#BÀI 1: NHẬP VÀO MỘT DÃY SỐ. KIỂM TRA CÓ BAO NHIÊU SỐ NGUYÊN TỐ TRONG DÃY.

import math

def snt(n):

if n<=1:

kt=False

else:

kt=True

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

if n%i==0:

kt=False

break

return kt

#a=['5','12','17','23','21','4','11']

a=[]

s=input()

a=s.split()

d=0

for i in a:

m=int(i)

if snt(m):

d+=1;

print('so luong so nguyen to la:',d)

#ĐƯA RA SỐ FIBONACCI THỨ N

def fibo(n):

f0=1

f1=1

f=1

s=[1,1]

for i in range(2,n+1):

f=f0+f1

s.append(f)

f0=f1

f1=f

print(s)

return f

n=5

print(fibo(n))

# NHẬP VÀO N. TÍNH N!

def giaithua(n):

gt=1

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

gt\*=i

return gt

n=0

print(giaithua(n))

n=8. Tính n!

1\*2\*3\*4\*5\*6\*7\*8= 40320

def giaithua(n):

gt=1

s=''

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

s+=str(i)+'\*'

gt\*=i

s=s.rstrip('\*')

print(s,end='')

return gt

n=8

print('=',giaithua(n))

#BÀI 4. NHẬP VÀO MỘT SỐ THẬP PHÂN N, CHUYỂN N SANG HỆ NHỊ PHÂN

def nhiphan(n):

s=''

while n>0:

du=n%2

s=str(du)+s

n=n//2

return s

n=18

print(n,'=',end='')

print(nhiphan(n))

**Bài 1.**Viết chương trình tìm tất cả các số chia hết cho 7 nhưng không phải bội số của 5, nằm trong đoạn 2000 và 3200 (tính cả 2000 và 3200). Các số thu được sẽ được in thành chuỗi trên một dòng, cách nhau bằng dấu phẩy.

j=[] **for** i **in** range(2000, 3201):

**if** (i%7==0) and (i%5!=0):

j.append(str(i))

print (','.join(j))

Nếu chỉ cần in ra màn hình kết quả, chúng ta có thể không cần sử dụng List.

**for** i **in** range(2000, 3201)

**Bài 03.**Với số nguyên n nhất định, hãy viết chương trình để tạo ra một dictionary chứa (i, i\*i) như là số nguyên từ 1 đến n (bao gồm cả 1 và n) sau đó in ra dictionary này. Ví dụ: Giả sử số n là 8 thì đầu ra sẽ là: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}.

n = int(input('Enter a number '))

d = dict()

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

d[i] = i\*i

print(d)

**Bài 04.**Viết chương trình chấp nhận một chuỗi số, phân tách bằng dấu phẩy từ giao diện điều khiển, tạo ra một List  và một tuple chứa mọi số.

Ví dụ: Đầu vào được cung cấp là 34,67,55,33,12,98 thì đầu ra là:

['34', '67', '55', '33', '12', '98']  
('34', '67', '55', '33', '12', '98')

Chương trình này chỉ đơn giản là sử dụng hàm split() và chuyển một List sang một tuple.

values=input("Nhập vào các giá trị:")

l=values.split(",")

t=tuple(l)

print (l)

print (t)

**Bài 05.**Viết một hàm tính giá trị bình phương của một số.

# square of a number

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

**def** square(x):

**return** x \* x

**Bài 06.**Viết chương trình tính số Fibonacci thứ n, với n nhập vào từ bàn phím.

# find fibonacci number

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

**def** fibonacci(n):

**if** n == 0:

**return** 0

**elif** n == 1:

**return** 1

**else**:

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

**Bài 07.**Viết một chương trình nhập vào một danh sách các số và tạo một danh sách mới chỉ gồm phần tử đầu tiên và cuối cùng của danh sách đó. Viết chương trình sử dụng hàm.

Ví dụ, nhập vào danh sách [5, 10, 15, 20, 25] thì kết quả trả về là danh sách [5, 25]

**def** list\_ends(a\_list):

**return** [a\_list[0], a\_list[len(a\_list)-1]]

**Bài 08.**Viết một hàm nhận vào ba số thực và trả về số lớn nhất trong ba số. Lưu ý, không sử dụng hàm max() của Python.

# max of three numbers

**def** max\_of\_three(a, b, c):

**if** a

**Bài 09.**Viết chương trình yêu cầu người dùng nhập vào một chuỗi và in ra màn hình thông báo chuỗi đó có phải là chuỗi **palindrome** hay không. (Chuỗi **Palindrome** là một chuỗi mà đọc xuôi và ngược đều như nhau, ví dụ ABCDCBA.)

Cách giải thứ nhất, sử dụng cách đảo ngược xâu:

wrd=input("Please enter a word")

wrd=str(wrd)

rvs=wrd[::-1]

print(rvs)

**if** wrd == rvs:

print("This word is a palindrome")

**else**:

print("This word is not a palindrome")

Cách thứ hai, sử dụng vòng lặp for

**def** reverse(word):

x = ''

**for** i **in** range(len(word)):

x += word[len(word)-1-i]

**return** x

word = input('give me a word:\n')

x = reverse(word)

**if** x == word:

print('This is a Palindrome')

**else**:

print('This is NOT a Palindrome')

**Bài 10.**Viết chương trình hỏi người dùng một số tự nhiên n và in ra tất cả ước số của con số đó.

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

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

**if** n % i == 0:

print(i)

**Bài 11.**Viết một chương trình (sử dụng các hàm) yêu cầu người dùng nhập một chuỗi dài gồm nhiều từ. In lại cho người dùng một chuỗi mới với thứ tự từ ngữ được đảo ngược lại với thứ tự ban đầu. Ví dụ, khi người dùng nhập chuỗi: Toi la Phuong thì in ra màn hình Phuong la Toi

sentense = input("Enter a sentence: ")

words = sentence.split()

words.reverse()

sentence = " ".join(words)

print(sentence)

**Bài 12.**Viết chương trình kiểm tra xem số n có là số nguyên tố hay không.

# check prime number

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

**for** i **in** range(2, n):

**if** n % i == 0:

print("Not a prime number")

**break**

**else**:

print("Prime number")

**Bài 13.**Viết một chương trình nhập vào hai số tự nhiên m, n. In ra màn hình mảng hai chiều mà phần tử trong hàng thứ i và cột thứ j của mảng là i\*j.

Ví dụ: Giá trị m, n nhập vào là 3 và 5 thì đầu ra là: [[0, 0, 0, 0, 0], [0, 1, 2, 3, 4], [0, 2, 4, 6, 8]]

m, n = map(int, input('Enter two numbers: ').split())

array2 = [[0 **for** i **in** range(n)] **for** j **in** range(m)]

**for** row **in** range(m):

**for** col **in** range(n):

array2[row][col] = row \* col

print(array2)

**Bài 14.**Viết một chương trình nhận chuỗi từ do người dùng nhập vào, phân tách nhau bởi dấu phẩy và in những từ đó thành chuỗi theo thứ tự bảng chữ cái, phân tách nhau bằng dấu phẩy.

Giả sử đầu vào được nhập là: without,hello,bag,world thì đầu ra sẽ là bag,hello,without,world.

items=[x **for** x **in** input("Nhập một chuỗi: ").split(',')]

items.sort()

print (','.join(items))

**Bài 15.**Viết chương trình giải phương trình bậc hai ax2+bx+c=0 với a, b, c là số nguyên và được nhập vào từ bàn phím.

a, b, c = map(int, input('Nhập a, b, c cách nhau bằng dấu cách: ').split())

**if** a == 0:

**if** b == 0:

**if** c == 0:

print("Phương trình có vô số nghiệm")

**else**:

print("Phương trình vô nghiệm")

**else**:

print("Phương trình có một nghiệm x =", -c/b)

**else**:

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 2 nghiệm phân biệt x1 =", (-b + delta\*\*0.5)/(2\*a), "và x2 =", (-b - delta\*\*0.5)/(2\*a))

**Bài 16.**Viết chương trình tính tổng của các chữ số của môt số nguyên dương n trong Python. Số nguyên dương n được nhập từ bàn phím.

**def** totalDigitsOfNumber(n):

total = 0;

**while** (n > 0):

total = total + n % 10;

n = int(n / 10);

**return** total;

n = int(input("Nhập số nguyên dương n = "));

print("Tổng các chữ số của", n , "là", totalDigitsOfNumber(n));

**Bài 17.**Câu hỏi:

Viết một chương trình chấp nhận đầu vào là chuỗi các số nhị phân 4 chữ số, phân tách bởi dấu phẩy, kiểm tra xem chúng có chia hết cho 5 không. Sau đó in các số chia hết cho 5 thành dãy phân tách bởi dấu phẩy.

Ví dụ đầu vào là: 0100,0011,1010,1001

Đầu ra sẽ là: 1010

Gợi ý:

Trong trường hợp dữ liệu đầu vào được nhập vào chương trình nó nên được giả định là dữ liệu được người dùng nhập vào từ giao diện điều khiển.

Code mẫu:

value = []

items=[x for x in input("Nhập các số nhị phân: ").split(',')]

for p in items:

intp = int(p, 2)

if not intp%5:

value.append(p)

# Bài tập Python 14, viết bởi Quantrimang.com

print (','.join(value))

**Bài 16:**

Câu hỏi:

Viết một chương trình chấp nhận đầu vào là một câu, đếm số chữ cái và chữ số trong câu đó. Giả sử đầu vào sau được cấp cho chương trình: hello world! 123

Thì đầu ra sẽ là:

Số chữ cái là: 10  
Số chữ số là: 3

Gợi ý:

Trong trường hợp dữ liệu đầu vào được nhập vào chương trình nó nên được giả định là dữ liệu được người dùng nhập vào từ giao diện điều khiển.

Code mẫu:

s = input("Nhập câu của bạn: ")

# Bài tập Python 16, Code by Quantrimang.com

d={"DIGITS":0, "LETTERS":0}

for c in s:

if c.isdigit():

d["DIGITS"]+=1

elif c.isalpha():

d["LETTERS"]+=1

else:

pass

print ("Số chữ cái là:", d["LETTERS"])

print ("Số chữ số là:", d["DIGITS"])

**Bài 17:**

Câu hỏi:

Viết một chương trình chấp nhận đầu vào là một câu, đếm chữ hoa, chữ thường.

Giả sử đầu vào là: Quản Trị Mạng

Thì đầu ra là:

Chữ hoa: 3

Chữ thường: 8

Gợi ý:

Trong trường hợp dữ liệu đầu vào được nhập vào chương trình nó nên được giả định là dữ liệu được người dùng nhập vào từ giao diện điều khiển.

Code mẫu:

s = input("Nhập câu của bạn: ")

d={"UPPER CASE":0, "LOWER CASE":0}

# Code by Quantrimang.com

for c in s:

if c.isupper():

d["UPPER CASE"]+=1

elif c.islower():

d["LOWER CASE"]+=1

else:

pass

print ("Chữ hoa:", d["UPPER CASE"])

print ("Chữ thường:", d["LOWER CASE"])

**Bài 20:**

Câu hỏi:

Viết chương trình tính số tiền thực của một tài khoản ngân hàng dựa trên nhật ký giao dịch được nhập vào từ giao diện điều khiển.

Định dạng nhật ký được hiển thị như sau:

D 100  
W 200

(D là tiền gửi, W là tiền rút ra).

Giả sử đầu vào được cung cấp là:

D 300

D 300

W 200

D 100

Thì đầu ra sẽ là:

500

Gợi ý:

Trong trường hợp dữ liệu đầu vào được nhập vào chương trình nó nên được giả định là dữ liệu được người dùng nhập vào từ giao diện điều khiển.

Code mẫu:

import sys

netAmount = 0

# Bài tập Python 20, Code by Quantrimang.com

while True:

s = input("Nhập nhật ký giao dịch: ")

if not s:

break

values = s.split(" ")

operation = values[0]

amount = int(values[1])

if operation=="D":

netAmount+=amount

elif operation=="W":

netAmount-=amount

else:

pass

print (netAmount)

**5. Bài tập Python level 3**

**Bài 21:**

Câu hỏi:

Một website yêu cầu người dùng nhập tên người dùng và mật khẩu để đăng ký. Viết chương trình để kiểm tra tính hợp lệ của mật khẩu mà người dùng nhập vào.

Các tiêu chí kiểm tra mật khẩu bao gồm:

1. Ít nhất 1 chữ cái nằm trong [a-z]  
2. Ít nhất 1 số nằm trong [0-9]  
3. Ít nhất 1 kí tự nằm trong [A-Z]  
4. Ít nhất 1 ký tự nằm trong [$ # @]  
5. Độ dài mật khẩu tối thiểu: 6  
6. Độ dài mật khẩu tối đa: 12

Chương trình phải chấp nhận một chuỗi mật khẩu phân tách nhau bởi dấu phẩy và kiểm tra xem chúng có đáp ứng những tiêu chí trên hay không. Mật khẩu hợp lệ sẽ được in, mỗi mật khẩu cách nhau bởi dấu phẩy.

Ví dụ mật khẩu nhập vào chương trình là: ABd1234@1,a F1#,2w3E\*,2We3345

Thì đầu ra sẽ là: ABd1234@1

Gợi ý:

Trong trường hợp dữ liệu đầu vào được nhập vào chương trình nó nên được giả định là dữ liệu được người dùng nhập vào từ giao diện điều khiển.

Code mẫu:

import re

value = []

items=[x for x in input("Nhập mật khẩu: ").split(',')]

# Bài tập Python 21, Code by Quantrimang.com

for p in items:

if len(p)<6 or len(p)>12:

continue

else:

pass

if not re.search("[a-z]",p):

continue

elif not re.search("[0-9]",p):

continue

elif not re.search("[A-Z]",p):

continue

elif not re.search("[$#@]",p):

continue

elif re.search("\s",p):

continue

else:

pass

value.append(p)

print (",".join(value))

**Bài 22:**

Câu hỏi:

Viết chương trình sắp xếp tuple (name, age, score) theo thứ tự tăng dần, name là string, age và height là number. Tuple được nhập vào bởi người dùng. Tiêu chí sắp xếp là:

Sắp xếp theo name sau đó sắp xếp theo age, sau đó sắp xếp theo score. Ưu tiên là tên > tuổi > điểm.

Nếu đầu vào là:

Tom,19,80  
John,20,90  
Jony,17,91  
Jony,17,93  
Json,21,85

Thì đầu ra sẽ là:

[('John', '20', '90'), ('Jony', '17', '91'), ('Jony', '17', '93'), ('Json', '21', '85'), ('Tom', '19', '80')]

Gợi ý:

Trong trường hợp dữ liệu đầu vào được nhập vào chương trình nó nên được giả định là dữ liệu được người dùng nhập vào từ giao diện điều khiển.

Sử dụng itemgetter để chấp nhận nhiều key sắp xếp.

Code mẫu:

from operator import itemgetter, attrgetter

# Bài tập Python 22 Code by Quantrimang.com

l = []

while True:

s = input()

if not s:

break

l.append(tuple(s.split(",")))

print (sorted(l, key=itemgetter(0,1,2)))

**Bài 23:**

Câu hỏi:

Xác định một class với generator có thể lặp lại các số nằm trong khoảng 0 và n, và chia hết cho 7.

Gợi ý:

Sử dụng yield.

Code mẫu:

def putNumbers(n):

i = 0

while i<n:

j=i

i=i+1

if j%7==0:

yield j

# Bài tập Python 23 Code by Quantrimang.com

for i in putNumbers (100):

print (i)

**Bài 24:**

Câu hỏi:

Một Robot di chuyển trong mặt phẳng bắt đầu từ điểm đầu tiên (0,0). Robot có thể di chuyển theo hướng UP, DOWN, LEFT và RIGHT với những bước nhất định. Dấu di chuyển của robot được đánh hiển thị như sau:

UP 5

DOWN 3

LEFT 3

RIGHT 3

Các con số sau phía sau hướng di chuyển chính là số bước đi. Hãy viết chương trình để tính toán khoảng cách từ vị trí hiện tại đến vị trí đầu tiên, sau khi robot đã di chuyển một quãng đường. Nếu khoảng cách là một số thập phân chỉ cần in só nguyên gần nhất.

Ví dụ: Nếu tuple sau đây là input của chương trình:

UP 5  
DOWN 3  
LEFT 3  
RIGHT 2

thì đầu ra sẽ là 2.

Gợi ý:

Trong trường hợp dữ liệu đầu vào được nhập vào chương trình nó nên được giả định là dữ liệu được người dùng nhập vào từ giao diện điều khiển.

Code mẫu:

import math

pos = [0,0]

while True:

s = input()

if not s:

break

movement = s.split(" ")

direction = movement[0]

steps = int(movement[1])

if direction=="UP":

pos[0]+=steps

elif direction=="DOWN":

pos[0]-=steps

elif direction=="LEFT":

pos[1]-=steps

elif direction=="RIGHT":

pos[1]+=steps

else:

pass

# Bài tập Python 24 Code by Quantrimang.com

print (int(round(math.sqrt(pos[1]\*\*2+pos[0]\*\*2))))