



🏠 Trang chủ

Trang của tôi » Học kỳ I năm học 2018-2019 » Đại Học Chính Qui » Khoa Khoa học và Kỹ thuật Máy tính »  
Ng/lý ngôn ngữ lập trình (CO3005)\_Nguyễn Hứa Phùng (DH\_HK181) » Functional Programming » Bài kiểm tra FP

**Đã bắt đầu vào lúc** Thursday, 20 September 2018, 8:37 AM

**Tình trạng** Đã hoàn thành

**Hoàn thành vào lúc** Thursday, 20 September 2018, 8:56 AM

**Thời gian thực hiện** 19 phút 26 giây

**Điểm** 6,00 của 10,00 (60%)

**Câu hỏi 1**

Không trả lời

Chấm điểm của  
1,00

Chọn biểu thức thích hợp để đảo một danh sách (tức đảo List(3,2,4,1) thành List(1,4,2,3))? Chú ý tất cả các chọn lựa đều có lỗi hoặc về văn phạm hoặc về ngữ nghĩa (sai kiểu) nhưng hãy chọn biểu thức ĐÚNG nhất

Chọn một:

- ☐ a. List(3,2,4,1).foldRight(List())((x,y)⇒x::y)
- ☐ b. List(3,2,4,1).map((x,y)⇒(y,x))
- ☐ c. List(3,2,4,1).foldLeft(List())((x,y) ⇒ x::y)
- ☐ d. List(3,2,4,1).foldLeft(List())((x,y) ⇒ y::x)

**Câu hỏi 2**

Hoàn thành

Điểm 1,00 của 1,00

Cho biết Python dùng phép toán \*\* để tính lũy thừa, ví dụ 2\*\*3 sẽ có kết quả là 8. Hãy điền vào các chỗ trống dưới đây của hàm g và hàm f để có thể thực hiện các tính toán như mong muốn? Để đảm bảo so trùng khớp với đáp án, KHÔNG ghi một khoảng trắng nào khi điền. Lưu ý mỗi chỗ trống không ghi quá 4 ký tự.

def f(x):

def g(y):

return y\*\*x

return g

square = f(2)

print(square(3)) # kết quả là 9

cube = f(3)

print(cube(2)) # kết quả là 8

**Câu hỏi 3**

Hoàn thành

Điểm 1,00 của 1,00

Gọi **dist(x:Int,lst:List[Int])** là hàm phân phối x với mỗi phần tử của danh sách lst để tạo một danh sách có các phần tử là các cặp gồm x và 1 phần tử của danh sách lst. Ví dụ `dist(2,List(3,4,5))` sẽ trả về `List((2,3), (2,4), (2,5))`. Cho một phần của định nghĩa hàm **dist** viết đệ qui như sau:

```
def dist(x:Int,lst:List[Int]):List[(Int,Int)] = lst match {  
    case nil => nil // nil tương đương List()  
    -----  
}
```

Hãy chọn đáp án thích hợp nhất cho dòng để trống?

Chọn một:

- ☐ a. `case h::t => (x+t) :: dist(x,t)`
- ☒ b. `case h::t => (x,h) :: dist(x,t)`
- ☐ c. `case h::t => x :: h :: dist(x,t)`
- ☐ d. `case h::t => h :: dist(x,t)`

**Câu hỏi 4**

Hoàn thành

Điểm 0,00 của 1,00

Hãy dùng **reduce** của Python để nối tất cả phần tử của một danh sách có tên là **lst** gồm các số nguyên. Giả sử hàm cấp cao này đã được import được từ package `functools`. Cho biết hàm để đổi số nguyên thành chuỗi là `str()`, ví dụ `str(3)` là `"3"`, và hàm nối chuỗi là `+`, ví dụ `"a" + "b"` là `"ab"`.

Để đảm bảo so trùng, khi viết cần phải:

- KHÔNG có khoảng trắng, CHỈ sử dụng khi cần thiết (để phân cách giữa keywords và biến), khi dùng thì chỉ viết 1 khoảng trắng
- tên thông số cho hàm đặt theo thứ tự: x, y, z, t.
- Chuỗi qui ước dùng dấu nháy đơn `'`, ví dụ chuỗi rỗng `"`.
- Dòng lệnh không quá 35 ký tự

Trả lời: `reduce(lamda a,b:str(a)+str(b),lst)`

**Câu hỏi 5**

Hoàn thành

Điểm 1,00 của 1,00

Cho `s` có kiểu `IntStack` (immutable stack) và đang chứa một stack có 3 phần tử là 2, 5, 7 (7 ở đỉnh stack). Cho chuỗi biểu thức sau:

```
s.pop
```

```
val p = s.pop
```

Chọn một:

- ☐ a. Sau khi thực thi xong 2 biểu thức trên, giá trị của `s` là một stack có một phần tử là 2
- ☐ b. Sau khi thực thi xong 2 biểu thức trên, giá trị của `p` là một stack có một phần tử là 2
- ☒ c. Sau khi thực thi xong 2 biểu thức trên, giá trị của `s` vẫn không đổi (tức là một stack có 3 phần tử 2,5,7)
- ☐ d. Sau khi thực thi xong 2 biểu thức trên, giá trị của `s` là một stack có hai phần tử là 2 và 5

**Câu hỏi 6**

Hoàn thành

Điểm 1,00 của 1,00

Giả sử cần phải định nghĩa một hàm có tên **foo**, với hai thông số `x` kiểu `Int` và `y` kiểu `Float`, nhưng khi gọi hàm `foo` thì có thể gọi chỉ với 1 thông số là một số nguyên (ví dụ như `List(1.2f, 3.0f, 1.2f).map(foo(3))`). Hãy viết định nghĩa của hàm `foo` trên Scala? Chú ý tuân theo quy tắc viết sau để đảm bảo so trùng đúng:

- viết từ chữ `def` đến trước dấu `=` (không viết dấu `=` và thân hàm)
- chỉ có 1 khoảng trắng giữa chữ `def` và tên hàm, không có khoảng trắng nào khác trong định nghĩa
- tên hàm, tên, thứ tự và kiểu các thông số đúng theo đề bài đã nêu
- không cần viết kiểu trả về

Trả lời: **Câu hỏi 7**

Hoàn thành

Điểm 1,00 của 1,00

Hãy chọn lệnh gọi đúng và ngắn nhất để xác định liệu có một phần tử lẻ trong một danh sách nguyên `L` không?

Chọn một:

- ☐ a. `L.exists(x => x % 2 == 1)`
- ☐ b. `L.exists( % 2 == 1)`
- ☐ c. `L.exists((x:Int)=>x % 2 == 1)`
- ☒ d. `L.exists(_ % 2 == 1)`

**Câu hỏi 8**

Hoàn thành

Điểm 0,00 của 1,00

Gọi **dist(x:Int,lst:List[Int])** là hàm phân phối x với mỗi phần tử của danh sách lst để tạo một danh sách có các phần tử là các cặp gồm x và 1 phần tử của danh sách lst. Ví dụ `dist(2,List(3,4,5))` sẽ trả về `List((2,3), (2,4), (2,5))`.

Hãy viết THÂN của định nghĩa hàm dist?

```
def dist(x: Int, lst: List[Int]) =
```

Chú ý tuân thủ qui tắc viết để đảm bảo so trùng đúng:

- chỉ viết THÂN hàm tức từ sau dấu =
- không có khoảng trắng ở đầu và cuối
- NGẮN NHẤT (cố dùng simplifying) có thể

Trả lời: `lst match {case nil => nil; case h::t:`

**Câu hỏi 9**

Hoàn thành

Điểm 0,00 của 1,00

Python không có sẵn hàm compose nên phải hiện thực như dưới đây. Hãy điền vào chỗ trống để hiện thực hàm compose và các thông số khi gọi hàm compose. Lưu ý chỗ trống thứ nhất không quá 4 ký tự, trong khi chỗ trống thứ hai không quá 22 ký tự. **KHÔNG DÙNG KHOẢNG TRẮNG** khi điền vào chỗ trống.

```
def compose(*g):
    def h(args):
        return reduce(lambda x,y: x*y, reversed(g), args)

    return h

def square(x):
    return x * x

def increase(x):
    return x + 1

def double(x):
    return x * 2

m = compose(square(increase(double(x))))

print(m(5)) # Kết quả là 121 do (5 * 2 + 1) * (5 * 2 + 1)
```

**Câu hỏi 10**

Hoàn thành

Điểm 1,00 của 1,00

Gọi `accu` là một hàm nhận vào 2 thông số: một danh sách số nguyên `L` và một số nguyên `N`. Hàm `accu` sẽ trả về một danh sách là các cặp gồm 1 phần tử trong `L` và `N`. Ví dụ `accu(List(1,2,3),4)` sẽ trả về `List((1,4),(2,4),(3,4))`. Hãy chọn hiện thực đúng nhất của hàm `accu`?

Chọn một:

- ☒ a. `def accu(sl:List[Int],n:Int) = sl.map(_ ,n)`
- ☐ b. `def accu(sl:List[Int],n:Int) = sl.foldLeft(List(n))((x,y)=>x::y)`
- ☐ c. `def accu(sl:List[Int],n:Int) = sl.foldLeft(List(n))((x,y)=>y::x)`
- ☐ d. `def accu(sl:List[Int],n:Int) = sl.foldLeft(List(n))(_ :: _)`

**Copyright 2007-2014 BKĐT-Đại Học Bách Khoa Tp.HCM. All Rights Reserved.**

Địa chỉ: Nhà A1- 268 Lý Thường Kiệt, Phường 14, Quận 10, Tp.HCM. Email: [elearning@hcmut.edu.vn](mailto:elearning@hcmut.edu.vn)

Phát triển dựa trên hệ thống Moodle