1. Tính:

a) (a + b)

b) a/b

c) ab

def Tong(*a*,*b*):

    return *a*+*b*

def Chia(*a*,*b*):

    return *a*/*b*

def Mu(*a*,*b*):

    return *a*\*\**b*

a = int(input('Nhap so thu nhat: '))

b = int(input('Nhap so thu hai: '))

print(f' {a} + {b} =  ', Tong(a,b))

print(f' {a} \ {b} = ', Chia(a,b))

print(f' {a} ^ {b} =  ', Mu(a,b))

A screenshot of a computer code

Description automatically generated

2. Tính diện tích hình chữ nhật khi biết bán kính

def DienTich(*a*,*b*):

    return *a*\**b*

a = int(input('Nhap chieu dai: '))

b = int(input('Nhap chieu rong: '))

print(f'Dien tich cua hinh chu nhat co 2 canh {a} va {b} la: ',DienTich(a,b))

A black background with white text

Description automatically generated

3. Xuất tất cả các số nguyên tố trong 1 khoảng cho trước

import math

def kiem\_tra\_so\_nguyen\_to(*n*):

   if *n* == 1:

       return False

   for i in range(2, int(math.sqrt(*n*))+1):

       if *n* % i == 0:

           return False

   return True

def liet\_ke\_so\_nguyen\_to(*a*, *b*):

   for i in range(*a*, *b* + 1):

       if kiem\_tra\_so\_nguyen\_to(i):

           print(i, *end*=' ')

try:

   a = int(input("Nhap so thu nhat: "))

   b = int(input("Nhap so thu hai: "))

   if a < 0 or b < 0:

       print("Vui long nhap so tu nhien!")

   elif a > b:

       print("So thu nhat lon hon so thu hai!")

   else:

       liet\_ke\_so\_nguyen\_to(a, b)

except:

   print("Dinh dang dau vao khong hop le!")

**A screenshot of a computer program

Description automatically generated**

4. Kiểm tra 1 số nguyên n có phải là số Fibonacci hay không

import math

n = int(input("Nhap 1 so nguyen: "))

def KiemTraTinhChat(*m*):

    n = int(math.sqrt (*m*))

    return n \* n == *m*

def KiemTraFi(*m*):

    KiemTraTinhChat(5 \* *m* \* *m* + 4)

    KiemTraTinhChat(5 \* *m* \* *m* - 4)

if (KiemTraFi (n) == True):

    print (n, "là số Fibonacci")

else:

    print (n , "không phải là số Fibnacci")



5. Tìm số Fibonacci thứ n (dùng đệ quy và không đệ quy)

\*Khong de quy

import math

n = int(input("Nhap 1 so nguyen: "))

def KiemTraTinhChat(*m*):

    n = int(math.sqrt (*m*))

    return n \* n == *m*

def KiemTraFi(*m*):

    KiemTraTinhChat(5 \* *m* \* *m* + 4)

    KiemTraTinhChat(5 \* *m* \* *m* - 4)

if (KiemTraFi (n) == True):

    print (n, "là số Fibonacci")

else:

    print (n , "không phải là số Fibnacci")

A screenshot of a computer

Description automatically generated

import math

def fibonacci(*n*):

    if (*n* < 0):

        return -1

    elif (*n* == 0 or *n* == 1):

        return *n*

    else:

        return fibonacci(*n* - 1) + fibonacci(*n* - 2)

n = int(input("Nhap 1 so nguyen: "))

try:

   n = int(input())

   if n <= 0:

       print("Vui long nhap so nguyen duong!")

   else:

       print(fibonacci(n))

except:

   print("Dinh dang dau vao khong hop le!")

6. Tính tổng n số Fibonacci đầu tiên (dùng đệ quy và không đệ quy)

7. Tính tổng căn bậc 2 của n số nguyên đầu tiên

N = int(input("Nhap n: "))

def Tinh():

    s = 0

    for i in range(1,N+1):

        s +=i\*\*2

    return s

print(f"Tongg can bac 2 cua n so nguyen dau tien: ",Tinh())



8. Giải phương trình bậc 2: ax2 + bx + c=0

import math

a = float(input('Nhap so a: '))

b = float(input('Nhap so b: '))

c = float(input('Nhap so c: '))

def TinhPhuongTrinh():

    return  b\*b-4 \*a\*c

if(TinhPhuongTrinh()<0):

    print('Phuong trinh vo nghiem!')

elif(TinhPhuongTrinh() == 0):

    print('Phuong trinh co 1 nghiem!',-b/(2\*a))

else:

    print('Phuong trinh co 2 nghiem la ' + str((-b+ math.sqrt(TinhPhuongTrinh()))/2\*a) + ' va '+ str((-b- math.sqrt(TinhPhuongTrinh()))/2\*a) )

A screen shot of a computer

Description automatically generated

9. Tính n!

n = int(input('Nhap 1 so tu nhien'))

if n > 0:

    giaithua=1

    for i in range(1,n+1):

        giaithua=giaithua\*i

    print(n,"!=",giaithua)

else:

    print("Vui lòng nhập n > 0")



10.In \* dạng tam giác dưới như hình bên, đầu vào là số hàng(cột)

n = int(input('Nhap 1 so tu nhien '))

for i in range(1, n+1):

    if(i==1):

        print('\*')

    elif(i==n):

        print('\*'\*(n))

    else:

        print('\*'+(' '\*(n-(n-i)-2))+'\*')

A screen shot of a computer

Description automatically generated

11. Đổi giờ - phút – giây: thời gian đầu vào là giây được đổi thành giờ, phút, giây.

Xuất kết quả ra màn hình dưới dạng: giờ:phút:giây. Ví dụ: soGiay = 3770 thì xuất

ra màn hình 1:2:50.

n = int(input('Nhap so giay: '))

def TinhThoiGian():

    gio = n // 3600

    phut = n%3600  // 60

    giay = n%3600%60

    print(f"{gio}:{phut}:{giay}")

TinhThoiGian()

A screen shot of a computer

Description automatically generated

12.Cho một mảng số nguyên: (nên viết 2-3 cách)

a) Xuât tất cả các số lẻ không chia hết cho 5

C1: Nhập trong khoảng a và b

try:

    print("Nhap tu so a den b de xuat cac so le khong chia het cho 5!")

    a = int(input("Nhap a: "))

    b = int(input("Nhap b: "))

    if a>b:

       print("So thu nhat lon hon so thu hai!")

    else:

       dem = 0

       for i in range(a, b+1):

           if (i % 5 != 0) and (i & 1 != 0):

               dem += 1

               if dem > 10:

                   print("\nIn qua 10 so roi!")

                   break

               print(i, *end*=" ")

       else:

           if dem == 0:

               print("Khong co so nao chia het cho 5")

           else:

               print("\nDa in het cac so le khong  chia het cho 5")

except:

   print("Dinh dang dau vao khong hop le!")

A black background with white text

Description automatically generated

C2: numbers = ["1","2","3","4","5","6","7","8","9","10"]

for x in numbers:

    if (int(x) % 5 != 0) and (int(x) & 1 != 0):

        print(x)

C3:

danhsach = []

n = int(input("nhap so phan tu co trong mang:"))

for i in range(n):

    danhsach.append(int(input()))

ketqua = []

for v in danhsach:

    if (v % 5 != 0) and (v % 2 !=0):

        ketqua.append(v)

if len(ketqua) == 0:

    ketqua = [0]

print(ketqua)

b) Xuất tất cả các số Fibonacci

c) Tìm số nguyên tố lớn nhất

import math

def fibonacci\_numbers(*arr*):

    a, b = 0, 1

    fib = []

    while b < max(*arr*):

        if b in *arr*:

            fib.append(b)

        a, b = b, a + b

    return fib

'''arr = [1, 2, 3, 5, 8, 10]

print(" So Fibonacci la:", fibonacci\_numbers(arr))'''

d) Tìm số Fibonacci bé nhất

e) Tính trung bình các số lẻ

f) Tính tích các phần tử là số lẻ không chia hết cho 3 trong mảng

import math

def fibonacci\_numbers(arr):

a, b = 0, 1

fib = []

while b < max(arr):

if b in arr:

fib.append(b)

a, b = b, a + b

return fib

'''arr = [1, 2, 3, 5, 8, 10]

print(" So Fibonacci la:", fibonacci\_numbers(arr))'''

arr = list(map(int, input("Enter a list of integers separated by spaces: ").split(" ")))

print("b) Fibonacci numbers:", fibonacci\_numbers(arr))

g) Đổi chỗ 2 phần tử của danh sách, đầu vào là 2 vị trí cần đổi chỗ

h) Đảo ngược trật tự các phần tử của danh sách

i) Xuất tất cả các số lớn thứ nhì của danh sáchCauK: Đếm số lần xuất hiện của một số trong danh sách

câu k

A screenshot of a computer program

Description automatically generated

C2

cauM: Xuất các số xuất hiện nhiều lần nhất trong danh sách

A computer screen shot of a program

Description automatically generated

CauL: Xuất các số xuất hiện n lần trong danh sách



