LAB 02

Task# 01:

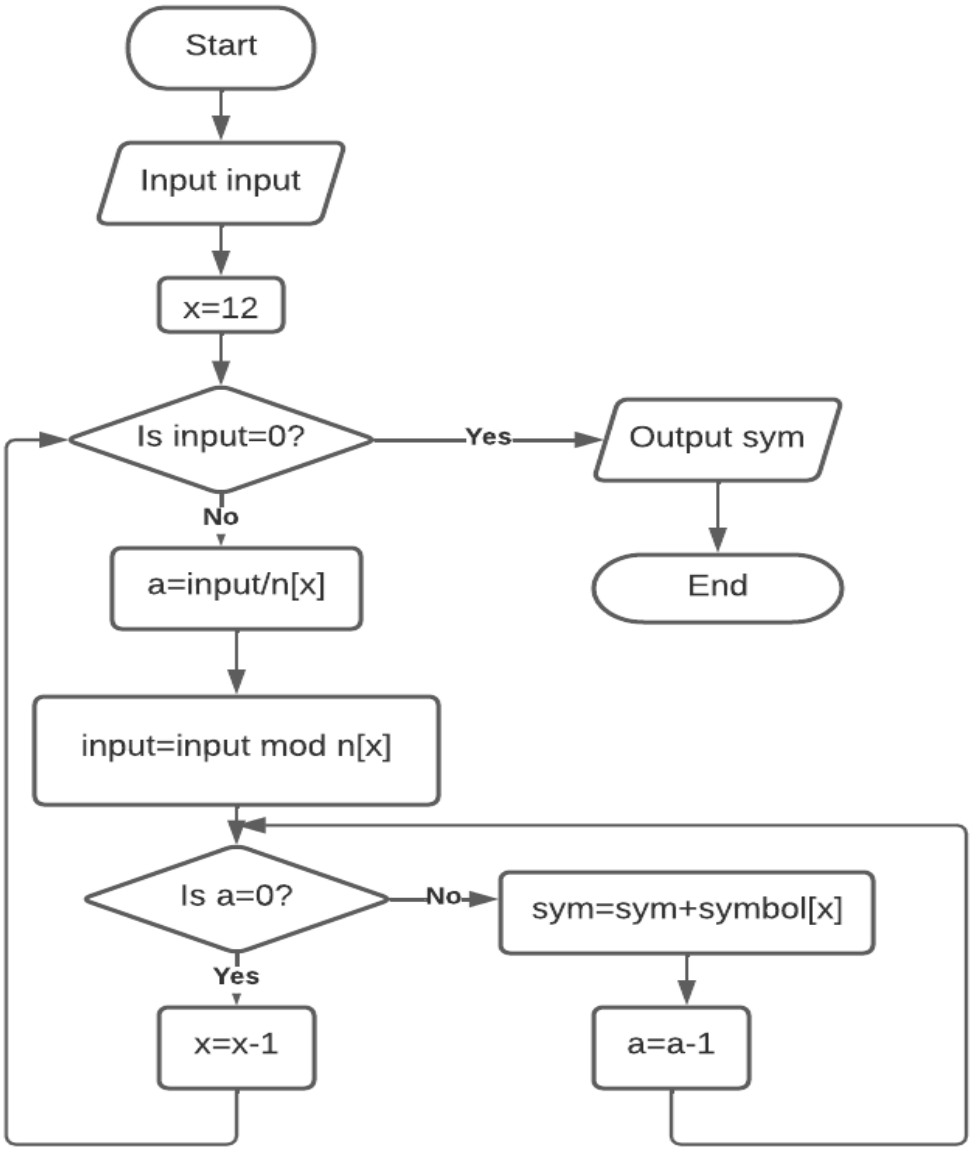
Write a program that converts a positive integer into the Roman number system. The Roman number system has digits I (1), V (5), X (10), L (50), C(100), D(500) and M(1000). Numbers up to 3999 are formed according to the following rules: a) As in the decimal system, the thousands, hundreds, tens and ones are expressed separately. b) The numbers 1 to 9 are expressed as: 1 I 6 VI 2 II 7 VII 3 III 8 VIII4 IV 9 IX 5 V (An I preceding a V or X is subtracted from the value, and there cannot be more than threeI's in a row.) c) Tens and hundreds are done the same way, except that the letters X, L, C, and C, D, Mare used instead of I, V, X respectively. Example: Your program should take an input, such as 1978, and convert it to Roman numerals, MCMLXXVIII.

Pseudocode:

```
Function RC(input : Integer)
  sym=""
  n[0]=1
  n[1]=4
  n[2]=5
  n[3]=9
  n[4]=10
  n[5]=40
  n[6]=50
  n[7]=90
  n[8]=100
  n[9]=400
  n[10]=500
  n[11]=900
  n[12]=1000
  symbol[0]="I"
  symbol[1]"IV"
  symbol[2]"V"
  symbol[3]"IX"
  symbol[4]"X"
  symbol[5]"XL"
  symbol[6]"L"
  symbol[7]"XC"
  symbol[8]"C"
  symbol[9]"CD"
  symbol[10]"D"
  symbol[11]"CM"
  symbol[12]"M"
  x=12
  While (input>0)
    a=input / n[x]
    input =input% n[x]
    While (a>0)
      sym=sym+symbol[x]
      a=a-1
    EndWhile
    x=x-1
  EndWhile
  return sym

Input(a)
sym=RC(a)
Output(sym)
```

Flowchart:


Start
Input input
x=12
Is input=0?
Yes
Output sym
No
End
a=input/n[x]
input=input mod n[x]
Is a=0?
No
sym=sym+symbol[x]
Yes
x=x-1
a=a-1

Code:

```
def printRoman(number):
    num = [1, 4, 5, 9, 10, 40, 50, 90,
        100, 400, 500, 900, 1000]
    sym = ["I", "IV", "V", "IX", "X", "XL",
        "L", "XC", "C", "CD", "D", "CM", "M"]
    i = 12

    while number:
        div = number // num[i]
        number %= num[i]

        while div:
            print(sym[i], end = "")
            div -= 1
        i -= 1



number = 1978
print("Roman value is:", end = " ")
printRoman(number)
```

```
Roman value is: MCMLXXVIII
```

Task# 02:

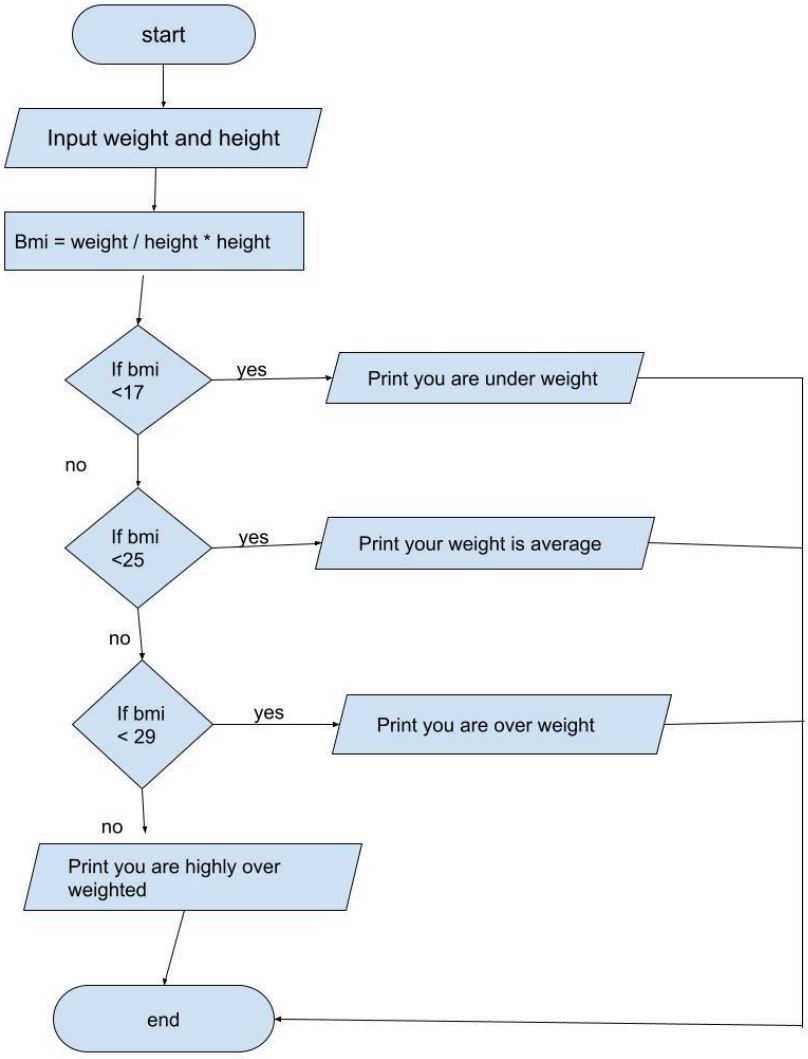
Write a program that calculates the user's body mass index (BMI) and classify it as underweight, normal, overweight, or obese, based on the table from the United States Centers for Disease Control.

Pseudocode:

Input(Mass)
Input(Height)
bmi=Mass/(Height*Height)
If bmi<=17.0 Then
  Output("Underweight")

```
ElseIf (bmi>17.0)&(bmi<=24.0) Then
   Output("Healthy Weight")
ElseIf (bmi>24.0)&(bmi<=29.0) Then
   Output("OverWeight")
Elseif (bmi>29.0)&(bmi<=39.0) Then
   Output("Obese")
Else
   Output("Severly Obese")
End If
```

Flowchart:



Code:

```
weight = float(input("enter weight in kg : "))
height = float(input("enter height in m : "))
bmi = weight / (height*height)
if (bmi<17):
    print(" you are under weight ")
elif(bmi>17) & (bmi<= 24):
    print("your weight is at best")
elif(bmi>24) & (bmi<= 29):
    print("you are over weight")
else:
    print("you are highly over weighted / obese")
```

```
enter weight in kg : 62
enter height in m : 1.828
your weight is at best
```

Task # 03:

Write a program to compute quotient and remainder of a number without using division ('/') operator and modulo ('%') operator. Also mention procedure for calculating

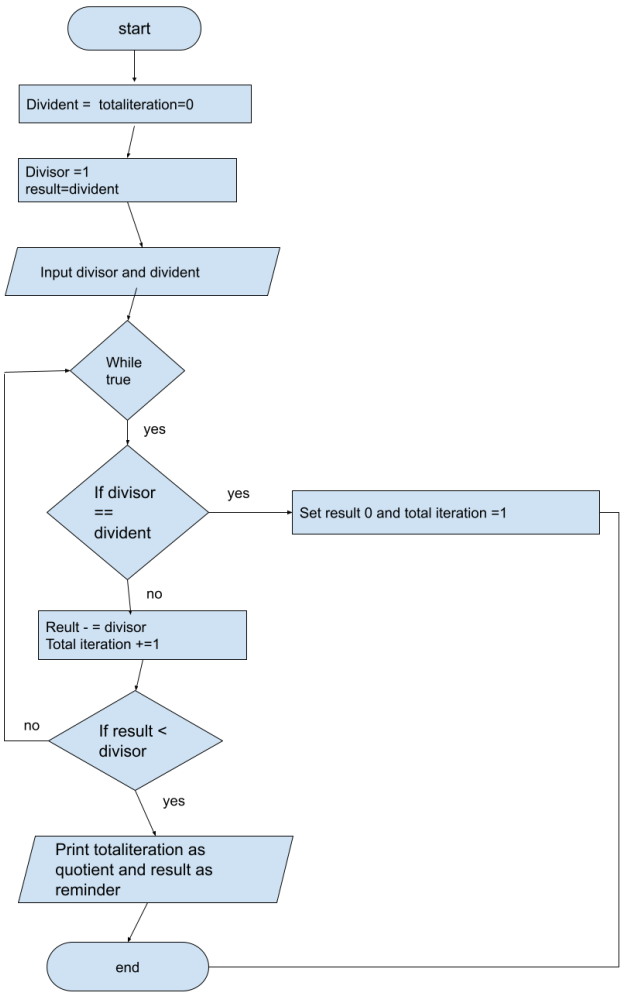
Pseudocode:

```
divisor=1
divident=0
result=divident
iterations=0
c=0

While(c=0)
  Input(divisor)
  Input(divident)
  If(divident>=divisor) Then
    break
  EndIf
EndWhile
```

```
While(c==0)
   If (divident==divisor) Then
      result=0
      iterations=1
      break
   Else
      result=result-divisor
      iterations=iterations+1
      If(result<divisor) Then
         break
      End If
   End If
EndWhile
Output(iterations)
Output(result)
```

Flowchart:



Code:

```
divisor=1
divident=0
result=divident
totalIteration=0
z=0

while(z==0):
    divisor= int(input("enter divisor : "))
    divident = int(input("enter divident : "))
    if(divident<divisor):
        continue
    else:
        break
while(z == 0):
    if(divident == divisor):
        result=0
        totalIteration=1
        break
    else:
        result= result - divisor
        totalIteration=totalIteration+1
        if(result<divisor):
            break


print("quotient is : " + str(totalIteration))
print("reminder is : " + str(result))
```

```
enter divisor : 2
enter divident : 2
quotient is : 1
reminder is : 0
```