcd is the max number of them.

Input

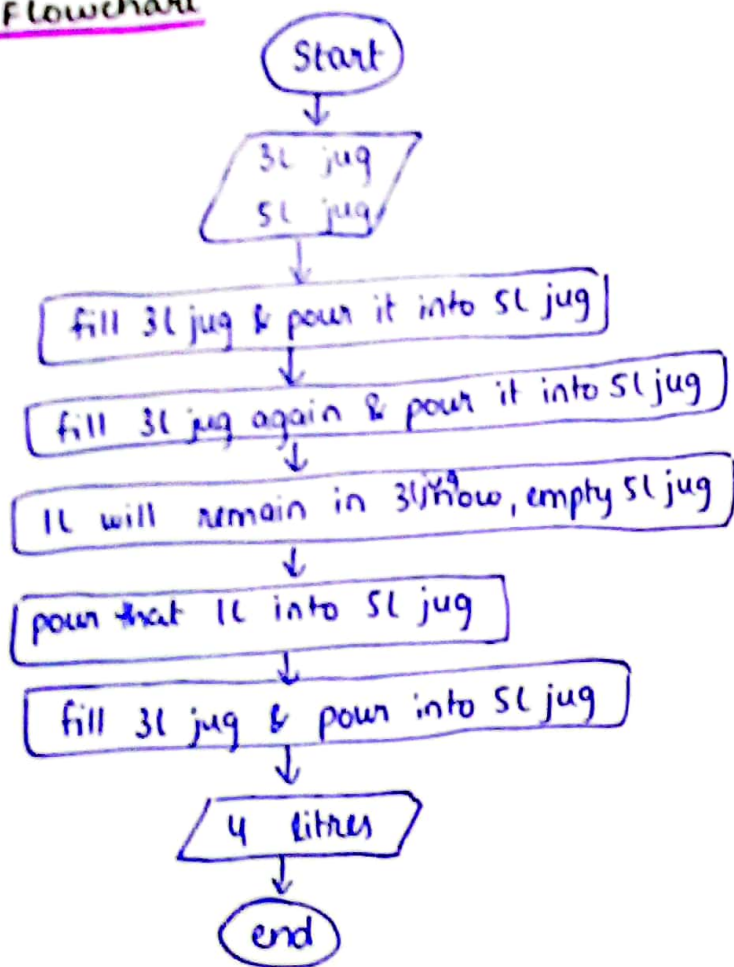
3L jug  
5L jug

Processing

fill 3, pour into 5  
fill 3 again, pour into 5  
1L will remain in 3 now. empty  
5L jug and fill it with 1L from 3 then  
fill 3 completely again & pour into 5  
& then you've exact 4L of water.

output  
exact  
4L of water

Flowchart



Pseudocode

Start  
Input 3L jug (3L)  
Input 5L jug (5L)  
full = 3  
empty = 0  
3Lj = 3  
5Lj = 3Lj  
3Lj = 0  
3Lj = 3  
5Lj = 3  
3Lj = 1  
5Lj = 0  
5Lj = 3Lj 1 of 3Lj  
3Lj = 3  
5Lj = 3  
output 4L in 5Lj  
End

My Approach

fill 3Lj & empty in 5Lj  
repeat and you'll be left with  
1L in 3Lj now empty out 5Lj  
and pour 1L from 3Lj to 5Lj. fill  
3Lj again and pour into 5Lj and  
you'll have exact 4 litres of water.



Input

m litre jug  
n litre jug  
unlimited supply  
of water

Processing

determine amount to measure  
find gcd of m & n, if  
gcd == 1, all amounts are  
possible less than m (if  
m is greater jug)

Output

desired amount  
of water measured  
using m & n  
jugs

Pseudo code

Start

jug A, jug B, amount = 0

Input n, m, x

If (x > m && x > n)

Print("not possible")

End If

while jug A != x && jug B != x do

If jug A == 0 then

jug A = m

Print("fill jug A with m litres")

Else If jug A + jug B ≤ n then

jug B = jug B + jug A

jug A = 0

Print("pour all water from jug A to jug B")

Else

amount = n - jug B

jug A = jug A - amount

jug B = n

Print("pour amount from jug A to jug B")

End If

If jug B == n then

jug B = 0

Print("empty jug B")

End If

End while

\*

\* If jug A == x || jug B == x then

Print("x litres extracted")

Print("jug A and jug B")

End If

End

My Approach

If gcd of m & n is 1, any amount can be extracted which is greater than m (~~max~~ ~~water jug~~) & lesser than n or vice versa if m is the greater jug.

Q13

Flowchart