

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_numbers(7)`:

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

Thực hiện câu lệnh `else: return n + 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

Thực hiện câu lệnh `else: return n + 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

Thực hiện câu lệnh `else: return n + 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

Thực hiện câu lệnh `else: return n + 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

Thực hiện câu lệnh else: return n + 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

Thực hiện câu lệnh else: return n + sum_of_numbers(1)

Bước 7: Hàm sum_of_numbers(1) được gọi:

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

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

sum_of_numbers(2) trả về $2 + 1 = 3$ -> sum_of_numbers(3) trả về $3 + 3 = 6$ ->

sum_of_numbers(4) trả về $6 + 4 = 10$ -> sum_of_numbers(5) trả về $10 + 5 = 15$ ->

sum_of_numbers(6) trả về $15 + 6 = 21$ -> sum_of_numbers(7) trả về $21 + 7 = 28$

Vậy: kết quả cuối cùng là 8, đây là tổng 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))
```

Quy trình các bước:

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

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

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

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

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

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

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

Bước 8: Hàm fibonacci (1) được gọi

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

Bước 9: Hàm fibonacci (0) được gọi:

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

Các hàm được thực hiện lần lượt theo thứ tự từ dưới lên:

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

Vậy: Kết quả cuối cùng là 21, đây là số Fibonacci thứ 8.

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 $n = 6$.

Lê Hằng Anh – Bài tập về nhà lab 9

```
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 `power(2,6)`

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

Thực hiện câu lệnh trong `else`: `2*power(2, 5)`

Bước 2: Hàm `power(2,5)` được gọi

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

Thực hiện câu lệnh trong `else`: `2*power(2, 4)`

Bước 2: Hàm `power(2,5)` được gọi

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

Thực hiện câu lệnh trong `else`: `2*power(2, 4)`

Bước 3: Hàm $\text{power}(2,4)$ được gọi

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

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

Bước 4: Hàm $\text{power}(2,3)$ được gọi

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

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

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

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

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

Bước 6: Hàm $\text{power}(2,1)$ được gọi

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

Thực hiện câu lệnh trong else: $2 * \text{power}(2, 0)$

Bước 7: Hàm $\text{power}(2,0)$ được gọi

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

Các hàm được thực hiện lần lượt theo thứ tự từ dưới lên:

- $\text{power}(2,1) = 2 * \text{power}(2,0) = 2 * 1 = 2$
- $\text{power}(2,2) = 2 * \text{power}(2,1) = 2 * 2 = 4$
- $\text{power}(2,3) = 2 * \text{power}(2,2) = 2 * 4 = 8$
- $\text{power}(2,4) = 2 * \text{power}(2,3) = 2 * 8 = 16$
- $\text{power}(2,5) = 2 * \text{power}(2,4) = 2 * 16 = 32$
- $\text{power}(2,6) = 2 * \text{power}(2,5) = 2 * 32 = 64$

Vậy: Kết quả cuối cùng của 2 mũ 6 là 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(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")
```

Quy trình các bước:

Bước 1: Gọi hàm thap_ha_noi (4, “A”, “C”, “B”)

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

Thực hiện câu lệnh else: hàm thap_ha_noi (3, “A”, “B”, “C”)

Bước 2: Hàm thap_ha_noi (3, “A”, “B”, “C”) được gọi:

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

Thực hiện câu lệnh else: thap_ha_noi (2, “A”, “C”, “B”)

Bước 3: Hàm thap_ha_noi (2, “A”, “C”, “B”) được gọi

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

Thực hiện câu lệnh else: thap_ha_noi (1, “A”, “B”, “C”)

Bước 4: Hàm thap_ha_noi (1, “A”, “B”, “C”) được gọi

Kiểm tra điều kiện cơ sở: $n = 1$, điều kiện thỏa mãn.

In ra “Chuyển đĩa 1 từ cột A sang C”.

Bước 5: Trở lại hàm `thap_ha_noi` (2, “A”, “C”, “B”)

In ra “Chuyển đĩa 2 từ cột A sang cột B”.

Gọi hàm `thap_ha_noi` (1, “C”, “A”, “B”)

Bước 6: Hàm `thap_ha_noi` (1, “C”, “A”, “B”) được gọi

Kiểm tra điều kiện cơ sở: $n = 1$, điều kiện thỏa mãn.

In ra “Chuyển đĩa 1 từ cột C sang B”.

Bước 7: Sau khi hoàn thành hàm `thap_ha_noi` (2, “A”, “C”, “B”).

Trở lại hàm `thap_ha_noi` (3, “A”, “B”, “C”)

In ra “Chuyển đĩa 3 từ cột A sang cột C”.

Gọi hàm `thap_ha_noi` (2, “B”, “A”, “C”)

Bước 8: Hàm `thap_ha_noi` (2, “B”, “A”, “C”) được gọi:

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

Thực hiện câu lệnh `else`: `thap_ha_noi` (1, “B”, “C”, “A”)

Bước 9: Hàm `thap_ha_noi` (1, “B”, “C”, “A”) được gọi

Kiểm tra điều kiện cơ sở: $n = 1$, điều kiện thỏa mãn.

In ra “Chuyển đĩa 1 từ cột B sang A”.

Bước 10: Trở lại hàm `thap_ha_noi`(2, “B”, “A”, “C”)

In ra “Chuyển đĩa 2 từ cột B sang cột C”.

Gọi hàm `thap_ha_noi`(1, “C”, “A”, “B”)

Bước 11: Hàm `thap_ha_noi` (1, “C”, “A”, “B”) được gọi

Kiểm tra điều kiện cơ sở: $n = 1$, điều kiện thỏa mãn.

In ra “Chuyển đĩa 1 từ cột A sang C”.

Bước 12: Sau khi hoàn thành hàm `thap_ha_noi`(3, “A”, “B”, “C”).

Trở lại hàm `thap_ha_noi` (4, “A”, “C”, “B”)

In ra “Chuyển đĩa 4 từ cột A sang cột B”.

Gọi hàm `thap_ha_noi` (3, “C”, “A”, “B”)

Bước 13: Hàm `thap_ha_noi` (3, “C”, “A”, “B”) được gọi:

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

Thực hiện câu lệnh `else`: `thap_ha_noi` (2, “C”, “B”, “A”)

Bước 14: Hàm `thap_ha_noi` (2, “C”, “B”, “A”) được gọi

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

Thực hiện câu lệnh `else`: `thap_ha_noi` (1, “C”, “A”, “B”)

Bước 15: Hàm `thap_ha_noi` (1, “C”, “A”, “B”) được gọi

Kiểm tra điều kiện cơ sở: $n = 1$, điều kiện thỏa mãn.

In ra “Chuyển đĩa 1 từ cột C sang B”.

Bước 16: Trở lại hàm `thap_ha_noi` (2, “C”, “B”, “A”)

In ra “Chuyển đĩa 2 từ cột C sang cột A”.

Gọi hàm `thap_ha_noi` (1, “B”, “C”, “A”)

Bước 17: Hàm `thap_ha_noi` (1, “B”, “C”, “A”) được gọi

Kiểm tra điều kiện cơ sở: $n = 1$, điều kiện thỏa mãn.

In ra “Chuyển đĩa 1 từ cột B sang A”.

Bước 18: Sau khi hoàn thành hàm `thap_ha_noi` (2, “C”, “B”, “A”)

Trở lại hàm `thap_ha_noi` (3, “C”, “A”, “B”)

In ra “Chuyển đĩa 3 từ cột C sang cột B”.

Gọi hàm `thap_ha_noi` (2, “A”, “C”, “B”).

Bước 19: Hàm `thap_ha_noi` (2, “A”, “C”, “B”) được gọi:

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

Thực hiện câu lệnh `else`: `thap_ha_noi` (1, “A”, “B”, “C”).

Bước 20: Hàm `thap_ha_noi` (1, “A”, “B”, “C”) được gọi

Kiểm tra điều kiện cơ sở: $n = 1$, điều kiện thỏa mãn.

In ra “Chuyển đĩa 1 từ cột A sang C”.

Bước 21: Trở lại hàm `thap_ha_noi` (2, “A”, “C”, “B”)

In ra “Chuyển đĩa 2 từ cột A sang cột B”.

Gọi hàm `thap_ha_noi` (1, “C”, “A”, “B”)

Bước 22: Hàm `thap_ha_noi` (1, “C”, “A”, “B”) được gọi

Kiểm tra điều kiện cơ sở: $n = 1$, điều kiện thỏa mãn.

In ra “Chuyển đĩa 1 từ cột C sang B”.

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)
```

Lê Hằng Anh – Bài tập về nhà lab 9

Quy trình các bước như sau:

Bước 1: Gọi hàm cho_ga (36, 100)

Kiểm tra điều kiện: $tong_so_con = 36$ and $tong_so_chan = 100$, không thỏa mãn điều kiện

Kiểm tra $tong_so_chan \% 2 \neq 0$, không thỏa mãn điều kiện

Khởi tạo vòng lặp với biến cho chạy từ 0 đến 36.

Bước 2: Bắt đầu vòng lặp

Với biến cho chạy từ 0 đến 36, ta có: $ga = 36 - cho$

Kiểm tra điều kiện: $ga * 2 + cho * 4 == tong_so_chan$, không có giá trị ga và cho thỏa mãn điều kiện.

Bước 3: Hàm cho_ga (35, 96) được gọi

Kiểm tra điều kiện: $\text{tong_so_con} = 35$ and $\text{tong_so_chan} = 96$, không thỏa mãn điều kiện

Kiểm tra $\text{tong_so_chan} \% 2 \neq 0$, không thỏa mãn điều kiện

Khởi tạo vòng lặp với biến cho chạy từ 0 đến 35.

Bước 4: Bắt đầu vòng lặp

Với biến cho chạy từ 0 đến 35, ta có: $\text{ga} = 35 - \text{cho}$

Kiểm tra điều kiện: $\text{ga} * 2 + \text{cho} * 4 == \text{tong_so_chan}$, không có giá trị ga và cho thỏa mãn điều kiện.

Bước 3: Hàm cho_ga(35, 96) được gọi

Kiểm tra điều kiện: $\text{tong_so_con} = 35$ and $\text{tong_so_chan} = 96$, không thỏa mãn điều kiện

Kiểm tra $\text{tong_so_chan} \% 2 \neq 0$, không thỏa mãn điều kiện

Khởi tạo vòng lặp với biến cho chạy từ 0 đến 36.

Bước 4: Bắt đầu vòng lặp

Với biến cho chạy từ 0 đến 35, ta có: $\text{ga} = 35 - \text{cho}$

Kiểm tra điều kiện: $\text{ga} * 2 + \text{cho} * 4 == \text{tong_so_chan}$, không có giá trị ga và cho thỏa mãn điều kiện.

Bước 5: Hàm cho_ga (34, 92) được gọi

Kiểm tra điều kiện: $\text{tong_so_con} = 34$ and $\text{tong_so_chan} = 92$, không thỏa mãn điều kiện

Kiểm tra $\text{tong_so_chan} \% 2 \neq 0$, không thỏa mãn điều kiện

Khởi tạo vòng lặp với biến cho chạy từ 0 đến 34.

Bước 6: Bắt đầu vòng lặp

Với biến cho chạy từ 0 đến 34, ta có: $ga = 34 - cho$

Kiểm tra điều kiện: $ga*2 + cho*4 == tong_so_chan$, không có giá trị ga và cho thỏa mãn điều kiện.

Bước 7: Hàm cho_ga(33, 88) được gọi

Kiểm tra điều kiện: $tong_so_con = 33$ and $tong_so_chan = 88$, không thỏa mãn điều kiện

Kiểm tra $tong_so_chan \% 2 != 0$, không thỏa mãn điều kiện

Khởi tạo vòng lặp với biến cho chạy từ 0 đến 33.

Bước 8: Bắt đầu vòng lặp

Với biến cho chạy từ 0 đến 33, ta có: $ga = 33 - cho$

Kiểm tra điều kiện: $ga*2 + cho*4 == tong_so_chan$, không có giá trị ga và cho thỏa mãn điều kiện.

Bước 9: Hàm cho_ga(32, 84) được gọi

Kiểm tra điều kiện: $tong_so_con = 32$ and $tong_so_chan = 84$, không thỏa mãn điều kiện

Kiểm tra $tong_so_chan \% 2 != 0$, không thỏa mãn điều kiện

Khởi tạo vòng lặp với biến cho chạy từ 0 đến 32.

Bước 10: Bắt đầu vòng lặp

Với biến cho chạy từ 0 đến 32, ta có: $ga = 32 - cho$

Kiểm tra điều kiện: $ga*2 + cho*4 == tong_so_chan$, không có giá trị ga và cho thỏa mãn điều kiện.

Bước 11: Hàm cho_ga(31, 80) được gọi

Kiểm tra điều kiện: $tong_so_con = 31$ and $tong_so_chan = 80$, không thỏa mãn điều kiện

Kiểm tra $tong_so_chan \% 2 \neq 0$, không thỏa mãn điều kiện

Khởi tạo vòng lặp với biến cho chạy từ 0 đến 31.

Bước 12: Bắt đầu vòng lặp

Với biến cho chạy từ 0 đến 31, ta có: $ga = 31 - cho$

Kiểm tra điều kiện: $ga*2 + cho*4 == tong_so_chan$, không có giá trị ga và cho thỏa mãn điều kiện.

Bước 13: Hàm cho_ga(30, 76) được gọi

Kiểm tra điều kiện: $tong_so_con = 30$ and $tong_so_chan = 76$, không thỏa mãn điều kiện

Kiểm tra $tong_so_chan \% 2 \neq 0$, không thỏa mãn điều kiện

Khởi tạo vòng lặp với biến cho chạy từ 0 đến 30.

Bước 14: Bắt đầu vòng lặp

Với biến cho chạy từ 0 đến 30, ta có: $ga = 30 - cho$

Kiểm tra điều kiện: $ga*2 + cho*4 == tong_so_chan$, không có giá trị ga và cho thỏa mãn điều kiện.

Bước 15: Hàm cho_ga (29, 72) được gọi

Kiểm tra điều kiện: $tong_so_con = 29$ and $tong_so_chan = 72$, không thỏa mãn điều kiện

Kiểm tra $tong_so_chan \% 2 \neq 0$, không thỏa mãn điều kiện

Khởi tạo vòng lặp với biến cho chạy từ 0 đến 29.

Bước 16: Bắt đầu vòng lặp

Với biến cho chạy từ 0 đến 29, ta có: $ga = 29 - cho$

Kiểm tra điều kiện: $ga*2 + cho*4 == tong_so_chan$, không có giá trị ga và cho thỏa mãn điều kiện.

Bước 17: Hàm cho_ga (28, 68) được gọi

Kiểm tra điều kiện: $tong_so_con = 28$ and $tong_so_chan = 68$, không thỏa mãn điều kiện

Kiểm tra $tong_so_chan \% 2 != 0$, không thỏa mãn điều kiện

Khởi tạo vòng lặp với biến cho chạy từ 0 đến 28.

Bước 18: Bắt đầu vòng lặp

Với biến cho chạy từ 0 đến 28, ta có: $ga = 28 - cho$

Kiểm tra điều kiện: $ga*2 + cho*4 == tong_so_chan$, thỏa mãn điều kiện.

Hàm trả về $ga = 22$, $cho = 6$

Bước 19: Ta có: Hàm cho_ga (28, 68) trả về $ga = 22$, $cho = 6$

Kiểm tra điều kiện $ga != -1$, thỏa mãn điều kiện

- ⇒ Hàm cho_ga (29, 72) trả về $ga = 22$, $cho = 7$
- ⇒ Hàm cho_ga (30, 76) trả về $ga = 22$, $cho = 8$
- ⇒ Hàm cho_ga (31, 80) trả về $ga = 22$, $cho = 9$
- ⇒ Hàm cho_ga (32, 84) trả về $ga = 22$, $cho = 10$
- ⇒ Hàm cho_ga (33, 88) trả về $ga = 22$, $cho = 11$
- ⇒ Hàm cho_ga (34, 92) trả về $ga = 22$, $cho = 12$
- ⇒ Hàm cho_ga (35, 96) trả về $ga = 22$, $cho = 13$
- ⇒ Hàm cho_ga (36, 100) trả về $ga = 22$, $cho = 14$

Bước 20:

In ra “Số gà là: 22”

“Số chó là: 14”

Nhung

Nhung

Nhung