

Bài 1 :

Bước 1: Gọi hàm `sum_of_numbers(7)`

Kiểm tra `n == 1`: không thỏa mãn (vì $7 \neq 1$), thực hiện lệnh trong `else`: `return 7 + sum_of_numbers(6)`

Bước 2: Hàm `sum_of_numbers(6)` được gọi

Kiểm tra `n == 1`: không thỏa mãn (vì $6 \neq 1$), thực hiện lệnh trong `else`: `return 6 + sum_of_numbers(5)`

Bước 3: Hàm `sum_of_numbers(5)` được gọi

Kiểm tra `n == 1`: không thỏa mãn (vì $5 \neq 1$), thực hiện lệnh trong `else`: `return 5 + sum_of_numbers(4)`

Bước 4: Hàm `sum_of_numbers(4)` được gọi

Kiểm tra `n == 1`: không thỏa mãn (vì $4 \neq 1$), thực hiện lệnh trong `else`: `return 4 + sum_of_numbers(3)`

Bước 5: Hàm `sum_of_numbers(3)` được gọi

Kiểm tra `n == 1`: không thỏa mãn (vì $3 \neq 1$), thực hiện lệnh trong `else`: `return 3 + sum_of_numbers(2)`

Bước 6: Hàm `sum_of_numbers(2)` được gọi

Kiểm tra `n == 1`: không thỏa mãn (vì $2 \neq 1$), thực hiện lệnh trong `else`: `return 2 + sum_of_numbers(1)`

Bước 7: Hàm `sum_of_numbers(1)` được gọi

Lần này, `n == 1` thỏa mãn. Hàm trả về giá trị 1

Từ các bước trên, chúng ta sẽ có một dãy lời gọi như sau và giải quyết từ dưới lên:

`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 $7 + 21 = 28$

Bài 3:

Bước 1: Gọi hàm `power(2, 6)`

Kiểm tra điều kiện cơ sở: `n = 6`, không thỏa mãn điều kiện cơ sở (`n` không bằng 0)

Thực hiện câu lệnh trong `else`: trả về $2 * \text{power}(2, 5)$

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 trong else: trả về $2 * \text{power}(2, 4)$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$\text{power}(2, 0)$ trả về 1

$\text{power}(2, 1)$ trả về $2 * 1 = 2$

$\text{power}(2, 2)$ trả về $2 * 2 = 4$

$\text{power}(2, 3)$ trả về $2 * 4 = 8$

$\text{power}(2, 4)$ trả về $2 * 8 = 16$

$\text{power}(2, 5)$ trả về $2 * 16 = 32$

$\text{power}(2, 6)$ trả về $2 * 32 = 64$

Vì vậy, kết quả cuối cùng là 64, đây là lũy thừa của 2 mũ 6

Bài 2 :

Bước 1: Gọi hàm $\text{fibonacci}(8)$

Kiểm tra điều kiện cơ sở: $n = 8$, không thỏa mãn điều kiện cơ sở ($n \leq 1$)

Thực hiện câu lệnh trong else: $\text{return fibonacci}(7) + \text{fibonacci}(6)$

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: $\text{return fibonacci}(6) + \text{fibonacci}(5)$

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: $\text{return fibonacci}(5) + \text{fibonacci}(4)$

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: $\text{return fibonacci}(4) + \text{fibonacci}(3)$

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: $\text{return fibonacci}(3) + \text{fibonacci}(2)$

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: $\text{return fibonacci}(2) + \text{fibonacci}(1)$

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: $\text{return fibonacci}(1) + \text{fibonacci}(0)$

Bước 8: Hàm fibonacci(1) và hàm fibonacci(0) được gọi

Đối với fibonacci(1), 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

Đối với fibonacci(0), kiểm tra điều kiện cơ sở: $n = 0$, thỏa mãn điều kiện cơ sở, hàm trả về 0

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

Hàm fibonacci(0) trả về 0

Hàm fibonacci(1) trả về 1

Hàm fibonacci(2) trả về $0+1 = 1$

Hàm fibonacci(3) trả về $1+1 = 2$

Hàm fibonacci(4) trả về $2+1 = 3$

Hàm fibonacci(5) trả về $2+3 = 5$

Hàm fibonacci(6) trả về $3+5 = 8$

Hàm fibonacci(7) trả về $5 + 8 = 13$

Hàm fibonacci(8) trả về $8 + 13 = 21$

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

Bài 4 :

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

Ta kiểm tra điều kiện cơ sở: $n = 4$. Vì không thỏa mãn điều kiện cơ sở, ta thực hiện câu lệnh trong phần `else`: Gọi hàm `thap_ha_noi(3, "A", "B", "C")`

In ra : “chuyển đĩa 4 từ cột A sang cột B”

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

Kiểm tra điều kiện cơ sở: $n = 3$. Vì không thỏa mãn điều kiện cơ sở, ta thực hiện câu lệnh trong phần `else`: Gọi hàm `thap_ha_noi(2, "A", "C", "B")`

In ra : “chuyển đĩa 3 từ cột A sang cột B”

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

Kiểm tra điều kiện cơ sở: $n = 2$. Vì không thỏa mãn điều kiện cơ sở, ta thực hiện câu lệnh trong phần `else`: Gọi hàm `thap_ha_noi(1, "A", "B", "C")`

In ra thông báo: “chuyển đĩa 2 từ cột A sang cột B”

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

Kiểm tra điều kiện cơ sở: $n = 1$. Điều kiện cơ sở được thỏa mãn, in ra thông báo: “Di chuyển đĩa 1 từ cột A sang cột B”.

Kết quả cuối cùng: Tổng hợp các thông báo in ra, ta có kết quả chuyển 4 đĩa từ cọc A sang cọc B, với trung gian là cọc C :

Chuyển đĩa 1 từ cột A sang cột B

Chuyển đĩa 2 từ cột A sang cột C

Chuyển đĩa 1 từ cột B sang cột A

Chuyển đĩa 3 từ cột A sang cột B

Chuyển đĩa 1 từ cột C sang cột B

Chuyển đĩa 2 từ cột C sang cột A

Chuyển đĩa 1 từ cột B sang cột C

Chuyển đĩa 4 từ cột A sang cột C

Chuyển đĩa 1 từ cột B sang cột A

Chuyển đĩa 2 từ cột B sang cột C

Chuyển đĩa 1 từ cột A sang cột B

Chuyển đĩa 3 từ cột B sang cột A

Chuyển đĩa 1 từ cột C sang cột A

Chuyển đĩa 2 từ cột C sang cột B

Chuyển đĩa 1 từ cột A sang cột C

Bài 5 :

Bước 1:

Gọi hàm cho_ga(36, 100).

Kiểm tra điều kiện cơ bản, không thỏa mãn vì tổng số con 36 và tổng số chân 100 không bằng 0.

Vòng lặp for chạy từ 0 đến 36 để tìm số chó và số gà. Trong trường hợp này, cả hai được tìm thấy nhanh chóng nên không cần dùng đến đệ quy.

Bước 2:

Tìm số chó (cho) và số gà (ga) bên trong vòng lặp.

Mỗi lần lặp, giả sử cho bằng giá trị của vòng lặp hiện tại, và ga sẽ là $36 - \text{cho}$.

Nếu $2 * \text{ga} + 4 * \text{cho} = 100$ (điều kiện số chân chó và gà), trả về kết quả.

Trong trường hợp không tìm được tổ hợp nào phù hợp, chương trình sẽ tiến hành gọi hàm đệ quy.

Bước 3 (đệ quy được áp dụng nếu không tìm thấy kết quả trong vòng lặp):

Gọi đệ quy cho_ga(tong_so_con - 1, tong_so_chan - 4).

Tiếp tục kiểm tra điều kiện cơ bản với giá trị mới sau mỗi lần đệ quy, đồng thời giải quyết bài toán theo cách tìm ở vòng lặp for.

Nếu giải quyết không thành công (-1, -1), tiếp tục gọi đệ quy với số con giảm dần và số chân giảm đi 4 mỗi lần, đại diện cho việc loại bỏ một chó giả định và thử thêm một chó vào tổng số con.

Lặp lại bước 3 cho đến khi tìm được tổ hợp phù hợp hoặc cho đến khi số lượng con giảm xuống còn 0.

Bước 4 (trả về kết quả):

Khi tìm thấy tổ hợp số chó và số gà hợp lệ, trả về kết quả và kết thúc quá trình đệ quy.

Bước 5 (kết thúc):

Nếu sau một chuỗi các lần gọi đệ quy không thể tìm thấy tổ hợp nào hợp lệ, hàm cuối cùng trả về -1, -1 để biểu thị không có giải pháp cho vấn đề, và quá trình tìm kiếm kết thúc.

Quay trở lại lời gọi hàm ban đầu cho_ga(36, 100) với kịch bản giả định rằng vòng lặp không thể tìm thấy tổ hợp phù hợp (mặc dù trong trường hợp thực tế có thể tìm thấy ngay):

Bước này gọi đệ quy với cho_ga(35, 96), và tiếp tục cho đến khi tìm ra giải pháp hoặc không còn khả năng nào nữa.

Sau khi hàm đệ quy tìm ra kết quả, giá trị của số gà và số chó sẽ được in ra:

Vũ Gia Huy _ DHKL17A2HN_23174600110 _ Ca sáng

Số gà là: 22

Số chó là: 14

Vũ Gia Huy