

Bài 1: Cho hàm đệ quy để tính tổng các số từ 1 đến n. Hãy giải thích từng bước thực hiện của hàm đệ quy này khi $n = 7$

```
1 def sum_of_numbers(n):  
2     if n == 1:  
3         return 1  
4     else:  
5         return n + sum_of_numbers(n-1)  
6 print(sum_of_numbers(7))
```

Quy trình các bước như sau:

Bước 1: gọi hàm `sum_of_number(7)`

Kiểm tra điều kiện cơ sở: $n=7$, không thỏa mãn đk

Thực hiện câu lệnh trong else: `return 7 + sum_of_number(6)`

Bước 2: gọi hàm `sum_of_number(6)`

Kiểm tra điều kiện cơ sở: $n=6$, không thỏa mãn đk

Thực hiện câu lệnh trong else: `return 6 + sum_of_number(5)`

Bước3: gọi hàm `sum_of_number(5)`

Kiểm tra điều kiện cơ sở: $n=5$, không thỏa mãn đk

Thực hiện câu lệnh trong else: `return 5 + sum_of_number(4)`

Bước4: gọi hàm `sum_of_number(4)`

Kiểm tra điều kiện cơ sở: $n=4$, không thỏa mãn đk

Thực hiện câu lệnh trong else: `return 4 + sum_of_number(3)`

Bước5: gọi hàm `sum_of_number(3)`

Kiểm tra điều kiện cơ sở: $n=3$, không thỏa mãn đk

Thực hiện câu lệnh trong else: `return 3 + sum_of_number(2)`

Bước6: gọi hàm `sum_of_number(2)`

Kiểm tra điều kiện cơ sở: $n=2$, không thỏa mãn đk

Thực hiện câu lệnh trong else: `return 2 + sum_of_number(1)`

Bước7: gọi hàm `sum_of_number(1)`

Kiểm tra điều kiện cơ sở: $n=1$, thỏa mãn đk. hàm trả về 1

Các hàm được thực hiện lần lượt theo thứ tự từ dưới lên trên(từ b7->b1)

Sum_of_number(2) trả về $2+1 = 3 \Rightarrow$ Sum_of_number(3) trả về $3+3 = 6 \Rightarrow$ Sum_of_number(4) trả về $4+6 = 10 \Rightarrow$ Sum_of_number(5) trả về $5+10 = 15 \Rightarrow$ Sum_of_number(6) trả về $6+15 = 21 \Rightarrow$ Sum_of_number(7) trả về $7+21 = 28$

Vì vậy, kết quả cuối cùng là 28

Bài2: Cho hàm đệ quy để tính số Fibonacci thứ n. Hãy giải thích từng bước thực hiện của hàm đệ quy này khi $n = 8$.

```
1 * def fibonacci(n):  
2 *     if n <= 1:  
3 *         return n  
4 *     else:  
5 *         return fibonacci(n-1) + fibonacci(n-2)  
6 * print(fibonacci(8))
```

Quy trình các bước như sau:

Bước 1: gọi hàm fibonacci(8):

Kiểm tra điều kiện cơ sở: $n=8$, không thỏa mãn đk

Thực hiện câu lệnh trong else: fibonacci(7) + fibonacci(6)

Bước 2: gọi hàm fibonacci(7):

Kiểm tra điều kiện cơ sở: $n=7$, không thỏa mãn đk

Thực hiện câu lệnh trong else: fibonacci(6) + fibonacci(5)

Bước3 gọi hàm fibonacci(6):

Kiểm tra điều kiện cơ sở: $n=6$, không thỏa mãn đk

Thực hiện câu lệnh trong else: fibonacci(5) + fibonacci(4)

Bước4: gọi hàm fibonacci(5):

Kiểm tra điều kiện cơ sở: $n=5$, không thỏa mãn đk

Thực hiện câu lệnh trong else: fibonacci(4) + fibonacci(3)

Bước5: gọi hàm fibonacci(4):

Kiểm tra điều kiện cơ sở: $n=4$, không thỏa mãn đk

Thực hiện câu lệnh trong else: fibonacci(3) + fibonacci(2)

Bước6 gọi hàm fibonacci(3):

Kiểm tra điều kiện cơ sở: $n=3$, không thỏa mãn đk

Thực hiện câu lệnh trong else: fibonacci(2) + fibonacci(1)

Bước7: gọi hàm fibonacci(2):

Kiểm tra điều kiện cơ sở: $n=2$, không thỏa mãn đk

Thực hiện câu lệnh trong else: $\text{fibonacci}(1) + \text{fibonacci}(0)$

Bước8: gọi hàm fibonacci(1):

Kiểm tra điều kiện cơ sở: $n=1$, thỏa mãn đk. hàm trả về 1

Bước9: gọi hàm fibonacci(0):

Kiểm tra điều kiện cơ sở: $n=0$, thỏa mãn đk. hàm trả về 0

Các hàm được thực hiện lần lượt theo thứ tự từ dưới lên trên(từ b9->b1)

$\text{fibonacci}(2)$ trả về $1+0=1 \Rightarrow \text{fibonacci}(3)$ trả về $1+1=2 \Rightarrow \text{fibonacci}(4)$ trả về $2+1=3 \Rightarrow$
 $\text{fibonacci}(5)$ trả về $3+2=5 \Rightarrow \text{fibonacci}(6)$ trả về $5+3=8 \Rightarrow \text{fibonacci}(7)$ trả về $8+5=13 \Rightarrow$
 $\text{fibonacci}(8)$ trả về $13+8=21$

kết quả cuối cùng: $\text{fibonacci}(8) = 21$

Bài 3: Cho hàm đệ quy để tính x mũ n. Hãy giải thích từng bước thực hiện của hàm đệ quy này khi $x = 2$ và $n = 6$.

```
1 def power(x, n):  
2     if n == 0:  
3         return 1  
4     else:  
5         return x * power(x, n-1)  
6 print(power(2, 6))
```

Quy trình các bước như sau:

Bước 1: gọi hàm $\text{power}(2,6)$:

Vì $n \neq 0$, hàm trả về $2 \times \text{power}(2,5)$

Bước 2: : gọi hàm $\text{power}(2,5)$:

Vì $n \neq 0$, hàm trả về $2 \times \text{power}(2,4)$

Bước 3: gọi hàm $\text{power}(2,4)$:

Vì $n \neq 0$, hàm trả về $2 \times \text{power}(2,3)$

Bước 4: gọi hàm $\text{power}(2,3)$:

Vì $n \neq 0$, hàm trả về $2 \times \text{power}(2,2)$

Bước 5 : gọi hàm $\text{power}(2,2)$:

Vì $n \neq 0$, hàm trả về $2 \times \text{power}(2,1)$

Bước 6: gọi hàm $\text{power}(2,1)$:

Vì $n \neq 0$, hàm trả về $2 \times \text{power}(2,0)$

Bước 8: : gọi hàm $\text{power}(2,0)$:

Vì $n = 0$, hàm trả về 1

Các hàm được thực hiện lần lượt theo thứ tự từ dưới lên trên(từ b8->b1)

$\text{power}(2,1)$ trả về $2 \times 1 = 2 \Rightarrow \text{power}(2,2)$ trả về $2 \times 2 = 4 \Rightarrow \text{power}(2,3)$ trả về $2 \times 4 = 8 \Rightarrow \text{power}(2,4)$ trả về $2 \times 8 = 16 \Rightarrow \text{power}(2,5)$ trả về $2 \times 16 = 32 \Rightarrow \text{power}(2,6)$ trả về $2 \times 32 = 64$

kết quả cuối cùng: 64

Bài 4: Cho hàm đệ quy giải bài toán Tháp Hà Nội. Hãy giải thích từng bước thực hiện của hàm đệ quy này chuyển 4 đĩa từ cọc A sang cọc B, với trung gian là cọc C.

```
1- def thap_ha_noi(n, A, C, B):
2-     if n == 1:
3-         print("Chuyển đĩa 1 từ cột {A} sang cột {B}")
4-     else:
5-         thap_ha_noi(n-1, A, B, C)
6-         print(f"Chuyển đĩa {n} từ cột {A} sang cột {B}")
7-         thap_ha_noi(n-1, C, A, B)
8-
9- # Chuyển 4 đĩa từ cọc A sang cọc B, với trung gian là cọc C
10 thap_ha_noi(4, "A", "C", "B")
```

Quy trình các bước như sau:

Bước 1: gọi hàm $\text{thap_ha_noi}(4, A, B, C)$:

. $n=4$, không phải trường hợp cơ sở, nên ta thực hiện ba bước:

- Chuyển 3 đĩa từ A sang C sử dụng B làm trung gian
- Chuyển đĩa 4 từ A sang B
- Chuyển 3 đĩa từ C sang B sử dụng A làm trung gian

Bước 2: gọi hàm $\text{thap_ha_noi}(3, A, C, B)$:

. $n=3$, không phải trường hợp cơ sở, nên ta thực hiện ba bước:

- Chuyển 2 đĩa từ A sang B sử dụng C làm trung gian
- Chuyển đĩa 3 từ A sang C
- Chuyển 2 đĩa từ B sang C sử dụng A làm trung gian

Bước 3: gọi hàm $\text{thap_ha_noi}(2, A, B, C)$:

. $n=2$, không phải trường hợp cơ sở, nên ta thực hiện ba bước:

- Chuyển 1 đĩa từ A sang C sử dụng B làm trung gian
- Chuyển đĩa 2 từ A sang B
- Chuyển 1 đĩa từ C sang B sử dụng A làm trung gian

Bước 4: gọi hàm `thap_ha_noi(1, A, C, B)`:

. $n=1$, là trường hợp cơ sở, in ra 1 “chuyển đĩa 1 từ A sang C”

Quay lại:

Bước 5: gọi hàm `thap_ha_noi(2, A, B, C)`:

- Đã thực hiện chuyển 1 đĩa từ A sang C
- In ra “chuyển đĩa 2 từ A sang B”
- Thực hiện bước 3: `thap_ha_noi(1, C, B, A)`

Bước 6: gọi hàm `thap_ha_noi(1, C, B, A)`:

. $n=1$, là trường hợp cơ sở, in ra “chuyển đĩa 1 từ C sang B”

Quay lại:

Bước 7: gọi hàm `thap_ha_noi(3, A, C, B)`:

- Đã thực hiện chuyển 2 đĩa từ A sang B
- In ra “chuyển đĩa 3 từ A sang C”
- Thực hiện bước 3: `thap_ha_noi(2, B, C, A)`

Bước 8: : gọi hàm `thap_ha_noi(2, B, C, A)`:

. $n=2$, không phải trường hợp cơ sở, nên ta thực hiện ba bước:

- Chuyển 1 đĩa từ B sang A sử dụng C làm trung gian
- Chuyển đĩa 2 từ B sang C
- Chuyển 1 đĩa từ A sang C sử dụng B làm trung gian

Bước 9: gọi hàm `thap_ha_noi(1, B, C, A)`:

. $n=1$, là trường hợp cơ sở, in ra “chuyển đĩa 1 từ B sang A”

Quay lại:

Bước 10: gọi hàm `thap_ha_noi(2, B, C, A)`:

- Đã thực hiện chuyển 1 đĩa từ B sang A
- In ra “chuyển đĩa 2 từ B sang C”
- Thực hiện bước 3: `thap_ha_noi(1, A, C, B)`

Bước 11: gọi hàm `thap_ha_noi(1, A, C, B)`

. $n=1$, là trường hợp cơ sở, in ra “chuyển đĩa 1 từ A sang C”

Quay lại:

Bước 12: gọi hàm `thap_ha_noi` (4, A, B, C)

- Đã thực hiện chuyển 3 đĩa từ A sang C
- In ra “chuyển đĩa 4 từ A sang B”
- Thực hiện bước 3: `thap_ha_noi`(3, C, B, A)

Bước 13: gọi hàm `thap_ha_noi` (3, A, B, C)

. $n=3$, không phải trường hợp cơ sở, nên ta thực hiện ba bước:

- Chuyển 2 đĩa từ C sang A sử dụng B làm trung gian
- Chuyển đĩa 3 từ C sang B
- Chuyển 2 đĩa từ A sang B sử dụng C làm trung gian

Bước 14: gọi hàm `thap_ha_noi` (2, C, B, A)

. $n=2$, không phải trường hợp cơ sở, nên ta thực hiện ba bước:

- Chuyển 1 đĩa từ C sang B sử dụng A làm trung gian
- Chuyển đĩa 2 từ C sang A
- Chuyển 1 đĩa từ B sang A sử dụng C làm trung gian

Bước 15: gọi hàm `thap_ha_noi` (1, C, B, A)

. $n=1$, là trường hợp cơ sở, in ra “chuyển đĩa 1 từ C sang B”

Quay lại:

Bước 16: gọi hàm `thap_ha_noi`(2, C, A, B):

- Đã thực hiện chuyển 1 đĩa từ C sang B
- In ra “chuyển đĩa 2 từ C sang A”
- Thực hiện bước 3: `thap_ha_noi`(1, B, A, C)

Bước 17: gọi hàm `thap_ha_noi` (1, B, A, C)

. $n=1$, là trường hợp cơ sở, in ra “chuyển đĩa 1 từ B sang A”

Quay lại:

Bước 18: gọi hàm `thap_ha_noi`(3, C, A, B):

- Đã thực hiện chuyển 2 đĩa từ C sang A
- In ra “chuyển đĩa 3 từ C sang B”
- Thực hiện bước 3: `thap_ha_noi`(2, B, A, C)

Bước 19: gọi hàm `thap_ha_noi` (2, A, B, C)

. $n=2$, không phải trường hợp cơ sở, nên ta thực hiện ba bước:

- Chuyển 1 đĩa từ A sang C sử dụng B làm trung gian
- Chuyển đĩa 2 từ A sang B

- Chuyển 1 đĩa từ C sang B sử dụng A làm trung gian

Bước 20: gọi hàm `thap_ha_noi(1, A, C, B)`

. $n=1$, là trường hợp cơ sở, in ra “chuyển đĩa 1 từ A sang C”

Quay lại:

Bước 21: gọi hàm `thap_ha_noi(2, A, B, C)`:

- Đã thực hiện chuyển 1 đĩa từ A sang C
- In ra “chuyển đĩa 2 từ A sang B”
- Thực hiện bước 3: `thap_ha_noi(1, C, B, A)`

Bước 22: gọi hàm `thap_ha_noi(1, C, B, A)`

. $n=1$, là trường hợp cơ sở, in ra “chuyển đĩa 1 từ C sang B”

Như vậy, hàm đệ quy đã hoàn thành việc chuyển 4 đĩa từ cọc A sang cọc B sử dụng cọc C làm trung gian

Bài 5: Cho hàm đệ quy giải bài toán cổ vừa gà vừa chó. Hãy giải thích từng bước thực hiện của hàm đệ quy của bài toán này.

```
1* def cho_ga(tong_so_con, tong_so_chan):
2*     if tong_so_con == 0 and tong_so_chan == 0:
3*         return 0, 0
4*     if tong_so_chan % 2 != 0:
5*         return -1, -1
6*     for cho in range(tong_so_con + 1):
7*         ga = tong_so_con - cho
8*         if ga * 2 + cho * 4 == tong_so_chan:
9*             return cho, ga
10*     cho, ga = cho_ga(tong_so_con - 1, tong_so_chan - 4)
11*     if ga != -1:
12*         return cho + 1, ga
13*     else:
14*         return -1, -1
15
16 tong_so_chan = 100
17 tong_so_con = 36
18 so_cho, so_ga = cho_ga(tong_so_con, tong_so_chan)
19 print("Số gà là:", so_ga)
20 print("Số chó là:", so_cho)
```

Quy trình các bước như sau:

Bước 1: Gọi hàm `cho_ga(36, 100)`.

Bước 2: Kiểm tra điều kiện cơ sở: $\text{tong_so_con} = 36$ và $\text{tong_so_chan} = 100$.

Không thoả mãn điều kiện cơ sở.

Bước 3: Kiểm tra điều kiện $\text{tong_so_chan} \% 2 \neq 0$.

- Kiểm tra xem tổng số chân có phải là số chẵn hay không.

- Trong trường hợp không phải số chẵn, không thể tạo ra một phân phối hợp lệ của gà và chó. Hàm sẽ trả về -1, -1.

Bước 4: Duyệt qua tất cả các lựa chọn có thể cho số lượng gà và chó:

Với mỗi lựa chọn số lượng chó (cho) từ 0 đến `tong_so_con`, ta tính số lượng gà tương ứng (ga) bằng cách trừ số lượng chó từ `tong_so_con`.

Kiểm tra xem có tổng số chân của gà và chó có bằng `tong_so_chan` không. Nếu có, trả về số lượng chó và gà tương ứng.

Bước 5: Nếu không tìm thấy một phân phối hợp lệ trong vòng lặp for, tiếp tục tìm kiếm bằng cách giảm số lượng con và số chân đi một đơn vị.

Gọi đệ quy cho `ga(tong_so_con - 1, tong_so_chan - 4)`.

Nếu tìm được một giải pháp hợp lệ, trả về số lượng chó và gà tương ứng

Nếu không, trả về -1, -1.

In ra kết quả: số gà: 22 và số chó:14

MAI CHI