

Tổng hợp 50 bài tập python cơ bản - nâng cao.

Giới thiệu:

Mong muốn giúp trẻ em Việt Nam phát triển tư duy phản biện, kỹ năng sáng tạo, nuôi dưỡng đam mê công nghệ. Học viện công nghệ TICA ra đời
www.tica.edu.vn

Các giảng viên là những người có nhiều năm kinh nghiệm ở các công ty phần mềm như FPT, CMC ...

Tham gia nhóm Phát triển tư duy công nghệ tại đây:
<https://www.facebook.com/groups/528679244566280>

Like fanpage của Học viện công nghệ TICA
<https://www.facebook.com/ticaacademy/>

Bài 1.

Cho một số nguyên, in "YES" nếu chữ số cuối của nó là 7 và in "NO" nếu không.

Ví dụ:

127 → YES

333 → NO

Bài 2.

Cho biết tọa độ của ba điểm A, B, C trên một đoạn thẳng. In khoảng cách từ điểm A đến điểm gần nó nhất.

Ví dụ:

10

35

30

Kết quả: 20

Bài 3.

Viết chương trình tính tổng các số trong một danh sách

Ví dụ:

Input: [2, 5, 8, 10, 12]

Output: Sum = 37

Bài 4.

Viết game đoán số may mắn (guess the number)

Máy tính nghĩ một số random từ 1 cho đến 15 và hỏi bạn đoán. Máy tính sẽ nói cho bạn khi bạn đoán sai là số may mắn là phải lớn hơn hoặc nhỏ hơn. Bạn sẽ chiến thắng nếu bạn đoán đúng số đó trong 5 lượt chơi.

Gợi ý:

```
import random
```

```
# random number from 1 to 15
```

```
random_number = random.randint(1, 15)
```

Máy tính sẽ tạo một số ngẫu nhiên từ 1 cho đến 15

Input/Output:

Tôi đang nghĩ một số giữa 1 và 15. Bạn hãy đoán số may mắn đó giúp tôi.

Số may mắn đó là: 10

Số bạn đoán phải nhỏ hơn 10

Số may mắn đó là: 2

Số bạn đoán phải lớn hơn: 2

Số may mắn đó là: 4

Chúc mừng bạn đã chiến thắng. Bạn đã đoán đúng sau 2 lượt chơi.

- Nếu sau 5 lượt chơi bạn không đoán được thì kết thúc chương trình và in ra
Bạn đã không may mắn, đây là số 4

Bài 5.

Một cửa hàng sẽ giảm giá 10% nếu tổng chi phí mua hàng lớn hơn 10.000
Người dùng về số lượng từ bàn phím
Giả sử, một đơn vị mặt hàng sẽ có giá 100 đồng.
In tổng chi phí hóa đơn cho người dùng.

Ví dụ:

Input: 120

Output: 10800

Input: 20

Output: 2000

Bài 6.

Viết chương trình in ra tất cả các số chẵn trong một danh sách

Ví dụ:

Input: [2, 5, 8, 10, 12]

Output: 2, 8, 10, 12

Bài 7.

Viết một chương trình tìm ra số lớn nhất của một danh sách mà không sử dụng hàm max()

Ví dụ:

Input: [2, 5, 8, 10, 12]

Output: 12

Bài 8.

Cho một danh sách các số, hãy tìm và in tất cả các phần tử lớn hơn phần tử trước đó.

Ví dụ1:

Input:

1 5 2 4 3

5 4

Ví dụ 2:

Input:

5 5 5 5 5

Output:

5 5 5 5

Bài 9.

Viết một chương trình xóa tất cả phần tử lặp lại (trùng lặp) ra khỏi danh sách.

Ví dụ:

Input: [1, 3, 5, 6, 3, 5, 6, 1]

Output: [1, 3, 5, 6]

Bài 10.

Get first, second best scores from the list.

List may contain duplicates.

Input: [86,86,85,85,85,83,23,45,84,1,2,0]

Output: should get 86, 85

Bài 11.

Given an array length 1 or more of ints, return the difference between the largest and smallest values in the array. Note: the built-in min(v1, v2) and max(v1, v2) functions return the smaller or larger of two values.

big_diff([10, 3, 5, 6]) → 7

big_diff([7, 2, 10, 9]) → 8

big_diff([2, 10, 7, 2]) → 8

Bài 12.

Viết một chương trình in ra các số chia hết cho 7 nhưng không chia hết cho 5 nằm trong khoảng 100 cho đến 1000 (tính cả 100 và 1000).

Kết quả: In trên một dòng và cách nhau bởi dấu phẩy.

Bài 13.

Viết một chương trình in tất cả các số nguyên tố nhỏ hơn n. Với n là số nguyên dương nhập từ bàn phím.

Ví dụ:

n = 12

Kết quả:

2 3 5 7 11

Bài 14.

Cho một danh sách số nguyên: [5, 10, 15, 20, 25, 46]

Tìm giá trị lớn nhất và nhỏ nhất của danh sách.

Gợi ý:

Không được sử dụng hàm max() và min()

Bài 15.

Viết chương trình nhập vào bán kính đường tròn, tính toán và in ra chu vi và diện tích hình tròn.

Bài 16.

Ask the user to enter a new password. Ask them to enter it again. If the two passwords match, display “Thank you”. If the letters are correct but in the wrong case, display the message “They must be in the same case”, otherwise display the message “Incorrect”

Bài 17.

Viết một chương trình nhập số nguyên dương n và tính tổng của các chữ số của số n.

Ví dụ:

Nếu người dùng nhập 3141 thì chương trình của bạn nên hiển thị $3 + 1 + 4 + 1 = 9$

Bài 18.

Ứng dụng chuyển đổi nhiệt độ từ độ C sang F, chuyển đổi từ kg sang pao(lb), diện tích, thể tích, tốc độ, thời gian (Easy)

<https://www.metric-conversions.org/vi/trong-luong/kilogam-sang-pao.htm>

Ý tưởng:

Xây dựng một chương trình gồm nhiều chức năng trên, người dùng chọn:

- chức năng 1: Chuyển đổi độ C sang độ F
- chức năng 2: Chuyển đổi độ F sang độ C
- chức năng 3: Chuyển đổi từ kg sang pao(lb)
- chức năng 4: Chuyển đổi từ pao(lb) sang kg
- chức năng 5: Chuyển đổi từ mét sang feet
- chức năng 6: Chuyển đổi từ feet sang meet
- chức năng 7: Thoát chương trình

Bài 19.

Làm trò chơi búa, đá, giấy chơi với máy tính

Ý tưởng:

https://youtu.be/eXQNi8hA_yM

Gợi ý:

Cách 1: Nhập số từ bàn phím

```
import random
```

```
choice = int(input("User turn: "))
```

```
random_choice = random.randint(1, 3)
```

Cách 2: Nhập chữ từ bàn phím

```
import random
```

```
options = ["Rock", "Paper", "Scissors"]
```

```
AI = random.choice(options)
```

Người chơi có 5 lượt chơi với máy tính, sau 5 lượt chơi thì thống kê người chơi đã thắng, hòa, thua bao nhiêu lượt với máy tính.

Bài 20.

Làm trò chơi đoán câu đố động vật (tên thủ đô các nước)

Ý tưởng:

<https://www.sciencekids.co.nz/quizzes/animal.html>

Chương trình hiển thị danh sách câu hỏi có sẵn, người dùng nhập câu trả lời.
Chương trình kiểm tra câu trả lời là đúng hay sai và đưa ra thông báo.

Người chơi có thể đoán sai tối đa 3 lần, nếu quá thì kết thúc trò chơi.

Bài 21.

Làm game chọn nhóm. Có một danh sách gồm 8 người chơi, hãy lựa chọn ngẫu nhiên 4 người chơi không trùng nhau để cho vào nhóm A, còn lại cho vào nhóm B.

Bài 22.

Tìm vị trí của giá trị chẵn đầu tiên trong mảng 1 chiều các số nguyên. Nếu mảng không có giá trị chẵn thì sẽ trả về -1

Bài 23.

Given the year number. You need to check if this year is a leap year. If it is, print LEAP, otherwise print COMMON.

The rules in Gregorian calendar are as follows:

a year is a leap year if its number is exactly divisible by 4 and is not exactly divisible by 100

a year is always a leap year if its number is exactly divisible by 400

Bài 24.

Viết chương trình tính tổng $S = 1 + 1/2 + 1/3 + \dots + 1/n$ với n là số nguyên dương nhập từ bàn phím.

Bài 25.

Liệt kê tất cả các ước số của số nguyên dương n .

Bài 26.

Phân tích một số thành tích các thừa số nguyên tố

Input: n = 120
Output: 2 2 2 3 5

Bài 27.

Given a non-empty string and an int n, return a new string where the char at index n has been removed. The value of n will be a valid index of a char in the original string (i.e. n will be in the range 0..len(str)-1 inclusive).

missing_char('kitten', 1) → 'ktten'
missing_char('kitten', 0) → 'itten'
missing_char('kitten', 4) → 'kittn'

Bài 28.

Given a string, return a new string where the first and last chars have been exchanged.

front_back('code') → 'eodc'
front_back('a') → 'a'
front_back('ab') → 'ba'

Bài 29.

You are driving a little too fast, and a police officer stops you. Write code to compute the result, encoded as an int value: 0=no ticket, 1=small ticket, 2=big ticket. If speed is 60 or less, the result is 0. If speed is between 61 and 80 inclusive, the result is 1. If speed is 81 or more, the result is 2. Unless it is your birthday -- on that day, your speed can be 5 higher in all cases.

caught_speeding(60, False) → 0
caught_speeding(65, False) → 1
caught_speeding(65, True) → 0

Bài 30.

Given 3 int values, a b c, return their sum. However, if one of the values is 13 then it does not count towards the sum and values to its right do not count. So for example, if b is 13, then both b and c do not count.

lucky_sum(1, 2, 3) → 6
lucky_sum(1, 2, 13) → 3
lucky_sum(1, 13, 3) → 1

Bài 31.

Given an array length 1 or more of ints, return the difference between the largest and smallest values in the array. Note: the built-in min(v1, v2) and max(v1, v2) functions return the smaller or larger of two values.

big_diff([10, 3, 5, 6]) → 7
big_diff([7, 2, 10, 9]) → 8
big_diff([2, 10, 7, 2]) → 8

Bài 32.

Tìm số chẵn cuối cùng trong mảng 1 chiều các số nguyên. Nếu mảng không có giá trị chẵn thì trả về -1

Bài 33.

Given a dictionary containing counts of both upvotes and downvotes, return what vote count should be displayed. This is calculated by subtracting the number of downvotes from upvotes.

Examples

get_vote_count({ "upvotes": 13, "downvotes": 0 }) → 13
get_vote_count({ "upvotes": 2, "downvotes": 33 }) → -31
get_vote_count({ "upvotes": 132, "downvotes": 132 }) → 0

Bài 34.

Sắp xếp mảng 1 chiều tăng dần

Input: 3 5 2 8 10 7 12
Output: 2 3 5 7 8 10 12

Lưu ý: Không sử dụng hàm sort()

Bài 35.

Đây là một chương trình tính tổng và trừ 2 số, các con tìm cách viết lại chương trình này bằng cách sử dụng hàm.

```
if command == "add":  
    print("lets add some numbers")  
    input1 = input("Number 1>")  
    input2 = input("Number 2>")  
    number1 = int(input1)  
    number2 = int(input2)  
    result = number1 + number2  
    output = str(result)  
    print(input1 + " + " + input2 + " = " + output)  
  
elif command == "subtract":  
    print("lets subtract some numbers")  
    input1 = input("Number 1>")  
    input2 = input("Number 2>")  
    number1 = int(input1)  
    number2 = int(input2)  
    result = number1 - number2  
    output = str(result)  
    print(input1 + " - " + input2 + " = " + output)
```

Bài 36.

Viết chương trình nhập vào 2 số nguyên dương a và b. Tìm ước số chung lớn nhất của a và b.

Ví dụ:

- Input:
 - a = 30

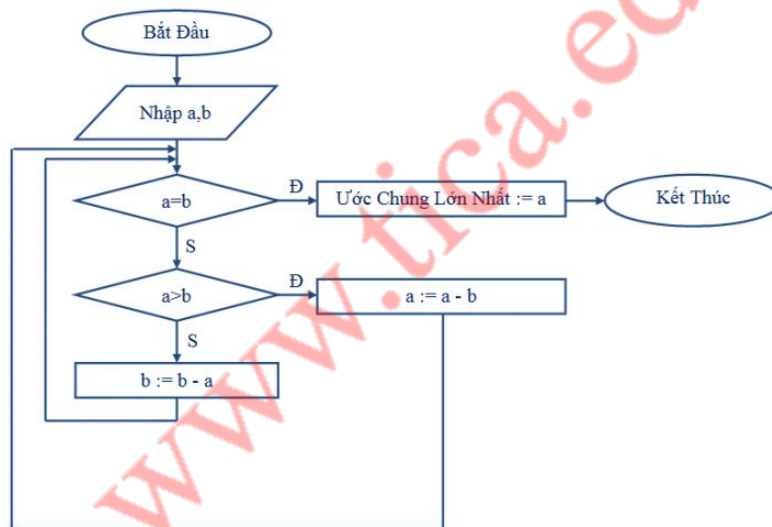
- $b = 40$
- Output:
 - $\text{UCLN} = 10$
 - $\text{BCNN} = 120$

Ước chung lớn nhất của hai số nguyên a và b là số nguyên dương lớn nhất mà a và b chia hết.

Bội số chung nhỏ nhất của hai số nguyên a và b là số nguyên dương nhỏ nhất chia hết cho cả a và b .

Nếu có số tự nhiên a chia hết cho số tự nhiên b thì ta gọi a là bội của b và b là ước của a .

SƠ ĐỒ KHỎI TÌM ƯỚC CHUNG LỚN NHẤT



Ví dụ:

Tìm ước chung lớn nhất của 27 và 45?

$$\text{UCLN}(27, 45) = 9$$

Bài 37.

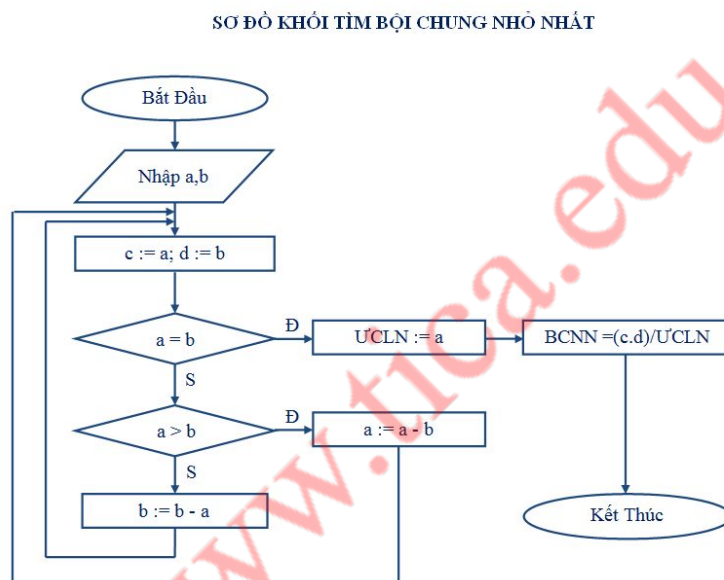
Viết chương trình nhập vào 2 số nguyên dương a và b. Tìm bội số chung nhỏ nhất của a và b.

Ví dụ:

$$\text{BCNN}(6,10) = 30$$

Bội số của 6: 6 12 18 24 **30** 36

Bội số của 10: 10 20 **30** 40



Bài 38.

Viết một hàm với đầu vào là tọa độ hai điểm trên mặt phẳng hai chiều và trả về độ dài đoạn thẳng nối hai điểm đó.

Ví dụ:

$$\text{line_length}([15, 7], [22, 11]) \rightarrow 8.06$$

$$\text{line_length}([0, 0], [0, 0]) \rightarrow 0$$

`line_length([0, 0], [1, 1]) → 1.41`

Bài 39.

Viết một hàm có tên `capital_indexes`. Hàm nhận một tham số duy nhất là một chuỗi. Hàm của bạn sẽ trả về một danh sách tất cả các chỉ số (index) trong chuỗi có chữ in hoa.

Ví dụ: gọi `capital_indexes("HeLlO")` sẽ trả về danh sách `[0, 2, 4]`.

Bài 40.

Kiểm tra đối xứng

Một chuỗi gọi là đối xứng khi đọc từ trái qua phải hay phải qua trái thì kết quả giống nhau.

Ví dụ: Chuỗi `"bob"` và chuỗi `"abba"` là đối xứng

Chuỗi `"abcd"` không phải đối xứng vì `"abcd" != "dcba"`.

Viết một hàm có tên `palindrome` kiểm tra tính đối xứng. Hàm trả `True` nếu đối xứng, `False` nếu không đối xứng.

Bài 41.

Hãy viết chương trình nhập vào số nguyên dương `n`. Kiểm tra xem `n` có phải là số chính phương hay không? (số chính phương là số khi lấy căn bậc 2 có kết quả là nguyên). Hãy viết chương trình kiểm tra số chính phương.

Bài 42.

Viết chương trình nhập vào số `n`.. xuất số đảo ngược của `n` đó..

Vd: `n = 123 => 321`
`n = 4320 → 0234`

Bài 43.

Cho số nguyên dương X , khi đảo ngược trật tự các chữ số của X ta sẽ thu được một số nguyên dương Y , Y được gọi là số đảo ngược của X .

Ví dụ: $X = 613$ thì $Y = 316$ là số đảo ngược của X .

Số nguyên dương Y được gọi là số nguyên tố nếu nó chỉ có hai ước số là 1 và chính nó, số 1 không phải là số nguyên tố.

Cho hai số nguyên dương P và Q ($1 \leq P \leq Q \leq 2109$; $Q - P \leq 105$).

Yêu cầu: Hãy tìm tất cả các số nguyên dương X nằm thỏa mãn $P \leq X \leq Q$ và số đảo ngược của số X là số nguyên tố.

Dữ liệu vào: Cho trong file văn bản TimSo.txt có cấu trúc như sau:

- Dòng 1: Ghi hai số nguyên dương P Q , hai số được ghi cách nhau ít nhất một dấu cách.

Dữ liệu ra: Ghi ra file văn bản KetQua.txt trên nhiều dòng, mỗi dòng ghi một số nguyên X tìm được

Ví dụ:

TimSo.txt

10 19

KetQua.txt

11

13

14

16

17

Bài 44.

Viết một chương trình chấp 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

Bài 45.

Write a function named `add_dots` that takes a string and adds "." in between each letter. For example, calling `add_dots("test")` should return the string "t.e.s.t".

Then, below the `add_dots` function, write another function named `remove_dots` that removes all dots from a string. For example, calling `remove_dots("t.e.s.t")` should return "test".

If both functions are correct, calling `remove_dots(add_dots(string))` should return back the original string for any string.

Bài 46.

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:

Input: hello world! 123

Thì đầu ra sẽ là:

Số chữ cái là: 10

Số chữ số là: 3

Bài 47.

Thiết kế trò chơi đoán từ vựng như chiếc nón kỳ diệu.

Máy tính sẽ hiện các ô chữ tương ứng với số chữ cái của từ bí mật.

Ví dụ:

Tên một loài động vật bơi ở biển có 4 chữ cái ?

Từ bí mật là Fish gồm 4 chữ cái

Welcome to Hangman!

— — — —
Guess your letter: I

Người dùng nhập từ ký tự muốn đoán. Sau đó chương trình sẽ hiện tất cả các ký tự đó nếu có trong từ bí mật.

 I

Trò chơi chỉ cho phép bạn đoán sai tối đa 5 lần, nếu quá 5 lần thì chương trình in ra từ bí mật.

Bài 48.

Đọc nội dung của một file INPUT.txt có cấu trúc như sau:

- Dòng đầu tiên ghi số lượng số nguyên có trong file
- Dòng tiếp theo là một dãy số nguyên

Ví dụ: Nội dung file INPUT.txt

```
8
1 2 3 4 5 6 7 8
```

Ghi vào file KETQUA.txt các số nguyên tố có trong mảng

Ví dụ: Kết quả file KETQUA.txt

```
1 3 5 7
```

Bài 49.

Cho một dữ liệu file input.txt

Hello! Welcome to Ha noi

Ha noi is the capital city of Vietnam

Good Luck!

Các con hãy lập trình đọc nội dung của file tìm ra từ đầu tiên có độ dài lớn nhất trong file trên. Các từ trong file sẽ cách nhau một khoảng trắng.

Kết quả: Từ Welcome có độ dài 7 ký tự.

Bài 50.

Viết một hàm đếm tần số xuất hiện mỗi từ trong một file. Ghi ra file trên mỗi dòng là từ và tần số xuất hiện của từ đó.

www.tica.edu.vn