

Bài 1 :

Bước 1: Gọi hàm `sum_of_number(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_number(7-1)`.

Bước 2: Hàm `sum_of_number (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_number(6-1)`

Bước 3: Hàm `sum_of_number (1)` 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_number(5-1)`.

==> Làm tương tự các bước trên cho đến khi $n = 1$

Bước 4: Hàm `sum_of_number (1)` được gọi.

Kiểm tra điều kiện cơ sở: $n = 0$, 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

- Giá trị 1 được trả về cho hàm `sum_of_numbers(2)`.
- Giá trị $1 + \text{sum_of_numbers}(2)$ được trả về cho hàm `sum_of_numbers(3)`.
- Giá trị $1 + \text{sum_of_numbers}(2) + \text{sum_of_numbers}(3)$ được trả về cho hàm `sum_of_numbers(4)`.
- Giá trị $1 + \text{sum_of_numbers}(2) + \text{sum_of_numbers}(3) + \text{sum_of_numbers}(4)$ được trả về cho hàm `sum_of_numbers(5)`.
- Giá trị $1 + \text{sum_of_numbers}(2) + \text{sum_of_numbers}(3) + \text{sum_of_numbers}(4) + \text{sum_of_numbers}(5)$ được trả về cho hàm `sum_of_numbers(6)`.
- Giá trị $1 + \text{sum_of_numbers}(2) + \text{sum_of_numbers}(3) + \text{sum_of_numbers}(4) + \text{sum_of_numbers}(5) + \text{sum_of_numbers}(6)$ được trả về cho hàm `sum_of_numbers(7)`.
- Hàm `sum_of_numbers(7)` trả về giá trị cuối cùng là 28

Bài 2 :

Quy trình các bước như sau:

Bước 1: Gọi hàm `Fibonacci(8)`.

Kiểm tra điều kiện cơ sở: $8 = 3$, không thỏa mãn điều kiện cơ sở.

Thực hiện câu lệnh trong else: `return Fibonacci(8-1) + Fibonacci(8-2)`

Bước 2: Hàm Fibonacci(7) được gọi.

Kiểm tra điều kiện cơ sở: $7 = 2$, không thỏa mãn điều kiện cơ sở.

Thực hiện câu lệnh trong else: $\text{Fibonacci}(7-1) + \text{Fibonacci}(7-2)$.

Bước 3: Hàm Fibonacci (6) được gọi.

Kiểm tra điều kiện cơ sở: $6 = 1$, không thỏa mãn điều kiện cơ sở.

Thực hiện câu lệnh trong else: $\text{Fibonacci}(6-1) + \text{Fibonacci}(6-2)$.

(Làm tương tự cho đến khi $n \leq 1$)

Bước 4: Hàm factorial(1) được gọi.

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

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

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: $\text{return } 2 * \text{power}(2,6-1)$

Ta được $n = 5$

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: $\text{return } 2 * \text{power}(2,5-1)$

Ta được $n = 4$

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: $\text{return } 2 * \text{power}(2,4-1)$

Ta được $n = 3$

Bước 3: 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: $\text{return } 2 * \text{power}(2,3-1)$

Ta được $n = 2$

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: $\text{return } 2 * \text{power}(2,2-1)$

Ta được $n = 1$

Bước 6: Hàm power (2,1) được gọi.

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

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

Ta được $n = 0$

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 1.

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

Khi đó $2 * \text{power}(2, 0) : 2 * 2^0$ Ta được kết quả 2

Tiếp tục $2 * 2 * 1$ Ta được kết quả 4

Tương tự $2 * \text{power}(2, 6)$ Ta được 64 (Tức là : 2 nhân với kết quả trước đó)

Bài 4 :

- Gọi hàm $\text{thap_ha_noi}(4, "A", "C", "B")$.
- Hàm $\text{thap_ha_noi}(4, "A", "C", "B")$ sẽ gọi đệ quy hàm $\text{thap_ha_noi}(3, "A", "B", "C")$.
- Hàm $\text{thap_ha_noi}(3, "A", "B", "C")$ sẽ gọi đệ quy hàm $\text{thap_ha_noi}(2, "A", "C", "B")$.
- Hàm $\text{thap_ha_noi}(2, "A", "C", "B")$ sẽ gọi đệ quy hàm $\text{thap_ha_noi}(1, "A", "B", "C")$.
- Hàm $\text{thap_ha_noi}(1, "A", "B", "C")$ sẽ in ra hướng dẫn di chuyển đĩa 1 từ cột A sang cột B.
- Hàm $\text{thap_ha_noi}(2, "A", "C", "B")$ sẽ in ra hướng dẫn di chuyển đĩa 2 từ cột A sang cột B.
- Hàm $\text{thap_ha_noi}(3, "A", "B", "C")$ sẽ in ra hướng dẫn di chuyển đĩa 3 từ cột A sang cột B.
- Hàm $\text{thap_ha_noi}(4, "A", "C", "B")$ sẽ in ra hướng dẫn di chuyển đĩa 4 từ cột A sang cột B.
- Hàm $\text{thap_ha_noi}(3, "C", "A", "B")$ sẽ gọi đệ quy hàm $\text{thap_ha_noi}(2, "C", "B", "A")$.
- Hàm $\text{thap_ha_noi}(2, "C", "B", "A")$ sẽ gọi đệ quy hàm $\text{thap_ha_noi}(1, "C", "A", "B")$.
- Hàm $\text{thap_ha_noi}(1, "C", "A", "B")$ sẽ in ra hướng dẫn di chuyển đĩa 1 từ cột C sang cột B.

- Hàm `thap_ha_noi(2, "C", "B", "A")` sẽ in ra hướng dẫn di chuyển đĩa 2 từ cột `C` sang cột `B`.
- Hàm `thap_ha_noi(3, "C", "A", "B")` sẽ in ra hướng dẫn di chuyển đĩa 3 từ cột `C` sang cột `B`.

Bài 5 :

Bước 1 : Gọi hàm `def cho_ga(tong_so_con , tong_so_chan)`

Bước 2 (dòng 2) : Kiểm tra ĐK cơ bản (Nếu `tong_so_con` và `tong_so_chan` đều bằng 0, thì nghĩa là không có gà và chó nào. Lúc này, hàm trả về kết quả là `(0, 0)`)

Bước 3 (dòng 4) : Kiểm tra tính hợp lệ (Tổng số chân của gà và chó phải là số chẵn. Nếu `tong_so_chan` là số lẻ, thì nghĩa là không có cách nào để chia số chân này thành 4 phần (cho gà) và 2 phần (cho chó). Lúc này, hàm trả về kết quả là `(0, 0)`)

Bước 4 (