```
n = int(input("Enter a number: "))
sum = 0
for i in range(1, n):
    if(n % i == 0):
        sum = sum + i
if (sum == n):
    print(" %d is a Perfect Number" %n)
else:
    print(" %d is not a Perfect Number" %n)
```

```
Enter a number: 28
 28 is a Perfect Number
```

```
def isPalindrome(x):
    return x == x[::-1]
x = input("enter a word: ")
y = isPalindrome(x)

if x:
    print("Yes")
else:
    print("No")
```

```
enter a word: NAMAN
Yes
```

```
from math import factorial

n = 6
for i in range(n):
  for j in range(n-i+1):
    print(end=" ")

  for j in range(i+1):
    print(factorial(i)//(factorial(j)*factorial(i-j)), end=" ")
  print()
```

```
print()
```

```
    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
```

```
def pangram(x):
    check=""
    small=x.lower()
    combine=small.replace(" ","")
    for i in combine:
        if i in check:
            return False
        else:
            check+=i
    return True
print(pangram("The quick brown fox jumps over the lazy dog"))
```

```
False
```

```
items=[n for n in input().split('-')]
items.sort()
print('-'.join(items))
```

```
red-green-black-blue
black-blue-green-red
```

```
def student_data(student_id, **kwargs):
    print(f'\nStudent ID: {student_id}')
    if 'student_name' in kwargs:
        print(f"Student Name: {kwargs['student_name']}")

    if 'student_name' and 'student_class' in kwargs:
            print(f"\nStudent Name: {kwargs['student_name']}")
```

```
    print(f"Student Class: {kwargs['student_class']}")


student_data(student_id='SV12', student_name='Jean Garner')

student_data(student_id='SV12', student_name='Jean Garner', student_class ='V')
```

```
Student ID: SV12
Student Name: Jean Garner

Student ID: SV12
Student Name: Jean Garner

Student Name: Jean Garner
Student Class: V
```

```
class Student:
   pass
class Marks:
   pass
student1 = Student()
marks1 = Marks()
print(isinstance(student1, Student))
print(isinstance(marks1, Student))
print(isinstance(marks1, Marks))
print(isinstance(student1, Marks))
print("\nCheck whether the said classes are subclasses of the built-in object class c
print(issubclass(Student, object))
print(issubclass(Marks, object))
```

```
True
False
True
False

Check whether the said classes are subclasses of the built-in object class o
```

```
True
True
```

```
def findTriplets(arr, n):
    found = False
    for i in range(0, n-2):
        for j in range(i+1, n-1):
            for k in range(j+1, n):
                if (arr[i] + arr[j] + arr[k] == 0):
                    print(arr[i], arr[j], arr[k])
                    found = True
    if (found == False):
        print("does not exist")
arr = [-25, -10, -7, -3, 2, 4, 8, 10]
n = len(arr)
findTriplets(arr, n)
```

```
-10 2 8
-7 -3 10
```

```
class validity:

  def f(str):

    a= ['()', '{}', '[]']

    while any(i in str for i in a):

      for j in a:

        str = str.replace(j, '')

    return not str

s = input("enter : ")
```

```
print(s, "-", "is balanced"

      if validity.f(s) else "is Unbalanced")
```

```
enter : [{()()}{()()}]
[{()()}{()()}] - is balanced
```