Bài 1:

1

print("Twinkle, twinkle, little star, \n\tHow I wonder what you are! \n\t\tUp above the world so high, \n\t\tLike a diamond in the sky. \nTwinkle, twinkle, little star, \n\tHow I wonder what you are!")

2

import platform

print(platform.python\_version())

3

import datetime

now = datetime.datetime.now()

print ("Current date and time : ")

print (now.strftime("%Y-%m-%d %H:%M:%S"))

4

from math import pi

r = float(input ("Input the radius of the circle : "))

print ("The area of the circle with radius " + str(r) + " is: " + str(pi \* r\*\*2))

5

fname = input("Input your First Name : ")

lname = input("Input your Last Name : ")

print ("Hello " + lname + " " + fname)

6

values = input("Input some comma seprated numbers : ")

list = values.split(",")

tuple = tuple(list)

print('List : ',list)

print('Tuple : ',tuple)

7

filename = input("Input the Filename: ")

f\_extns = filename.split(".")

print ("The extension of the file is : " + repr(f\_extns[-1]))

8

color\_list = ["Red","Green","White" ,"Black"]

print( "%s %s"%(color\_list[0],color\_list[-1]))

9

exam\_st\_date = (11,12,2014)

print( "The examination will start from : %i / %i / %i"%exam\_st\_date)

10

a = int(input("Input an integer : "))

n1 = int( "%s" % a )

n2 = int( "%s%s" % (a,a) )

n3 = int( "%s%s%s" % (a,a,a) )

print (n1+n2+n3)

11

print(abs.\_\_doc\_\_)

12

import calendar

y = int(input("Input the year : "))

m = int(input("Input the month : "))

print(calendar.month(y, m))

13

print("""

a string that you "don't" have to escape

This

is a ....... multi-line

heredoc string --------> example

""")

14

from datetime import date

f\_date = date(2014, 7, 2)

l\_date = date(2014, 7, 11)

delta = l\_date - f\_date

print(delta.days)

15

pi = 3.1415926535897931

r= 6.0

V= 4.0/3.0\*pi\* r\*\*3

print('The volume of the sphere is: ',V)

16

def difference(n):

if n <= 17:

return 17 - n

else:

return (n - 17) \* 2

print(difference(22))

print(difference(14))

17

def near\_thousand(n):

return ((abs(1000 - n) <= 100) or (abs(2000 - n) <= 100))

print(near\_thousand(1000))

print(near\_thousand(900))

print(near\_thousand(800))

print(near\_thousand(2200))

18

def sum\_thrice(x, y, z):

sum = x + y + z

if x == y == z:

sum = sum \* 3

return sum

print(sum\_thrice(1, 2, 3))

print(sum\_thrice(3, 3, 3))

19

def new\_string(text):

if len(text) >= 2 and text [:2] == "Is":

return text

return "Is" + text

print(new\_string("Array"))

print(new\_string("IsEmpty"))

20

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

mod = num % 2

if mod > 0:

print("This is an odd number.")

else:

print("This is an even number.")

21

def list\_count\_4(nums):

count = 0

for num in nums:

if num == 4:

count = count + 1

return count

print(list\_count\_4([1, 4, 6, 7, 4]))

print(list\_count\_4([1, 4, 6, 4, 7, 4]))

22

def substring\_copy(text, n):

flen = 2

if flen > len(text):

flen = len(text)

substr = text[:flen]

result = ""

for i in range(n):

result = result + substr

return result

print(substring\_copy('abcdef', 2))

print(substring\_copy('p', 3));

23

def is\_vowel(char):

all\_vowels = 'aeiou'

return char in all\_vowels

print(is\_vowel('c'))

print(is\_vowel('e'))

24

def is\_vowel(char):

all\_vowels = 'aeiou'

return char in all\_vowels

print(is\_vowel('c'))

print(is\_vowel('e'))

25

def is\_group\_member(group\_data, n):

for value in group\_data:

if n == value:

return True

return False

print(is\_group\_member([1, 5, 8, 3], 3))

print(is\_group\_member([5, 8, 3], -1))

Bài 2:

1

a)

a = int(input("Nhap a :"))

b = int(input("Nhap b :"))

def Tong( a, b):

    return a+b

print(Tong(a,b))

b)

a = int(input("Nhap a :"))

b = int(input("Nhap b :"))

def Thuong( a, b):

    return a/b

print(Thuong(a,b))

c)

a = int(input("Nhap a :"))

b = int(input("Nhap b :"))

def LuyThua(a, b):

    return pow(a,b)

print(LuyThua(a,b))

2

3

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())

   b = int(input())

   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!")

4

try:

   n = int(input("Nhap so:"))

   if n <= 0:

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

   elif n == 1 or n == 2:

       print(1)

   else:

       soThuNhat, soThuHai = 1, 1

       for i in range(n-2):

           soThuNhat, soThuHai = soThuHai, soThuNhat + soThuHai

       print(soThuHai)

except:

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

5

Không dung đệ quy

def fibonacci(n):

    f0 = 0

    f1 = 1

    fn = 1

    if (n < 0):

        return -1

    elif (n == 0 or n == 1):

        return n

    else:

        for i in range(2, n):

            f0 = f1

            f1 = fn

            fn = f0 + f1

        return fn

print("10 số đầu tiên của dãy số fibonacci: ")

sb = ""

for i in range(0, 10):

    sb = sb + str(fibonacci(i)) + ", "

print(sb)

dùng đệ quy

def fibonacci(n):

    if (n < 0):

        return -1

    elif (n == 0 or n == 1):

        return n

    else:

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

print("10 số đầu tiên của dãy số fibonacci: ")

sb = ""

for i in range(0, 10):

    sb = sb + str(fibonacci(i)) + ", "

print(sb)

6

7

import math

n = int(input("Nhap n: "))

def Cau7(n):

    tong = 0

    for i in range(1, n+1):

        tong = tong +  math.sqrt(i)

    return tong

print("Tong la:",Cau7(n))

8

import math

a = float(input("Mời bạn nhập hệ số a: "))

while True:

    if a == 0:

        a = float(input("Số a phải khác 0. Mời nhập lại số a: "))

    else:

        break

b = float(input("Mời bạn nhập hệ số b: "))

while True:

    if b == 0:

        b = float(input("Số b phải khác 0. Mời nhập lại số b: "))

    else:

        break

c = float(input("Mời bạn nhập hệ số c: "))

delta = b\*\*2 - 4 \* a \* c

if delta < 0:

    print("Phương trình vô nghiệm")

elif delta == 0:

    print("Phương trình có nghiệm kép x1 = x2 = ", -(b / (2 \* a)) )

else:

    print("Phương trình có hai nghiệm phân biệt:")

    print("x1 = ", (-(b) + math.sqrt(delta))/(2\*a) )

    print("x2 = ", (-(b) - math.sqrt(delta))/(2\*a) )

9

n = int(input("Nhập số cần tính giai thừa: "))

def giaiThua(n):

    if n == 0:

        return 1

    return n \* giaiThua(n - 1)

print (giaiThua(n))

10

def print\_lower\_triangle(rows):

    for i in range(rows):

        for j in range(i + 1):

            if j == 0 or j == i or i == rows - 1:

                print("\*", end="")

            else:

                print(" ", end="")

        print()

# Nhập số hàng từ người dùng

rows = int(input("Nhập số hàng: "))

print\_lower\_triangle(rows)

11

print("nhập số giây:")

t=int(input())

a = int(t / 3600)

t = t % 3600

b = int(t / 60)

c = t % 60

print(str(a)+":"+str(b)+":"+str(c))

12

a

odd\_not\_divisible\_by\_5 = [num for num in numbers if num % 2 != 0 and num % 5 != 0]

print("Các số lẻ không chia hết cho 5:", odd\_not\_divisible\_by\_5)

b

def is\_perfect\_square(x):

    sqrt = int(math.sqrt(x))

    return sqrt \* sqrt == x

def is\_fibonacci(n):

    if n <= 0:

        return False

    return is\_perfect\_square(5 \* n \* n + 4) or is\_perfect\_square(5 \* n \* n - 4)

fibonacci\_numbers = [num for num in numbers if is\_fibonacci(num)]

print("Các số Fibonacci:", fibonacci\_numbers)

c

def is\_prime(n):

    if n <= 1:

        return False

    if n <= 3:

        return True

    if n % 2 == 0 or n % 3 == 0:

        return False

    i = 5

    while i \* i <= n:

        if n % i == 0 or n % (i + 2) == 0:

            return False

        i += 6

    return True

prime\_numbers = [num for num in numbers if is\_prime(num)]

max\_prime = max(prime\_numbers)

print("Số nguyên tố lớn nhất:", max\_prime)

d

min\_fibonacci = min(fibonacci\_numbers)

print("Số Fibonacci bé nhất:", min\_fibonacci)

e

odd\_numbers = [num for num in numbers if num % 2 != 0]

avg\_odd = sum(odd\_numbers) / len(odd\_numbers)

print("Trung bình các số lẻ:", avg\_odd)

f

odd\_not\_divisible\_by\_3 = [num for num in numbers if num % 2 != 0 and num % 3 != 0]

product\_odd\_not\_divisible\_by\_3 = 1

for num in odd\_not\_divisible\_by\_3:

    product\_odd\_not\_divisible\_by\_3 \*= num

print("Tích các số lẻ không chia hết cho 3:", product\_odd\_not\_divisible\_by\_3)

g

def swap\_elements(lst, pos1, pos2):

    lst[pos1], lst[pos2] = lst[pos2], lst[pos1]

pos1 = int(input("Nhập vị trí thứ nhất: "))

pos2 = int(input("Nhập vị trí thứ hai: "))

swap\_elements(numbers, pos1, pos2)

print("Danh sách sau khi đổi chỗ:", numbers)

h

reversed\_numbers = numbers[::-1]

print("Danh sách sau khi đảo ngược:", reversed\_numbers)

i

sorted\_numbers = sorted(numbers, reverse=True)

second\_largest = sorted\_numbers[1]

print("Số lớn thứ nhì:", second\_largest)

j

def sum\_of\_digits(n):

    total = 0

    while n > 0:

        total += n % 10

        n //= 10

    return total

total\_digits\_sum = sum(sum\_of\_digits(num) for num in numbers)

print("Tổng các chữ số:", total\_digits\_sum)

k

number\_to\_count = int(input("Nhập số cần đếm: "))

count = numbers.count(number\_to\_count)

print(f"Số lần xuất hiện của {number\_to\_count}: {count} lần")

i

n = int(input("Nhập số lần xuất hiện: "))

numbers\_appearing\_n\_times = [num for num in set(numbers) if numbers.count(num) == n]

print(f"Các số xuất hiện {n} lần:", numbers\_appearing\_n\_times)

m

max\_occurrences = max(numbers.count(num) for num in numbers)

numbers\_with\_max\_occurrences = [num for num in set(numbers) if numbers.count(num) == max\_occurrences]

print("Các số xuất hiện nhiều lần nhất:", numbers\_with\_max\_occurrences)