



[Trang của tôi](#) / [Khoá học](#) / [Học kỳ II năm học 2021-2022 \(Semester 2 - Academic year 2021-2022\)](#)

/ [Chương Trình Chất Lượng Cao dạy bằng Tiếng Anh \(High-Quality training program\)](#)

/ [Khoa Khoa học và Kỹ thuật Máy tính \(Faculty of Computer Science and Engineering.\)](#) / [Khoa Học Máy Tính](#)

/ [Principles of Programming Languages \(CO3005\)_Nguyễn Hứa Phùng_\(CC_HK212\)](#) / 1-Introduction / [Python Programming Code CC03](#)

Câu hỏi 1

Chính xác

Chấm điểm của 1,00

Write a Python function, whose name is area, which accepts the radius of a circle as an input parameter and return the area

For example:

| Test | Result |
|-------------------------------------------------------------------------------------------------------------------------------------------|--------|
| <pre>res = area(1.1) expect = 3.8013271108436504 delta = 0.000000001 print((res > expect - delta) and (res < expect + delta))</pre> | True |

Answer: (penalty regime: 0 %)

```
1 import math
2 def area(rad):
3     return (rad ** 2) * math.pi
```

Kiểm tra

| | Test | Expected | Got | |
|---|----------------------------------------------------------------------------------------------------------------------------------------------|----------|------|---|
| ✓ | <pre>res = area(2.6) expect = 21.237166338267002 delta = 0.000000001 print((res > expect - delta) and (res < expect + delta))</pre> | True | True | ✓ |
| ✓ | <pre>res = area(12.6) expect = 498.75924968391547 delta = 0.000000001 print((res > expect - delta) and (res < expect + delta))</pre> | True | True | ✓ |
| ✓ | <pre>res = area(102.45) expect = 32974.164346060104 delta = 0.000000001 print((res > expect - delta) and (res < expect + delta))</pre> | True | True | ✓ |

Passed all tests! ✓

Câu hỏi **2**

Chính xác

Chăm điểm của 1,00

Write a Python function **check(lst,n)** to test whether all numbers of the list **lst** is greater than the number **n**.

For example:

| Test | Result |
|-----------------------------|--------|
| print(check([21,12,5,8],3)) | True |

Answer: (penalty regime: 0 %)

```
1 def check(l, num):
2     for i in l:
3         if num >= i:
4             return False
5
6     return True
```

Kiểm tra

| | Test | Expected | Got | |
|---|--------------------------------------|----------|-------|---|
| ✓ | print(check([21,12,5,8],3)) | True | True | ✓ |
| ✓ | print(check([21,12,5,8],7)) | False | False | ✓ |
| ✓ | print(check([21,12,5,8],22)) | False | False | ✓ |
| ✓ | print(check([21,12,1000,100,90],11)) | True | True | ✓ |
| ✓ | print(check([21,12,1000,100,90],12)) | False | False | ✓ |

Passed all tests! ✓

Câu hỏi **3**

Chính xác

Chăm điểm của 1,00

Write a Python function gcd to return the greatest common divisor (GCD) of two positive integer parameters

For example:

| Test | Result |
|-------------------|--------|
| print(gcd(24,36)) | 12 |

Answer: (penalty regime: 0 %)

```
1 def gcd(a, b):  
2     if (b == 0):  
3         return a  
4     else:  
5         return gcd(b, a % b)
```

Kiểm tra

| | Test | Expected | Got | |
|---|----------------------|----------|-----|---|
| ✓ | print(gcd(24,36)) | 12 | 12 | ✓ |
| ✓ | print(gcd(24,9)) | 3 | 3 | ✓ |
| ✓ | print(gcd(54,36)) | 18 | 18 | ✓ |
| ✓ | print(gcd(1071,462)) | 21 | 21 | ✓ |

Passed all tests! ✓

Câu hỏi **4**

Chính xác

Chăm điểm của 1,00

Write a Python program which accepts a sequence of comma-separated numbers from user and generate a list and a tuple with those numbers.

For example:

Input:

13,2,4,5

Output

[13,2,4,5]

(13,2,4,5)

For example:

| Input | Result |
|----------|------------------------------------------------|
| 13,2,4,5 | ['13', '2', '4', '5'] ('13', '2', '4', '5') |

Answer: (penalty regime: 0 %)

```

1 x = input()
2 num = x.split(',')
3 list = []
4 for n in num:
5     list.append(n)
6 print(list)
7 print(tuple(list))

```

Kiểm tra

| | Input | Expected | Got | |
|---|----------|------------------------------------------------|------------------------------------------------|---|
| ✓ | 13,2,4,5 | ['13', '2', '4', '5'] ('13', '2', '4', '5') | ['13', '2', '4', '5'] ('13', '2', '4', '5') | ✓ |
| ✓ | 12 | ['12'] ('12',) | ['12'] ('12',) | ✓ |

| | Input | Expected | Got | |
|---|-------------------|--------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|---|
| ✓ | 1,2 | ['1', '2'] ('1', '2') | ['1', '2'] ('1', '2') | ✓ |
| ✓ | 1, 3,4,15,22,16,7 | ['1', ' 3', '4', '15', '22', '16', '7'] ('1', ' 3', '4', '15', '22', '16', '7') | ['1', ' 3', '4', '15', '22', '16', '7'] ('1', ' 3', '4', '15', '22', '16', '7') | ✓ |

Passed all tests! ✓

Câu hỏi **5**

Chính xác

Chăm điểm của 1,00

Write a Python function **product(lst)** to return the product of the list **lst** of integers

For example:

| Test | Result |
|----------------------------|--------|
| print(product([3,4,7,11])) | 924 |

Answer: (penalty regime: 0 %)

```
1 def product(list):  
2     prod = 1  
3     for i in list:  
4         prod *= i  
5     return prod
```

Kiểm tra

| | Test | Expected | Got | |
|---|----------------------------|----------|-----|---|
| ✓ | print(product([3,4,7,11])) | 924 | 924 | ✓ |
| ✓ | print(product([3])) | 3 | 3 | ✓ |
| ✓ | print(product([3,4])) | 12 | 12 | ✓ |
| ✓ | print(product([3,4,7])) | 84 | 84 | ✓ |

Passed all tests! ✓

Câu hỏi **6**

Chính xác

Chấm điểm của 1,00

Write a Python function **sum_of_cube(n)** that takes a positive integer n and returns the sum of the cube of all the positive integers smaller than n.

For example:

| Test | Result |
|-----------------------|--------|
| print(sum_of_cube(8)) | 784 |

Answer: (penalty regime: 0 %)

```

1 def sum_of_cube(n):
2     sum = 0
3     while (n > 0):
4         n -= 1
5         sum += n ** 3
6     return sum

```

Kiểm tra

| | Test | Expected | Got | |
|---|------------------------|----------|------|---|
| ✓ | print(sum_of_cube(8)) | 784 | 784 | ✓ |
| ✓ | print(sum_of_cube(12)) | 4356 | 4356 | ✓ |
| ✓ | print(sum_of_cube(4)) | 36 | 36 | ✓ |
| ✓ | print(sum_of_cube(1)) | 0 | 0 | ✓ |

Passed all tests! ✓

◀ Python-introduction

Chuyển tới...

[Link Video Meet of session 4/1/2022 ▶](#)

Copyright 2007-2021 Trường Đại Học Bách Khoa - ĐHQG Tp.HCM. All Rights Reserved.

Địa chỉ: Nhà A1- 268 Lý Thường Kiệt, Phường 14, Quận 10, Tp.HCM.

Email: elearning@hcmut.edu.vn

Phát triển dựa trên hệ thống Moodle