

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))
```

-Bước 1: gọi hàm `sum_of_numbers(7)`.

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

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

-Bước 2: hàm `sum_of_numbers(6)` được gọi

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

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

-Bước 3: hàm `sum_of_numbers(5)` được gọi

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

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

-Bước 4: hàm `sum_of_numbers(4)` được gọi

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

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

-Bước 5: hàm `sum_of_numbers(3)` được gọi

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

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

-Bước 6: hàm `sum_of_numbers(2)` được gọi

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

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

-Bước 7: hàm `sum_of_numbers(1)` được gọi

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

Các hàm được thực hiện lần lượt theo thứ tự từ dưới lên trên(bước 7 lên bước 1)

`Sum_of_numbers(7)` trả về $1+2+3+4+5+6+7=28$

Vậy kết quả cuối cùng là 28, đây là tổng các số từ 1 đến 7

Bài 2: 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))
```

-Bước 1: gọi hàm Fibonacci(8)

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

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

-Bước 2: hàm fibonacci(7) được gọi

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

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

-Bước 3: hàm fibonacci(6) được gọi

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

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

-Bước 4: hàm fibonacci(5) được gọi

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

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

-Bước 5: hàm fibonacci(4) được gọi

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

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

-Bước 6: hàm fibonacci(3) được gọi

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

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

-Bước 7: hàm fibonacci(2) được gọi

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

Thực hiện câu lệnh trong else: return Fibonacci(1)+Fibonacci(0)

-Bước 8: hàm fibonacci(1) được gọi

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

-Bước 9: hàm fibonacci(0) được gọi

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

Các bước được thực hiện lần lượt từ dưới lên trên(bước 9 đến bước1)

Fibonacci(0) trả về 0 \rightarrow fibonacci(1) trả về 1 \rightarrow fibonacci(2) trả về $0+1=1$ \rightarrow fibonacci(3) trả về $1+1=2$

Finonacci(4) trả về $1+2=3$ \rightarrow fibonacci(5) trả về $3+2=5$ \rightarrow fibonacci(6) trả về $3+5=8$ \rightarrow fibonacci(7) trả về $5+8=13$ \rightarrow fibonacci(8) trả về $8+13=21$

Vậy kết quả trả về là 21, đây là số fibonacci thứ 8 trong chuỗi

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))
```

-Bước 1: gọi hàm power(2,6)

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

Thực hiện câu lệnh else: return $2 * \text{power}(2,5)$

-Bước 2: hàm power(2,5) được gọi

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

Thực hiện câu lệnh else: return $2 * \text{power}(2,4)$

-Bước 3: hàm power(2,4) được gọi

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

Thực hiện câu lệnh else: return $2 * \text{power}(2,3)$

-Bước 4: hàm power(2,3) được gọi

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

Thực hiện câu lệnh else: return $2 * \text{power}(2,2)$

-Bước 5: hàm power(2,2) được gọi

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

Thực hiện câu lệnh else: return $2 * \text{power}(2,1)$

-Bước 6: hàm `power(2,1)` được gọi

Kiểm tra điều kiện cơ sở $n=1$, không thoả mãn điều kiện cơ sở

Thực hiện câu lệnh `else: return 2* power(2,1)`

-Bước 7: hàm `power(2,0)` được gọi

Kiểm tra điều kiện cơ sở $n=0$, thoả mãn điều kiện cơ sở. Hàm trả về 1

Các bước thực hiện lần lượt từ dưới lên trên(bước 7 lên bước 1)

`Power(2,1)` trả về $2*1=2 \rightarrow \text{power}(2,2) trả về $2*\text{power}(2,1)=4 \rightarrow \text{power}(2,3)$ trả về $2*\text{power}(2,2)=8 \rightarrow \text{power}(2,4)$ trả về $2*\text{power}(2,3)=16 \rightarrow \text{power}(2,5)$ trả về $2*\text{power}(2,4)=32 \rightarrow \text{power}(2,6)$ trả về $2*\text{power}(2,5)=64$$

Vậy cuối cùng kết quả trả về là 64, là kết quả của 2 mũ 6

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(f"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")
```

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

Kiểm tra điều kiện `if`, không thoả mãn điều kiện

Thực hiện câu lệnh `else`

Chia nhỏ 2 bước hơn:

Bước 1.1: `thap_ha_noi(3,"A","B","C")`

Bước 1.2: in ra " chuyển đĩa 4 từ cột A sang cột B"

Bước 1.3: `thap_ha_noi(3,"C","A","B")`

Bước 1.4: hàm `thap_ha_noi(3,"A","B","C")` được gọi

Kiểm tra điều kiện, không thoả mãn điều kiện

Thực hiện câu lệnh `else`

Chia 2 bước nhỏ tiếp

Bước 1.4.1: `thap_ha_noi(2,"A","B","C")`

Bước 1.4.2: in ra "chuyển đĩa 3 từ cột A sang cột B"

Bước 1.4.3: `thap_ha_noi(2,"C","B","A")`

Gọi hàm đến khi nào thoả mãn điều kiện cơ sở thì dừng lại

Bước 1.4.4: hàm `thap_ha_noi(2,"A","B","C")` được gọi

Bước 1.4.4.1: `thap_ha_noi(1,"A","B","C")`

Bước 1.4.4.2: in ra "chuyển đĩa 2 từ cột A sang cột B"

Bước 1.4.4.3: `thap_ha_noi(1,"C","B","A")`

Bước 1.4.4.4: gọi `thap_ha_noi(1,"A","B","C")`

Lúc này $n=1$ thoả mãn điều kiện cơ sở nên in ra "chuyển đĩa 1 từ cột A sang cột B"

Vậy bài toán kết thúc.

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)
```

Bước 1: gọi hàm `cho_ga(100,36)`

Kiểm tra điều kiện $tong_so_con=0$ và $tong_so_chan=0$, không thoả mãn điều kiện (vì nếu bằng 0 thì không có chó và gà)

Bước 2: kiểm tra điều kiện $tong_so_chan \% 2 \neq 0$, không thoả mãn điều kiện (vì số chân của chó là 4 số chân của gà là 2 nó đều là số chẵn nên không thể không chia hết cho 2)

Bước 3: thực hiện vòng lặp cho lặp từ 0 đến 36

$Ga = tong_so_con - cho$

Sau đó thực hiện lệnh if $ga*2+cho*4=100$ thì dừng lại

Ví dụ: $cho=0, ga=36 \rightarrow 36*2=72$ (không $=100$)

Cho đến $cho=14, ga=22 \rightarrow 22*2+14*4=100$ (tong_so_chan)

Bước 4: nếu bước 3 không ra kết quả thì gọi hàm $cho_ga=cho_ga(tong_so_con-1, tong_so_chan-4)$

Giảm số con đi 1 và số chân đi 4

Nếu kết quả trả về không phải -1, thì tăng số chó lên 1 giữ nguyên số gà

Vậy kết quả cuối cùng của bài toán là chó=14 con, gà=22 con

TRAN_THI_THU_HUYEN