

#Amstrong number

```
def Amstrong(number):  
    n=number  
    Sum=0  
    while(n!=0):  
        num=n%10  
        Sum=Sum+num*num*num  
        n=n//10  
    return Sum  
number=int(input("Enter Number: "))  
if(number==Amstrong(number)):  
    print(number,"is AMSTRONG number")  
else:  
    print(number,"is not AMSTRONG number")
```

#SUM of Natural number

```
def natural(num):  
    Sum=0  
    for i in range (num):  
        Sum=Sum+i  
    return Sum  
num=int(input("Enter number: "))  
print(natural(num)," is the sum of",num,"Natural number")
```

#LCM

```
def gcd(a,b):
```

```
if(a<b):  
    if (a==0):  
        return b  
    else:  
        return gcd(b%a,a)  
else:  
    if(b==0):  
        return a  
    else:  
        return gcd(a%b,b)  
  
def lcm(a,b):  
    lcm=a*b/gcd(a,b)  
    return lcm  
  
a,b=(input("Enter two number by spacing: ").split())  
a,b=[int (a),int (b)]  
print("Lcm of ",a,"&",b,"is",lcm(a,b))
```

#Reverse NUMBER

```
def reverse(num):  
    n=num  
    Sum=0  
    while(n!=0):
```

```
num=n%10

Sum=Sum*10+num

n=n//10

return Sum

num=int(input("Enter the number:"))

print("Reverse of",num,"is",reverse(num))
```

#Swaping of Number

```
def Swap(a,b):

    l=[]

    l.append(b)

    l.append(a)

    return l

a,b=input("enter nubers:").split()

a,b=[int (a), int(b)]

print("Swap of",a,b,"is",*Swap(a,b))
```

#Chicken & Rabbit

```
def solve(heads,legs):

    error_msg="No solution"

    chicken_count=0

    rabbit_count=0


    if(legs%2==0):
```

```
chicken_count=(legs/2)-legs+2*heads
rabbit_count=(legs-2*heads)/2
print(chicken_count,rabbit_count)
```

```
else:
    print(error_msg)
return 0
```

```
solve(150,400)
```

#Pallindrom

```
def pallindrom(num):
    temp=num
    add=0
    while (temp!=0):
        number=temp%10
        add=add*10+number
        temp=temp//10
    return add
```

```
num=int(input("Enter number: "))
if(num==pallindrom(num)):
    print(num,"is Pallindrom")
else:
    print(num,"is not Pallindrom")
```

#Perfect number

```
def perfect_no(num):
    a=0
```

```
for i in range (1,num):
    if (num%i==0):
        a=a+i

return a

num=int(input("Enter number:"))

if (num==perfect_no(num)):
    print(num,"is PERFECT number")
else:
    print(num,"is not PERFECT number")
```

#Strong number

```
def fact(number):
    fact=1
    while(number!=0):
        fact=fact*number
        number=number-1
    return fact
```

```
def strong(num):
    temp=num
    a=0
    while(temp!=0):
        number=temp%10
        a=a+fact(number)
        temp=temp//10
```

```
return a
```

```
num=int(input("Enter Number to check:"))
```

```
if (num==strong(num)):
```

```
    print(num,"is STRONG number")
```

```
else:
```

```
    print(num,"is not STRONG number")
```

Factorial number

```
def fcto(num):
```

```
    fact=1
```

```
    for i in range (num):
```

```
        fact=fact*num
```

```
        num=num-1
```

```
    return fact
```

```
num=int(input("Enter the number: "))
```

```
print(fcto(num))
```

#Fibonacci Series up to n number of terms

```
def fabbonaci_series(num):
```

```
    n1=0
```

```
    n2=1
```

```
    next_num=n1+n2
```

```
    for i in range (0,num):
```

```
        print(next_num)
```

```
n1=n2  
n2=next_num  
next_num=n1+n2
```

```
num=int(input("Enter nth number of series" ))  
fabbonaci_series(num)
```

#Fibonacci Sequence Up to a Certain Number"""

```
def fibonacci_series(num):  
    n1=0  
    n2=1  
    next_num=n1+n2  
    while(num>=next_num):  
  
        print(next_num)  
        n1=n2  
        n2=next_num  
        next_num=n1+n2
```

```
num=int(input("Enter number upto you want series: "))  
fibonacci_series(num)
```

#Fibonacci Sequence starting from 100

```
def fabbonaci_series(num):  
    n1=0  
    n2=99  
    next_num=100  
    for i in range (0,num):
```

```
print(next_num)

n1=n2

n2=next_num

next_num=n1+n2
```

```
num=int(input("Enter nth number of series" ))

fabbonaci_series(num)
```

#Leap yr :1

```
def leap_yr(yr):

    if(yr%4==0 ):

        if(yr%100==0):

            if(yr%400==0):

                print("leap")

            else:

                print("not leap")

        else:

            print(" leap")

    else:

        print("not leap")

yr=int(input("Enter the year which is to be check: "))

leap_yr(yr)
```

#Leap yr :2

```
def leap_yr(yr):

    if (( year%400 == 0)or (( year%4 == 0 ) and ( year%100 != 0))):

        print(yr,"is a Leap Year")

    else:

        print(yr,"is Not the Leap Year")

yr=int(input("Enter the number:"))

leap_yr(yr)
```


#prime number

```
n=int(input("Enter the number:-"))
```

```
for i in range(2,n//2):
```

```
    if(n%i==0):
```

```
        a=1
```

```
        break;
```

```
if(a==0):
```

```
    print(n,"is Prime number")
```

```
else:
```

```
    print(n,"is not prime number")
```

Q)

Jack and his three friends have decided to go for a trip by sharing the expenses of the fuel equally.

Write a Python program to calculate the amount (in Rs) each of them need to put in for the complete (both to and fro) journey.

The program should also display True, if the amount to be paid by each person is divisible by 5, otherwise it should display False. (**Hint:** Use the relational operators in print statement.)

Assume that mileage of the vehicle, amount per litre of fuel and distance for one way are given.

Test your code by using the given sample inputs.

Verify your code by using the 2nd sample input(highlighted) given below:

Sample Input			Expected Output
Mileage of the vehicle (km/litre of fuel)	Amount per litre of fuel (Rs)	Distance for one way (kms)	
12	65	96	260.0 True
12	40	190	

Note: You have to always submit your assignments/exercises!!

Estimated time: 30 minutes

```
mileage=12
```

```
amount_per_litre=40
```

```
distance_one_way=190
```

```
per_head_cost=0
```

```
divisible_by_five=False
```

```
per_head_cost=(distance_one_way*2/mileage*amount_per_litre)/4
```

```
if(per_head_cost%5 == 0):
```

```

divisible_by_five=True

else:

    divisible_by_five=False

print(per_head_cost)

print(divisible_by_five)

```

Q)

Question	Submission
----------	------------

Write a Python program to calculate and display the interest on a loan amount (Rupees) using the formula:
 $\text{interest} = (\text{principal} * \text{rate of interest} * \text{time}) / 100$

Test your code by using the given sample inputs.

Verify your code by using the 2nd sample input(highlighted) given below:

Sample Input			Expected Output
Principal	Rate of Interest	Time	
20000	5	10	10000.0
7800	7.7	26	

Note: You have to always submit your assignments/exercises!!

Estimated time: 15 minutes

```

principal=7800

rate_of_interest=7.7

time=26

interest=0

interest=(principal * rate_of_interest * time)/100

print(interest)

```

Q)

Write a python program to find and display the product of three positive integer values based on the rule mentioned below:

It should display the product of the three values except when one of the integer value is 7. In that case, 7 should not be included in the product and the values to its left also should not be included.

If there is only one value to be considered, display that value itself. If no values can be included in the product, display -1.

Note: Assume that if 7 is one of the positive integer values, then it will occur only once. Refer the sample I/O given below.

Sample Input	Expected Output
1, 5, 3	15
3, 7, 8	8
7, 4, 3	12
1, 5, 7	-1

Estimated Time: 20 minutes

-*- coding: utf-8 -*-

''''

Created on Fri Jun 14 16:14:32 2019

@author: Madhur

''''

#PF-Assgn-15

```
def find_product(num1,num2,num3):
```

```
    product=0
```

```
    #write your logic here
```

```
    if(num3%7==0 and num3!=0):
```

```
        product=(-1)
```

```
    elif(num2%7==0 and num2!=0):
```

```
        product=num3
```

```
    elif(num1%7==0 and num1!=0):
```

```
        product=num2*num3
```

```
    else:
```

```
        product=num1*num2*num3
```

```
    return product
```

#Provide different values for num1, num2, num3 and test your program

```
product=find_product(7,1,9)
```

```
print(product)
```