PROGRAM: Check if a number is Perfect

Code:

```
print("Code to check if a number is Perfect ")
def check_no(x):
   sum=0
   for i in range(1,x):
      if(x%i==0):
         sum=sum+i
   if sum==x:
      print(f"{x} is a Perfect number.")
   else:
      print(f"{x} is Not a perfect number")

a=int(input("Enter a number "))
check_no(a)
```

OUTPUT:

```
Code to check if a number is Perfect
Enter a number 8128
8128 is a Perfect number.

Code to check if a number is Perfect
Enter a number 225
225 is Not a perfect number
```

PROGRAM: Check if a String is Pangram

Code:

```
print("Code to check if a String is Pangram")
def checkpangram(str):
   flag=True
   alphabet="abcdefghijklmnopqrstuvwxyz"
   for char in alphabet:
```

```
        if char not in str:
            flag=False
    if(flag==True):
        print("Given sentence is a Pangram")
    else:
        print("Given sentence is NOT a Pangram")

sentence=(input("Enter a string "))
sentence=sentence.lower()
checkpangram(sentence)
```

OUTPUT:

```
Code to check if a String is Pangram
Enter a string The Quick Brown Fox Jumps Over The Lazy Dog
Given sentence is a Pangram

Code to check if a String is Pangram
Enter a string Hello people
Given sentence is NOT a Pangram
```

PROGRAM: To built a calculator using variable length arguments

Code:

```
print("Code to built a variable length arguments")
def addition(*args):
    sum=0
    for num in args:
        sum=sum+num
    return sum

print(f"The sum of 6 is {addition(6)}")
print(f"The sum of 55,20 is {addition(55,20)}")
```

```
print(f"The sum of 6,6,7,8,1,2,1 is {addition(6,6,7,8,1,2,1)}")
```

OUTPUT:

```
Code to built a variable length arguments
The sum of 6 is 6
The sum of 55,20 is 75
The sum of 6,6,7,8,1,2,1 is 31
```

```
def subtraction(*args):
    x=args[0]
    for num in range(1,len(args)):
        x=x-args[num]
    return x

print(f"The difference of 6 is {subtraction(6)}")
print(f"The difference of 55,20 is {subtraction(55,20)}")
print(f"The difference of 55,25,30 is {subtraction(55,25,30)}")
```

OUTPUT:

```
The difference of 6 is 6
The difference of 55,20 is 35
The difference of 55,25,30 is 0
```

```
def multiplication(*args):
    x=1
    for num in args:
        x=x*num
    return x
print(f"The product of 6 is {multiplication(6)}")
print(f"The product of 5,2 is {multiplication(5,2)}")
print(f"The product of 5,2,3 is {multiplication(5,2,3)}")
```

OUTPUT:

```
The product of 6 is 6
The product of 5,2 is 10
The product of 5,2,3 is 30
```

```
def division(*args):
   ans=args[0]
   for num in range(1,len(args)):
      ans=ans/args[num]
   return ans
print(f"The product of 6,6 is {division(6,6)}")
print(f"The product of 5,2 is {division(5,2)}")
print(f"The product of 5,2,3 is {division(5,2,3)}")
```

OUTPUT:

```
The product of 6,6 is 1.0
The product of 5,2 is 2.5
The product of 5,2,3 is 0.8333333333333334
```

PROGRAM: Factorial of a number using recursion

Code:

```
print("Code to find factorial of a number with recursion")
def factorial(x):
   if x==0:
      return 1
   elif x==1:
      return 1
   else:
      return(x*factorial(x-1))
num=int(input("Enter a number "))
```

```
ans=factorial(num)
print(f"Factorial of {num} is {ans}")
```

OUTPUT:

```
Code to find factorial of a number with recursion
Enter a number 6
Factorial of 6 is 720

Code to find factorial of a number with recursion
Enter a number 4
Factorial of 4 is 24
```

Program: To have two decorators and perform tasks like finding square ,cube ,multiplying by 2,incrementing by 4

Code:

```
def sq(x):
   return (x*x)
def cube(x):
   return (x**3)
def decor_add4(func,a):
   return (func(a)+4)
def decor_mul2(func,b):
   return (func(b)*2)

num=int(input("Enter a number: "))
print(f"The Square of {num} is {sq(num)}")
print(f"The Cube of {num} is {cube(num)}")
print(f"On incrementing the square {sq(num)} by 4, we get {decor_add4(sq,num)}")
print(f"On multiplying the square {sq(num)} by 2, we get {decor_mul2(sq,num)}")
print(f"On incrementing the cube {cube(num)} by 4, we get {decor_add4(cube,num)}")
print(f"On multiplying the cube {cube(num)} by 2, we get {decor_mul2(cube,num)}")
```

OUTPUT:

Enter a number: 5

The Square of 5 is 25

The Cube of 5 is 125

On incrementing the square 25 by 4, we get 29

On multiplying the square 25 by 2, we get 50

On incrementing the cube 125 by 4, we get 129

On multiplying the cube 125 by 2, we get 250

Program: To accept a list of numbers and performs following operations on a list written in another module: Sum and product of all elements, adding new elements, sum of elements at even indices

Code:

```
# Operations Module

def summation(list):
    sum=0
    for ele in range(0,len(list)):
        sum=sum+int(list[ele])
    print(sum)


def multiplication(list):
    product=1
    for ele in range(0,len(list)):
        product=product*int(list[ele])
    print(product)


def sumofeven(list):
    sum=0
    for ele in range(0,len(list)):
        if(ele %2==0):
            sum=sum+int(list[ele])
    print(sum)
```

```
def addnewelem(old_list,x):
    old_list.append(x)
    return(old_list)


#New module where we call functions of operation module
import operations
user_str=input("Enter the elements with space : ")
strlist=user_str.split(" ")
int_list=list(map(int,strlist))
print("Enter 1 for summation of all elements \nEnter 2 to get product of all elements \nEnter 3 for summation of even elements \nEnter 4 for adding a new element \n")
while(1):
    choice=int(input("Enter choice: "))
    if(choice==1):
        print("Summation of all elements")
        operations.summation(int_list)
    elif(choice==2):
        print("Product of all elements")
        operations.multiplication(int_list)
    elif(choice==3):
        print("Sum of even elements")
        operations.sumofeven(int_list)
    elif(choice==4):
        x=int(input("Enter a number: "))
        print("After adding new element")
        print(operations.addnewelem(int_list,x))
    else:
        print("Wrong option.Taking exit")
        break
```

OUTPUT:

```
Enter the elements with space : 1 2 3 4 5
Enter 1 for summation of all elements
Enter 2 to get product of all elements
Enter 3 for summation of even elements
Enter 4 for adding a new element

Enter choice: 1
Summation of all elements
15
Enter choice: 2
Product of all elements
120
Enter choice: 3
Sum of even elements
9
Enter choice: 4
Enter a number: 6
After adding new element
[1, 2, 3, 4, 5, 6]
Enter choice: 5
Wrong option. Taking exit
```