

Bài 1:

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: $n=1$, điều kiện cơ sở được thỏa mãn

Hàm trả về 1

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

`sum_of_numbers(1)` trả về 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 $4 + 6 = 10$ ->

`sum_of_numbers(5)` trả về $5 + 10 = 15$ -> `sum_of_numbers(6)` trả về $6 + 15 = 21$ ->

`sum_of_numbers(7)` trả về $7 + 21 = 28$. Vì vậy, kết quả cuối cùng là 28.

Bài 2:

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

Bước 9: Hàm fibonacci(0) được gọi

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

Các hàm được thực hiện lần lượt theo thứ tự từ dưới lên trên (từ bước 9 lên bước 1)

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

fibonacci(3) trả về $1+1=2 \rightarrow$ fibonacci(4) trả về $2+1=3 \rightarrow$ fibonacci(5) trả về $3+2=5 \rightarrow$

fibonacci(6) trả về $5+3=8$ → fibonacci(7) trả về $8+5=13$ → fibonacci(8) trả về $13+8=21$.

Vì vậy, kết quả cuối cùng là 21, đây là số Fibonacci thứ 8.

Bài 3:

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: `return 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 cơ sở

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

Bước 6: Hàm `power(2, 1)` được gọi

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

Thực hiện câu lệnh trong else: `return 2 * power(2, 0)`

Bước 7: Hàm `power(2, 0)` được gọi

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

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

`power(2, 0)` trả về 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 $2*8=16$ → `power(2, 5)` trả về $2*16=32$ → `power(2, 6)` trả về $2*32=64$. Vì vậy, kết quả cuối cùng là 64.

Bài 4:

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

Bắt đầu với 4 đĩa ở cột A, chuyển chúng sang cột B, sử dụng cột C làm cột trung gian.

Bước 2: Gọi hàm `thap_ha_noi(3, "A", "B", "C")`

Chuyển 3 đĩa từ cột A sang cột C, sử dụng cột B làm cột trung gian

Bước 3: Gọi hàm `thap_ha_noi(2, "A", "C", "B")`

Chuyển 2 đĩa từ cột A sang cột B, sử dụng cột C làm trung gian

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

Có 1 đĩa ở cột A, di chuyển nó sang cột B

Hàm trả về dòng in "Chuyển đĩa 1 từ cột A sang cột B"

Bước 5: Trở lại hàm trước đó `thap_ha_noi(2, "A", "C", "B")`

Đã chuyển xong 1 đĩa từ cột A sang cột B

In ra "Chuyển đĩa 2 từ cột A sang cột B"

Bây giờ cột B chứa đĩa 2, cột C là cột trung gian, và cột A là cột đích

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

Có 1 đĩa ở cột C, di chuyển nó sang cột B

Hàm trả về dòng in "Chuyển đĩa 1 từ cột C sang cột B"

Bước 7: Trở lại hàm gọi trước đó `thap_ha_noi(2, "A", "C", "B")`

Đã chuyển xong 1 đĩa từ cột C sang cột B

Quay lại hàm gọi trước đó để tiếp tục chuyển các đĩa còn lại

Bước 8: In ra "Chuyển đĩa 3 từ cột A sang cột B"

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

Có 1 đĩa ở cột A, di chuyển nó sang cột B

Hàm trả về dòng in "Chuyển đĩa 1 từ cột A sang cột B"

Bước 10: Trở lại hàm gọi trước đó `thap_ha_noi(3, "A", "B", "C")`

Đã chuyển xong 1 đĩa từ cột A sang cột B

In ra "Chuyển đĩa 4 từ cột A sang cột B"

Bây giờ toàn bộ 4 đĩa đã được chuyển từ cột A sang cột B

Vì vậy, tất cả các đĩa đã được chuyển từ cột A sang cột B

Bài 5:

Bước 1: Kiểm tra điều kiện cơ bản

Nếu cả `tong_so_con` và `tong_so_chan` đều bằng 0, có nghĩa là không có con thú nào, hàm trả về (0, 0).

Nếu `tong_so_chan` không chia hết cho 2, có nghĩa là số chân không thể tạo thành số lượng con thú nguyên tắc, hàm trả về (-1, -1).

Bước 2: kiểm tra từng trường hợp có thể

Trong bước này, chúng ta sử dụng một vòng lặp `for` để thử tất cả các trường hợp có thể của số lượng chó và gà. Mỗi lần lặp, chúng ta tính số chó và số gà dựa trên số chân và số con thú.

Nếu tổng số chân và tổng số con thú phù hợp với một trường hợp cụ thể (số chân của gà và chó đúng như số chân đã cho), hàm trả về số lượng chó và gà tương ứng.

Bước 3:

Nếu không có trường hợp nào phù hợp trong vòng lặp `for`, hàm gọi đệ quy với số con thú giảm đi 1 và số chân giảm đi 4 (do mỗi con thú bao gồm 4 chân).

Nếu kết quả của đệ quy không phải là một trường hợp hợp lệ (-1, -1), ta cộng thêm 1 cho số con thú và trả về kết quả.

Nếu kết quả của đệ quy vẫn là một trường hợp không hợp lệ, hàm trả về (-1, -1).

Nguyễn Tiến Dũng