```
#Write a python program function to multiply all the numbers in a list

def multi(num):
    result = 1
    for i in num:
        result = i * result
        # print(result)  '"this show iteration of each value in list"'
    return result

num = [1,2,3,4,5]
print(multi(num))


# Q.2 Write a python function to reverse a string

def ulto(word):
    sabda = word[::-1]
    return sabda

word = input('enter the string : ')
print(ulto(word))

'"Q.3 Write a python function to find factorial of a given non negativenumber"'

def factorial(num):
    if (num==0 or num==1):
        return 1

    else:
        return num * factorial(num-1)

print(factorial(5))
print(factorial(1))
print(factorial(7))
print(factorial(2))


'"Q.5 Write a python function that accepts a string and calculate the number of upper case letters and lower case letters"'


def calcStr(sabda):
    lower_count = 0
    upper_count = 0

    for char in sabda:
        if char.isupper():
            upper_count += 1
        elif char.islower():
            lower_count +=1

    return upper_count, lower_count
```

```
sabda = input('Enter the string as you wish : ')
upper, lower = calcStr(sabda)
print("Number of uppercase letters:", upper)
print("Number of lowercase letters:", lower)


'''Q.7 Write a Python function that takes a number as a parameter and check the
number is prime or not'''

def check_prime(n):
    if n == 1:
        print('Number is not a prime number.')
    elif n > 1:
        for i in range(2):
            if (n % 2) == 0:
                print(n,'is not a prime number.')
                break
            else:
                print(n,'is a prime number.')
    else:
        print(n,'is not a prime number.')

check_prime(9)


'''Q.8 Write a python function to print the even numbers from a given list'''

def even_num(num):
    a = []
    for i in num:
        if i % 2 == 0:
            a.append(i)
        else:
            pass
    return a


num = [1,2,3,4,5,6,7,8,9,12,10,60,55]
print(even_num(num))
```

'''Write a python function that takes a list and returns a new list with unique elements of the first list'''

```
def unique(nums):
    new_list = []

    for i in nums:
        if i not in new_list:
            new_list.append(i)

    return new_list

print(unique([1,2,2,3,3,3,4,4,5,5,6,7,8,8,9,0,2,]))
```

Create a set of classes representing different animals, introducing multiple levels of inheritance and abstract classes.

```
from abc import ABC, abstractmethod

class Animal(ABC):
    def __init__(self, name, species):
        self.name = name
        self.species = species


    @abstractmethod
    def move(self):
        return 'Animals do migrate.'



class Mammal(Animal):
    def __init__(self, name, species, mammal):
        super().__init__(name, species)
        self.mammal = mammal

    def move(self):
        return 'Mammal can travel.'



class Dog(Mammal):


    def bark(self):
        return 'Dog can bark'


class Cat(Mammal):
```

```
    def bark(self):
        return 'cat can do meow'


my_dog = Dog('khalid','Husky','mammal')
print(my_dog.move())
print(my_dog.bark())


# my_cat = Cat('Bhola','catwoman','mammal')
# print(my_cat.move())
# print(my_cat.bark())



class Bird(Animal):
    def __init__(self, name, species, bird):
        super().__init__(name, species)
        self.bird = bird

    def move(self):
        return 'Bird can fly.'
class Eagle(Bird):

    def fly(self):
        return 'Eagle can fly very high.'

class penguin(Bird):

    def fly(self):
        return 'penguin can not fly.'




my_bird = Eagle('chill','carnivorous','bird')
print(my_bird.move())
print(my_bird.fly())


class Fish(Animal):
    def __init__(self, name, species, fish):
        super().__init__(name, species)
        self.fish = fish

    def move(self):
        return 'fish can swim.'


class Salmon(Fish):
    def swim(self):
        return 'Salmon can swim fast.'
```

```
macha = Salmon('goldy','fewa tal','macha')
print(macha.move())
print(macha.swim())
```

Create a set of classes representing different animals, introducing multiple levels of inheritance and abstract classes.

```
class Vehicle:
    def __init__(self,make, model, year, fuel_type):
        self.make = make
        self.model = model
        self.year = year
        self.fuel_type = fuel_type



class Car(Vehicle):
    def __init__(self, make, model, year, fuel_type):
        super().__init__(make, model, year, fuel_type)


class ElectricCar(Car):
    def __init__(self, make, model, year, fuel_type, battery_capacity):
        super().__init__(make, model, year, fuel_type)
        self.battery_capacity = battery_capacity

    def info(self):
        return f'{self.make}-{self.model}-{self.year}-{self.fuel_type}-{self.battery_capacity}'

my_BYD = ElectricCar('BYD','BT5','2022','electric','80kwh')
print(my_BYD.info())


class Truck(Vehicle):
    def __init__(self, make, model, year, fuel_type):
        super().__init__(make, model, year, fuel_type)


class HyrbidTruck(Truck):
    def __init__(self, make, model, year, fuel_type, electric_motor_power):
        super().__init__(make, model, year, fuel_type)
        self.electric_motor_power = electric_motor_power

    def info(self):
        return f'{self.make}-{self.model}-{self.year}-{self.fuel_type}-{self.electric_motor_power}'


my_truck = HyrbidTruck('Benz','Monster','2019','Diesel','battery')
```

```
print(my_truck.info())

'''Create a program that generates a word pyramid pattern based on user input'''


str = input("Enter the string as you want : ")
word = len(str)
for i in range(0, word):
    for j in range(0, word-0-i):
        print(end=" ")
    for j in range(0,i+1):
        print(str[j], end=" ")
    print()
print()

# to print downward:

for i in range(word,0,-1):
    for j in range(0, word-i):
        print(end=" ")
    for j in range(0,i):
        print(str[j], end=" ")
    print()


'''LIST Manipulation  Odd-Even sorter'''

def check(numbers):
    Odd_num = []
    Even_num = []
    for i in numbers:
        if i % 2 == 0:
            Even_num.append(i)
        else:
            Odd_num.append(i)

    print(f'odd numbers : {Odd_num}')
    print(f'even numbers : {Even_num}')


numbers = int(input('ente the numbers. \n'))
A =[]
for i in range(int(numbers)):
    k = int(input(""))
    A.append(k)

check(numbers)

# Prime Factorization

def prime_factors(n):
    factors = []
```

```
    divisor = 2

    while divisor <= n:
        if n % divisor == 0:
            factors.append(divisor)
            n = n / divisor
        else:
            divisor += 1

    return factors

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

print("prime factors of ", number,"are:", prime_factors(number))


# Creating a simple grading system where a student's score is entered, and the program determines the corresponding grade.

def grade(score, name):
    if 90<= score <= 100:
        print(f"{name}. your grade is A")
    elif(80<=score < 90):
        print(f"{name}. your grade is B.")
    elif(70<= score < 80):
        print(f"{name}. your grade is C.")
    elif(60 <= score <70):
        print(f"{name}. your grade is D.")
    else:
        print(f"{name}. your grade is F.")


name = input("enter the name : ")
score = int(input("enter the score : "))

print(grade(score,name))


Create a program that calculates the ticket price for a movie based on the age and whether the customer is a student.


def ticket_price(age, is_student):

    #check if age is a valid numeric value.

    if not isinstance(age, int) or age < 0:
        print( "Invalid age entered. Please enter a non-negative integer.")
    else:
        pass

        if age <=12:
```

```
        print("children Ticket Price is $10.")
    elif age <=17:
        print("Teenagers Ticket price is $15.")
    elif is_student:
        print("Your Ticket price is $18")
    else:
        print("Adult Ticket price is $20")

age = int(input("Enter your Age Please : "))
is_student = bool(input("Are you a Student?\nIf not press Enter.: "))
ticket_price(age,is_student)


''' create a program that analyzes a given text and counts the frequency of each unique word '''


def count(user_input):
    word = (user_input.split())
    print(word)
    word2 = [] #initalize empty list

    #loop till string values present in list word2
    for i in word:

        # checking for hte duplicate
        if i not in word2:

            word2.append(i) #insert word in word2

    for i in range(0, len(word2)):

        # count the frequency of each word present in word2
        print('Frequency of ', word2[i], 'is :', word.count(word2[i]))


user_input = input("Enter the string for word count.")
count(user_input)

'''Create a program that checks if a given string can be rearranged to form a palindromic string'''

def check(user_input):

    word_1 = user_input.lower()
    print(word_1)
    word_2 = word_1[::-1]
    print(word_2)

    if word_1 == word_2:
        print("The word is Palindrome")
    else:
        print("The word is not Palindrome")
```

```
user_input = input("Enter to check palindrome. ")

check(user_input)
```