**BÁO CÁO THỰC HÀNH**

**Môn học: Lập trình ứng dụng Web**

**Kỳ báo cáo: Bài tập 3**

**Tên chủ đề: GIỚI THIỆU VỀ PYTHON**

*GVHD: Trần Tuấn Dũng*

*Ngày báo cáo: 30/11/2021*

**Nhóm: 07 (ghi số thứ tự nhóm)**

1. **THÔNG TIN CHUNG:**

*(Liệt kê tất cả các thành viên trong nhóm)*

Lớp: NT208.M11.ATCL

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Nguyễn Khải Đăng | 19521317 | 19521317@gm.uit.edu.vn |
|  |  |  |  |

1. **NỘI DUNG THỰC HIỆN:[[1]](#footnote-1)**

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.**

# Sinh viên thực hiện tìm hiểu và trả lời các câu hỏi sau:

## Phân biệt các kiểu dữ liệu: int, float, str, list, tuple, set, dict, bool, bytes

*Trả lời:*

* Text Type: str
* Numeric Types: int, float, complex
* Sequence Types: list, tuple, range
* Mapping Type: dict
* Set Types: set, frozenset
* Boolean Type: bool
* Binary Types: bytes, bytearray, memoryview

Trong đó:

**Python numbers:**

* Int: hoặc integer, là một số nguyên, dương hoặc âm, không có số thập phân, có độ dài không giới hạn.

x = 1

y = 35656222554887711

z = -3255522

* Float: hoặc "số dấu phẩy động" là một số, dương hoặc âm, chứa một hoặc nhiều số thập phân. Float cũng có thể là các số khoa học với chữ "e" để biểu thị lũy thừa của 10.

x = 1.10

y = 1.0

z = -35.59

x = 35e3

y = 12E4

z = -87.7e100

* Complex (số phức) được viết với "j" là phần ảo:

x = 3+5j

y = 5j

z = -5j

**Python Strings**

* String Literals: Các ký tự chuỗi trong python được bao quanh bởi dấu ngoặc kép đơn hoặc dấu ngoặc kép.

print("Hello")

print('Hello')

**Multiline Strings:**

* Có thể gán một chuỗi nhiều dòng cho một biến bằng cách sử dụng ba dấu ngoặc kép

a = """Lorem ipsum dolor sit amet,

**Chuỗi là Mảng:**

* Giống như nhiều ngôn ngữ lập trình phổ biến khác, chuỗi trong Python là các mảng byte đại diện cho các ký tự unicode. Tuy nhiên, Python không có kiểu dữ liệu ký tự, một ký tự đơn giản chỉ là một chuỗi có độ dài là 1. Dấu ngoặc vuông có thể được sử dụng để truy cập các phần tử của chuỗi.

a = "Hello, World!"

print(a[1])

**Python Booleans:**

* Boolean đại diện cho một trong hai giá trị: True hoặc False.

**Python Lists:**

* List là một tập hợp được sắp xếp và có thể thay đổi. Trong Python, list được viết bằng dấu ngoặc vuông.

thislist = ["apple", "banana", "cherry"]

print(thislist)

**Python Tuples:**

* Tuple là một tập hợp được sắp xếp theo thứ tự và không thể thay đổi. Trong Python, tuple được viết bằng dấu ngoặc tròn.

thistuple = ("apple", "banana", "cherry")

print(thistuple)

**Python Sets:**

* Set là một tập hợp không có thứ tự và không được lập chỉ mục. Trong Python, set được viết bằng dấu ngoặc nhọn.

thisset = {"apple", "banana", "cherry"}

print(thisset)

**Python Dictionaries:**

* Dictionary là một tập hợp không có thứ tự, có thể thay đổi và được lập chỉ mục. Trong Python dictionary được viết bằng dấu ngoặc nhọn và chúng có các khóa và giá trị**.**

thisdict = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

print(thisdict)

## Cú pháp và ví dụ minh hoạ của các lệnh:

* if … (và các trường hợp khác của if…)

a = input()

b = input()

*try*:

    so1 = int(a)

    so2 = int(b)

    tich = so1 \* so2

*if* tich > 1000:

        print("tong hai so la: ", so1+so2)

*else*:

        print("tong hai so la: ", tich)

*except*:

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

* Vòng lặp For (lặp lại trong khoảng a ≤ i < b, lặp lần lượt các phần tử trong list)

*for* i *in* range(a, b-1):

    print(i);

* Vòng lặp While

*while* k > 0:

    print(k)

* Khai báo và sử dụng hàm

def hello():

    print('Hello!')

## Python Lambda là gì? Ví dụ sử dụng lambda để chuyển các phần tử âm trong list về giá trị 0.

Trong Python, hàm vô danh(Lamba) là hàm được định nghĩa mà không có tên. Nếu các hàm bình thường được định nghĩa bằng cách sử dụng từ khóa def, thì hàm vô danh được định nghĩa bằng cách sử dụng từ khóa lambda. Vì lý do đó mà hàm vô danh còn được gọi là hàm Lambda.

Ví dụ:

Code:

lambda\_test = lambda *a*:  a *if* a > 0 *else* 0

list\_test = [-2, 2, 3]

*for* item *in* list\_test:

    print (lambda\_test(item))

Ảnh có chứa văn bản

Mô tả được tạo tự động

1. Kết quả khi thực hiện hàm

# Bài tập lập trình Python

## Cho người dùng nhập vào hai số nguyên và trả về tích của chúng. Nếu kết quả lớn hơn 1000 thì trả về tổng của chúng

Code:

a = input()

b = input()

*try*:

    so1 = int(a)

    so2 = int(b)

    tich = so1 \* so2

*if* tich > 1000:

        print("Tổng hai so la: ", so1+so2)

*else*:

        print("Tích hai so la: ", tich)

*except*:

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

Kết quả:

Ảnh có chứa văn bản

Mô tả được tạo tự động

1. Kết quả bài tập 1

## Cho người dùng nhập vào số nguyên dương n. Sử dụng hàm range() để sinh ra dãy số 0 ≤ x < n, thực hiện tính tổng dãy trên và xuất kết quả ra màn hình.

Code:

n = input()

sum=0

*try*:

*for* item *in* range(0, int(n)):

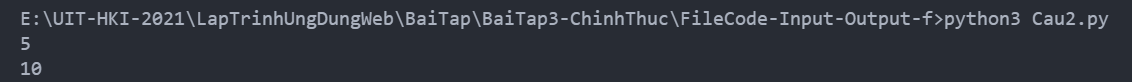
        sum+=item

    print(sum)

*except*:

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

Kết quả:



1. Kết quả bài tập 2

## Cho phép người dùng nhập chuỗi ký tự từ bàn phím. Hiển thị những ký tự ở vị trí chẵn ra màn hình (theo thứ tự trong chuỗi).

Code:

text = input()

*try*:

*for* item *in* text:

*if*(text.index(item)%2!=0):

            print (item)

*except*:

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

Kết quả:

Ảnh có chứa văn bản

Mô tả được tạo tự động

1. Kết quả bài tập 3

## Cho phép người dùng nhập chuỗi ký tự từ bàn phím và số nguyên n. Hiển thị ra màn hình n ký tự cuối cùng trong chuỗi trên

Code:

n = input()

text = input()

*try*:

    print(text[len(text)-int(n):len(text)])

*except*:

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

Kết quả:

Ảnh có chứa văn bản

Mô tả được tạo tự động

1. Kết quả bài tập 4

## Cho danh sách các số nguyên được nhập từ bàn phím. Trả về True nếu số đầu tiên và số cuối cùng của danh sách giống nhau. Ngược lại, trả về False

Code:

n = int(input("Nhập số phần tử của dãy số : "))

mylist = []

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

   val = int(input('Nhập một số: '))

   mylist.append(val)

*if* (mylist[0] == mylist[n-1]):

   print("true")

*else*:

   print("false")

Kết quả:

Ảnh có chứa văn bản

Mô tả được tạo tự động

1. Kết quả bài tập 5

## Cho phép người dùng nhập vào số nguyên dương n. Thực hiện sinh ra n số nguyên ngẫu nhiên và in ra màn hình những số nguyên tố trong dãy số vừa sinh ra theo thứ tự tăng dần.

Code:

def is\_prime(*n*):

    count = 0

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

*if* n % i == 0:

            count += 1

*if* count == 2:

        print(n)

*return* False

n = int(input("Nhập số phần tử của dãy số : "))

mylist = []

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

   val = int(input('Nhập một số: '))

   mylist.append(val)

mylist2 = sorted(mylist)

*for* item *in* mylist2:

    is\_prime(item)

Kết quả:

Ảnh có chứa văn bản

Mô tả được tạo tự động

1. Kết quả bài tập 6

## Cho một chuỗi bất kỳ nhập từ bàn phím, đảo ngược chuỗi trên và xuất ra màn hình. Với các ký tự in thường, chuyển thành in hoa; và ngược lại, chuyển các ký tự in hoa sang in thường.

Code:

text = input()

print(text.swapcase())

Kết quả:

Ảnh có chứa văn bản

Mô tả được tạo tự động

1. Kết quả bài tập 7

## Cho 1 list gồm n số nguyên được sinh ngẫu nhiên (n nhập từ bàn phím), hãy ghi các phần tử của list này ra file “f.txt” sao cho:

* Mỗi phần tử nằm riêng lẻ trên 1 dòng
* Dòng dưới sẽ thụt lùi vào trong so với dòng trên 1 dấu cách

Code:

*import* random

n = int(input("Nhập số phần tử của dãy số : "))

mylist = []

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

   val = random.randint(1, 1000)

   mylist.append(val)

path\_w = 'f.txt'

s = ''

*for* item *in* mylist:

    temp = ''

*for* i *in* range(mylist.index(item)):

        temp += ' '

    s += temp + str(item) + '\n'

*with* open(path\_w, *mode*='w') *as* f:

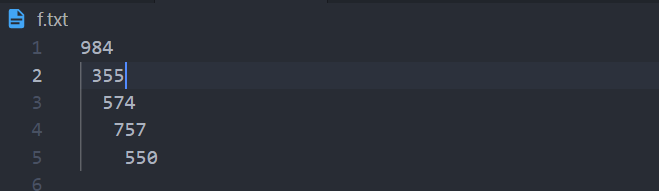
    f.write(s)

Kết quả:

Ảnh có chứa văn bản

Mô tả được tạo tự động

1. Chạy lệnh



1. Kết quả bài tập 8

## Hãy đọc file “f.txt” trong bài trên và xuất ra màn hình biểu thức phép tính cộng và tổng của chúng của các số có trong file.

Ví dụ:

Nội dung file f.txt: 5 12 15

Output: 5 + 12 + 15 = 32

Code:

mylist = []

f = open('f.txt', 'r', *encoding*='UTF-8')

data = f.read()

data = data.replace(" ", "")

data = data.rstrip("\n")

mylist = data.split("\n")

*for* i *in* mylist:

*if*(len(mylist)-1 != mylist.index(i)):

        print(i+ '+', *end*='')

*else*:

        print(i+ '= ', *end*='')

sums=0

*for* i *in* mylist:

    sums += int(i)

print(sums)

Kết quả:

Ảnh có chứa văn bản

Mô tả được tạo tự động

1. Kết quả bài tập 9

## Hãy chèn thêm dòng chữ “Lập trình ứng dụng WEB” (viết Tiếng Việt, có dấu) vào cuối file “f.txt” mà không làm mất nội dung trước đó của file f. (khi mở file f lên phải đọc được Tiếng Việt có dấu)

Code:

path\_a = 'f.txt'

f = open(path\_a)

print(f.read())

l = 'Lập trình ứng dụng WEB'

*with* open(path\_a, *mode*='a') *as* f:

    f.write('\n' + l)

*with* open(path\_a) *as* f:

    print(f.read())

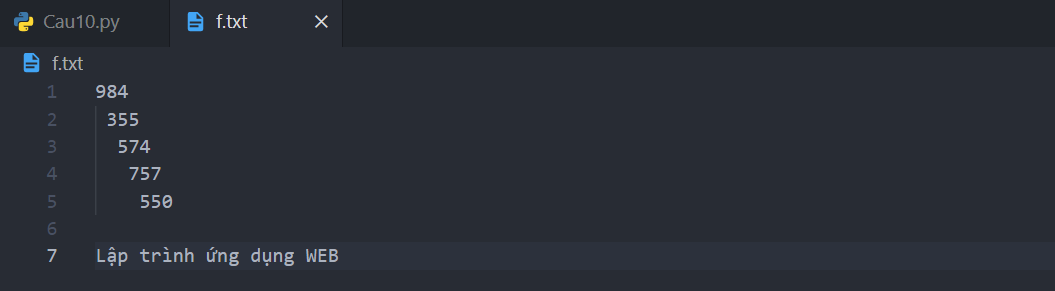
f.close()

Kết quả:

Ảnh có chứa văn bản

Mô tả được tạo tự động

1. Kết quả bài tập 10



1. File f.txt

## Cho danh sách (list) gồm 10 số nguyên được nhập từ bàn phím (có thể trùng nhau). Hãy tạo một danh sách y lấy các phần tử từ x không trùng nhau. Ghi thêm y vào file f.txt sao cho không làm mất nội dung trước đó của file f.txt.

Code:

n = int(input("Nhập số phần tử của dãy số : "))

mylist = []

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

   val = int(input('Nhập một số: '))

   mylist.append(val)

mylist2 = list(set(mylist))

path\_a = 'f.txt'

f = open(path\_a)

print(f.read())

*with* open(path\_a, *mode*='a') *as* f:

*for* item *in* mylist2:

        f.write('\n'+ str(item))

*with* open(path\_a) *as* f:

    print(f.read())

f.close()

Kết quả:

Ảnh có chứa văn bản

Mô tả được tạo tự động

1. Kết quả bài tập 4

## Cho file input.txt, mỗi dòng là các số nguyên từ 0 đến 10 000 000. Thực hiện đọc những số này và ghi kết quả vào file output.txt

Ví dụ:

|  |  |
| --- | --- |
| Input.txt | Input.txt |
| 125000  1100075 | một trăm hai mươi lăm nghìn một triệu một trăm nghìn không trăm bảy mươi lăm |

Code:

SoTuNhien = ['không', 'một', 'hai', 'ba', 'bốn', 'năm', 'sáu', 'bảy', 'tám', 'chín']

DonVi = ['mươi', 'mốt','hai', 'ba', 'bốn', 'lăm', 'sáu', 'bảy', 'tám', 'chín']

Chuc = ['linh', 'mười','hai', 'ba', 'bốn', 'lăm', 'sáu', 'bảy', 'tám', 'chín']

Lop = ['tỷ', 'triệu', 'nghìn']

Cum = ['trăm', 'chục', 'đơn vị']

def nhomCum3So(*so*):

    chieudai\_so = len(so)

    sodu, songuyen = chieudai\_so % len(Cum), chieudai\_so//len(Cum)

    list\_CumLe = so[:sodu]

    list\_CumChan = so[sodu:]

    list\_Cum3So = []

*if* list\_CumLe != "":

        list\_Cum3So = [list\_CumLe]

    begin = 0

    end = begin +3

*for* \_ *in* range(songuyen):

        list\_Cum3So.append(list\_CumChan[begin:end])

        begin = end

        end = begin +3

*return* list\_Cum3So

def themTienTo\_Lop(*list\_Cum3So*):

    so\_label = len(list\_Cum3So)

    soLop = len(Lop)

    list\_TienTo = []

*if* so\_label==1:

*return* list\_TienTo, list\_Cum3So

*elif* so\_label>1 and so\_label <= soLop:

        list\_TienTo = [\*Lop[-so\_label+1:]]

*else*:

        sochu\_ty = 0

*for* i\_label *in* range(so\_label):

*if* id\_Lop < (-soLop):

                id\_Lop =-1

                list\_TienTo.insert(0, Lop[id\_Lop])

                id\_Lop -= 1

*elif* i\_label !=0:

                list\_TienTo.insert(0, Lop[id\_Lop])

                id\_Lop -= 1

*if* list\_TienTo[0] == Lop[0]:

                    list\_TienTo[0] += (" "+Lop[0])\*sochu\_ty

                    sochu\_ty += 1

*return* list\_TienTo, list\_Cum3So

def phanCum\_themTienTo (*so*):

    list\_Cum3So = nhomCum3So(so)

    list\_TienTo, list\_Cum3So = themTienTo\_Lop(list\_Cum3So)

*return* list\_TienTo, list\_Cum3So

*# Đọc số có 2 chữ số*

def docSoChuc(*so2ChuSo*):

*if* so2ChuSo[0] == "0":

*return* "linh" +" "+ SoTuNhien[int(so2ChuSo[1])]

*# Xét các số hàng đơn vị*

*if* so2ChuSo[1] == "0":

*return* Chuc[1] *# Số 10*

*elif* so2ChuSo[1] == "1":

*return* Chuc[1] + " "+ SoTuNhien[1]

*else*:

*return* Chuc[1] + " " +DonVi[int(so2ChuSo[1])]

*else*:

*if* DonVi[0] == DonVi[int(so2ChuSo[1])]:

*return* SoTuNhien[int(so2ChuSo[0])] +" " + DonVi[int(so2ChuSo[1])]

*else*:

*return* SoTuNhien[int(so2ChuSo[0])] +" " + DonVi[0] +" "+ DonVi[int(so2ChuSo[1])]

def docSoTram(*so*):

*if* so[1:] == "00":

*return* SoTuNhien[int(so[0])]+" " +Cum[0]

*else*:

*return* SoTuNhien[int(so[0])]+" " +Cum[0]+" "+docSoChuc(so[1:])

def docso\_TungCum(*so*):

    chieudai\_so = len(so)

*if* chieudai\_so == 1:

*return* SoTuNhien[int(so)]

*elif* chieudai\_so == 2:

*return* docSoChuc(so)

*else*:

*return* docSoTram(so)

def doc\_Cum3so(*list\_cum3So*):

    kq\_DocSo =[]

*# Đọc các số theo từng lớp*

*for* lop *in* list\_cum3So:

*if* lop != "":

            kq\_DocSo.append(docso\_TungCum(lop))

*return* kq\_DocSo

def ghepCacLop(*list\_TienTo*, *ketqua\_DocCum*):

    ketQuaDocSo = ''

*if* list\_TienTo == []:

        ketQuaDocSo = ketqua\_DocCum[0]

*elif* list\_TienTo != []:

*for* i\_cum *in* range(len(ketqua\_DocCum)):

*if* i\_cum < len(list\_TienTo):

                ketQuaDocSo += (ketqua\_DocCum[i\_cum] + " " +list\_TienTo[i\_cum] + " ")

*else*: ketQuaDocSo +=  ketqua\_DocCum[i\_cum]

*else*:

        ketQuaDocSo += ketqua\_DocCum[0]

    ketQuaDocSo = ketQuaDocSo.capitalize()

*return* ketQuaDocSo

*# Đọc số*

def docso(*so*):

    list\_TienTo, list\_Cum3So = phanCum\_themTienTo(so)

    ketqua\_DocCum = doc\_Cum3so(list\_Cum3So)

    ketQuaDocSo = ghepCacLop(list\_TienTo, ketqua\_DocCum)

*return* ketQuaDocSo

path\_w = 'output.txt'

s = ''

mylist = []

f = open('input.txt', 'r', *encoding*='UTF-8')

data = f.read()

mylist = data.split("\n")

*for* i *in* mylist:

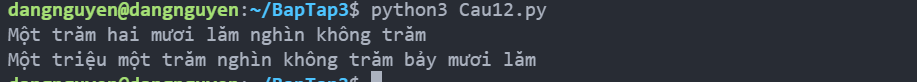
    print(docso(i))

    s += docso(i) + '\n'

*with* open(path\_w, *mode*='w') *as* f:

    f.write(s)

Kết quả:



1. Kết quả bài tập 4

Ảnh có chứa văn bản

Mô tả được tạo tự động

1. Kết quả bài tập 4

**HẾT**

1. Ghi nội dung công việc, các kịch bản trong bài Thực hành [↑](#footnote-ref-1)