

Bài 1:Hàm đệ quy tính tổng từ 1 đến n.

Bước 1: Gọi hàm `sum_of_number(7)`:

- ❖ $n=7$, không thỏa mãn điều kiện cơ sở.
- ❖ Thực hiện câu lệnh trong else: `return 7 + sum_of_number(6)`.

Bước 2: Gọi hàm `sum_of_number(6)`:

- ❖ $n=6$, không thỏa mãn điều kiện cơ sở.
- ❖ Thực hiện câu lệnh trong else: `return 6 + sum_of_number(5)`.

Bước 3: Gọi hàm `sum_of_number(5)`:

- ❖ $n=5$, không thỏa mãn điều kiện cơ sở.
- ❖ Thực hiện câu lệnh trong else: `return 5 + sum_of_number(4)`.

Bước 4: Gọi hàm `sum_of_number(4)`:

- ❖ $n=4$, không thỏa mãn điều kiện cơ sở.
- ❖ Thực hiện câu lệnh trong else: `return 4 + sum_of_number(3)`.

Bước 5: Gọi hàm `sum_of_number(3)`:

- ❖ $n=3$, không thỏa mãn điều kiện cơ sở.
- ❖ Thực hiện câu lệnh trong else: `return 3 + sum_of_number(2)`.

Bước 6: Gọi hàm `sum_of_number(2)`:

- ❖ $n=2$, không thỏa mãn điều kiện cơ sở.
- ❖ Thực hiện câu lệnh trong else: `return 2 + sum_of_number(1)`.

Bước 7: Gọi hàm `sum_of_number(1)`:

- ❖ $n=1$, thỏa mãn điều kiện cơ sở.
- ❖ Hàm trở về 1.

Các hàm được thực hiện từ dưới lên trên(từ bước 7 lên bước 1):

- ❖ `sum_of_number(2)` trả về: $2+1=3$.
- ❖ `sum_of_number(3)` trả về: $3+3=6$.
- ❖ `sum_of_number(4)` trả về: $6+4=10$.
- ❖ `sum_of_number(5)` trả về: $10+5=15$.
- ❖ `sum_of_number(6)` trả về: $15+6=21$.
- ❖ `sum_of_number(7)` trả về: $21+7=28$.

Vì vậy,tổng các số từ 1 đến 7 là:28.

Bài 2: Hàm tính số Fibonacci thứ n.

Bước 1: Gọi hàm fibonacci(8):

- ❖ $n=8$, không thỏa mãn điều kiện cơ sở.
- ❖ Thực hiện câu lệnh trong else: fibonacci(7) + fibonacci(6).

Bước 2: Gọi hàm fibonacci(7):

- ❖ $n=7$, không thỏa mãn điều kiện cơ sở.
- ❖ Thực hiện câu lệnh trong else: fibonacci(6) + fibonacci(5).

Bước 3: Gọi hàm fibonacci(6):

- ❖ $n=6$, không thỏa mãn điều kiện cơ sở.
- ❖ Thực hiện câu lệnh trong else: fibonacci(5) + fibonacci(4).

Bước 4: Gọi hàm fibonacci(5):

- ❖ $n=5$, không thỏa mãn điều kiện cơ sở.
- ❖ Thực hiện câu lệnh trong else: fibonacci(4) + fibonacci(3).

Bước 5: Gọi hàm fibonacci(4):

- ❖ $n=4$, không thỏa mãn điều kiện cơ sở.
- ❖ Thực hiện câu lệnh trong else: fibonacci(3) + fibonacci(2).

Bước 6: Gọi hàm fibonacci(3):

- ❖ $n=3$, không thỏa mãn điều kiện cơ sở.
- ❖ Thực hiện câu lệnh trong else: fibonacci(2) + fibonacci(1).

Bước 7: Gọi hàm fibonacci(2):

- ❖ $n=2$, không thỏa mãn điều kiện cơ sở.
- ❖ Thực hiện câu lệnh trong else: fibonacci(1) + fibonacci(0).

Bước 8: Gọi hàm fibonacci(1):

- ❖ $n=1$, thỏa mãn điều kiện cơ sở.
- ❖ Hàm trả về 1.

Bước 9: Gọi hàm fibonacci(0):

- ❖ $n=1$, thỏa mãn điều kiện cơ sở.
- ❖ Hàm trả về 0.

Các hàm được thực hiện từ dưới lên trên (từ bước 9 lên bước 1):

- ❖ fibonacci(2) trả về: $1+0=1$.
- ❖ fibonacci(3) trả về: $1+1=2$.
- ❖ fibonacci(4) trả về: $2+1=3$.

- ❖ fibonacci(5) trả về: $3+2=5$.
- ❖ fibonacci(6) trả về: $5+3=8$.
- ❖ fibonacci(7) trả về: $8+5=13$.
- ❖ fibonacci(8) trả về: $13+8=21$.

Vì vậy, số Fibonacci thứ 8 là 21.

Bài 3: Hàm đệ quy tính x mũ n.

Bước 1: Gọi hàm power(2,6):

- ❖ $n=6$, không thỏa mãn điều kiện cơ sở.
- ❖ Thực hiện câu lệnh trong else: $2 * \text{power}(2, 5)$.

Bước 2: Gọi hàm power(2,5):

- ❖ $n=5$, không thỏa mãn điều kiện cơ sở.
- ❖ Thực hiện câu lệnh trong else: $2 * \text{power}(2, 4)$.

Bước 3: Gọi hàm power(2,4):

- ❖ $n=4$, không thỏa mãn điều kiện cơ sở.
- ❖ Thực hiện câu lệnh trong else: $2 * \text{power}(2, 3)$.

Bước 4: Gọi hàm power(2,3):

- ❖ $n=3$, không thỏa mãn điều kiện cơ sở.
- ❖ Thực hiện câu lệnh trong else: $2 * \text{power}(2, 2)$.

Bước 5: Gọi hàm power(2,2):

- ❖ $n=2$, không thỏa mãn điều kiện cơ sở.
- ❖ Thực hiện câu lệnh trong else: $2 * \text{power}(2, 1)$.

Bước 6: Gọi hàm power(2,1):

- ❖ $n=1$, không thỏa mãn điều kiện cơ sở.
- ❖ Thực hiện câu lệnh trong else: $2 * \text{power}(2, 0)$.

Bước 7: Gọi hàm power(2,0):

- ❖ $n=0$, thỏa mãn điều kiện cơ sở.
- ❖ Hàm trở về 1.

Các hàm được thực hiện từ dưới lên trên (từ bước 7 lên bước 1):

- ❖ power(2,1) trả về: $2*1=2$.
- ❖ power(2,2) trả về: $2*2=4$.
- ❖ power(2,3) trả về: $2*4=8$.
- ❖ power(2,4) trả về: $2*8=16$.

❖ $\text{power}(2,5)$ trả về: $2 \cdot 16 = 32$.

❖ $\text{power}(2,6)$ trả về: $2 \cdot 32 = 64$.

Vì vậy, 2 mũ 6 bằng 64.

Bài 4: Hàm đệ quy giải bài toán Tháp Hà Nội.

Gọi hàm `thap_ha_noi(4,"A","C","B")`:

- ❖ Gọi hàm `thap_ha_noi(3,"A","B","C")`:
 - Gọi hàm `thap_ha_noi(2,"A","C","B")`:
 - Gọi hàm `thap_ha_noi(1,"A","B","C")`: In ra: "Chuyển đĩa 1 từ cột A sang cột C".
 - In ra: "Chuyển đĩa 2 từ cột A sang cột B".
 - Gọi hàm `thap_ha_noi(1,"C","B","A")`: In ra: "Chuyển đĩa 1 từ cột C sang cột A".
 - In ra "Chuyển đĩa 3 từ cột A sang cột C".
 - Gọi hàm `thap_ha_noi(2,"B","C","A")`:
 - Gọi hàm `thap_ha_noi(1,"B","A","C")`: In ra: "Chuyển đĩa 1 từ cột B sang cột C".
 - In ra: "Chuyển đĩa 2 từ cột B sang cột A".
 - Gọi hàm `thap_ha_noi(1,"C","B","A")`: In ra: "Chuyển đĩa 1 từ cột C sang cột B".
- ❖ In ra "Chuyển đĩa 4 từ cột A sang cột B".
- ❖ Gọi hàm `thap_ha_noi(3,"C","B","A")`:
 - Gọi hàm `thap_ha_noi(2,"C","A","B")`:
 - Gọi hàm `thap_ha_noi(1,"C","B","A")`: In ra: "Chuyển đĩa 1 từ cột C sang cột A".
 - In ra: "Chuyển đĩa 2 từ cột C sang cột B".
 - Gọi hàm `thap_ha_noi(1,"A","B","C")`: In ra: "Chuyển đĩa 1 từ cột A sang cột C".
 - In ra: "Chuyển đĩa 3 từ cột C sang cột A".
 - Gọi hàm `thap_ha_noi(2,"B","A","C")`:
 - Gọi hàm `thap_ha_noi(1,"B","C","A")`: In ra: "Chuyển đĩa 1 từ cột B sang cột A".
 - In ra: "Chuyển đĩa 2 từ cột B sang cột C".
 - Gọi hàm `thap_ha_noi(1,"A","C","B")`: In ra: "Chuyển đĩa 1 từ cột A sang cột B".

Các hàm được thực hiện từ trên xuống dưới và kết quả của hàm này như sau:

- ❖ Chuyển đĩa 1 từ cột A sang cột C.

- ❖ Chuyển đĩa 2 từ cọc A sang cọc B.
- ❖ Chuyển đĩa 1 từ cọc C sang cọc A.
- ❖ Chuyển đĩa 3 từ cọc A sang cọc C.
- ❖ Chuyển đĩa 1 từ cọc B sang cọc C.
- ❖ Chuyển đĩa 2 từ cọc B sang cọc A.
- ❖ Chuyển đĩa 1 từ cọc C sang cọc B.
- ❖ Chuyển đĩa 4 từ cọc A sang cọc B.
- ❖ Chuyển đĩa 1 từ cọc C sang cọc A.
- ❖ Chuyển đĩa 2 từ cọc C sang cọc B.
- ❖ Chuyển đĩa 1 từ cọc A sang cọc C.
- ❖ Chuyển đĩa 3 từ cọc C sang cọc A.
- ❖ Chuyển đĩa 1 từ cọc B sang cọc A.
- ❖ Chuyển đĩa 2 từ cọc B sang cọc C.
- ❖ Chuyển đĩa 1 từ cọc A sang cọc B.

Bài 5: Hàm đệ quy giải bài toán cổ vừa gà vừa chó:

Gọi hàm cho_ga(36,100):

- ❖ tong_so_con=36 và tong_so_chan=100, không thỏa mãn điều kiện cơ sở ở dòng 2.
- ❖ Thực hiện lệnh tiếp theo tong_so_chan % 2 = 0, không thỏa mãn điều kiện cơ sở ở dòng 4 .
- ❖ Thực hiện vòng lặp for ở dòng 6, kiểm tra từ 0 đến 36:
 - Khởi tạo biến ga = 100 – cho.
 - Thực hiện câu lệnh if ở dòng 8: $ga * 2 + cho * 4 = 100$ nếu đúng thì trả về cho, ga nếu sai thì thoát khỏi vòng lặp.
- ❖ Gọi hàm đệ quy cho_ga(35,96):
 - Nếu ga = 1 thì trả về giá trị -1,-1.
 - Ngược lại trả về cho+1, ga.

Ví dụ: chó = 4, gà = 25 => $4 * 4 + 25 * 2 = 66 \neq 100$ (loại).

Hàm đệ quy: gà giảm bớt 1 là 24, chó tăng lên 1 là 5 => $5 * 4 + 24 * 2 = 68 \neq 100$ (loại).

Cứ thực hiện như vậy đến khi tìm được kết quả đúng là chó=14,gà=22 thì in ra “Số gà là: 22” “Số chó là: 14”.

Vậy số chó tìm được là 14 , số gà tìm được là 22.

Nguyễn Đình Trọng