**Bài 1:**

def sum\_of\_numbers(n):

    if n ==1:

        return 1

    else:

        return n +sum\_of\_numbers(n-1)

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 cơ sở.

- Thực hiện dòng lệnh trong else: return 7 + sum\_of\_numbers(6).

**Bước 2:** Gọi hàm sum\_of\_numbers(6).

- Kiểm tra điều kiện cơ sở n = 6, không thỏa mãn điều kiện cơ sở.

- Thực hiện dòng lệnh trong else: return 6 + sum\_of\_numbers(5).

**Bước 3:** Gọi hàm sum\_of\_numbers(5).

- Kiểm tra điều kiện cơ sở n = 5, không thỏa mãn điều kiện cơ sở.

- Thực hiện dòng lệnh trong else: return 5 + sum\_of\_numbers(4).

**Bước 4:** Gọi hàm sum\_of\_numbers(4).

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

- Thực hiện dòng lệnh trong else: return 4 + sum\_of\_numbers(3).

**Bước 5:** Gọi hàm sum\_of\_numbers(3).

- Kiểm tra điều kiện cơ sở n = 3, không thỏa mãn điều kiện cơ sở.

- Thực hiện dòng lệnh trong else: return 3 + sum\_of\_numbers(2).

**Bước 6:** Gọi hàm sum\_of\_numbers(2).

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

- Thực hiện dòng lệnh trong else: return 2 + sum\_of\_numbers(1).

**Bước 7:** Gọi hàm sum\_of\_numbers(1).

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

- Thực hiện dòng lệnh if: return 1.

**Các bước được thực hiện theo thứ tự từ dưới lên trên( từ bước 6 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: 28.

**Bài 2:**

def fibonacci(n):

    if n <==1:

        return n

    else:

        return fibonacci(n-1) + fibonacci(n-2)

print(fibonacci(8))

***Quy trình đọc các bước:***

**Bước 1:** Gọi hàm Fibonacci(8).

- Kiểm tra điều kiện n = 8, không thỏa mãn.

- Thực hiện dòng lệnh trong else: return Fibonacci(7) + Fibonacci (6).

**Bước 2:** Gọi hàm Fibonacci(7).

- Kiểm tra điều kiện n = 7, không thỏa mãn.

- Thực hiện dòng lệnh trong else: return Fibonacci(6) + Fibonacci (5).

**Bước 3:** Gọi hàm Fibonacci(6).

- Kiểm tra điều kiện n = 6, không thỏa mãn.

- Thực hiện dòng lệnh trong else: return Fibonacci(5) + Fibonacci (4).

**Bước 4:** Gọi hàm Fibonacci(5).

- Kiểm tra điều kiện n = 5, không thỏa mãn.

- Thực hiện dòng lệnh trong else: return Fibonacci(4) + Fibonacci (3).

**Bước 5:** Gọi hàm Fibonacci(4).

- Kiểm tra điều kiện n = 4, không thỏa mãn.

- Thực hiện dòng lệnh trong else: return Fibonacci(3) + Fibonacci (2).

**Bước 6:** Gọi hàm Fibonacci(3).

- Kiểm tra điều kiện n = 3, không thỏa mãn.

- Thực hiện dòng lệnh trong else: return Fibonacci(2) + Fibonacci (1).

**Bước 7:** Gọi hàm Fibonacci(2).

- Kiểm tra điều kiện n = 2, không thỏa mãn.

- Thực hiện dòng lệnh trong else: return Fibonacci(1) + Fibonacci (0).

**Bước 8:** Gọi hàm Fibonacci(1).

- Kiểm tra điều kiện n = 1, thỏa mãn.

- Thực hiện dòng lệnh trong if: return 1.

**Bước 9:** Gọi hàm Fibonacci(0).

- Kiểm tra điều kiện n = 0, thỏa mãn.

- Thực hiện dòng lệnh trong if: return 0.

**Quay lại từng bước trước đó để thực hiện các phép tính:**

- Fibonacci(2): 1 + 0 = 1.

- Fibonacci(3): 1 + 1 = 2.

- Fibonacci(4): 2 + 1 = 3.

- Fibonacci(5): 3 + 2 = 5.

- Fibonacci(6): 5 + 3 = 8.

- Fibonacci(7): 8 + 5 = 13.

- Fibonacci(8): 13 + 8 = 21.

Vậy kết quả in ra là: 21.

**Bài 3:**

def power( x ,n):

    if n == 0:

        return 1

    else:

        return x \* power(x, n-1)

print(power(2,6))

***Quy trình đọc các bước như sau:***

**Bước 1:** Gọi hàm power(2,6).

- Kiểm tra điều kiện n = 6, điều kiện không thỏa mãn.

- Tiếp tục lệnh else: return 2\* power(2,5).

**Bước 2:** Gọi hàm power(2,5).

- Kiểm tra điều kiện n = 5 điều kiện không thỏa mãn.

- Tiếp tục lệnh else: return 2\* power(2,4).

**Bước 3:** Gọi hàm power(2,4).

- Kiểm tra điều kiện n = 4, điều kiện không thỏa mãn.

- Tiếp tục lệnh else: return 2\* power(2,3).

**Bước 4:** Gọi hàm power(2,3).

- Kiểm tra điều kiện n = 3, điều kiện không thỏa mãn.

- Tiếp tục lệnh else: return 2\* power(2,2).

**Bước 5:** Gọi hàm power(2,2).

- Kiểm tra điều kiện n = 2, điều kiện không thỏa mãn.

- Tiếp tục lệnh else: return 2\* power(2,1).

**Bước 6:** Gọi hàm power(2,1).

- Kiểm tra điều kiện n = 1, điều kiện không thỏa mãn.

- Tiếp tục lệnh else: return 2\* power(2,0).

**Bước 7:** Gọi hàm power(2,0).

- Kiểm tra điều kiện n = 0, điều kiện thỏa mãn.

- Sử dụng lệnh if : return 1.

**Quay lại từng bước thực hiện các phép nhân.**

- Hàm power(2,1): 2\*1 = 2.

- Hàm power(2,2): 2\*2 = 4.

- Hàm power(2,3): 2\*4 = 8.

- Hàm power(2,4): 2\*8= 16.

- Hàm power(2,5): 2\*16= 32.

- Hàm power(2,6): 2\*32= 64.

Vậy kết quả in ra được là: 64.

**Bài 4:**

def thap\_ha\_noi(n,A,B,C):

    if n ==1:

        print(f"Chuyển đĩa 1 cột từ cột {A} sang cột {B}")

    else:

        thap\_ha\_noi(n-1,A,B,C)

        print(f"Chuyển đĩa {n} từ cột {A} sang cột {B}")

        thap\_ha\_noi(n-1,C,A,B)

thap\_ha\_noi(4, "A","C","B")

Các bước thực hiện của hàm đệ quy khi chuyển 4 đĩa từ cọc A sang B với trung gian C, tức là thap\_ha\_noi(4, “A”, “C”, “B”).

**Bước 1:** Gọi thap\_ha\_noi(4, “A”, “C”, “B”).

- Kiểm tra điều kiện n =4, không thỏa mãn.

- Thực hiện dòng lệnh else: thap\_ha\_noi(3, “A”, “B”, “C”).

**Bước 2:** Gọi thap\_ha\_noi(3, “A”, “B”, “C”).

- Kiểm tra điều kiện n =3, không thỏa mãn.

- Thực hiện dòng lệnh else: thap\_ha\_noi(2, “A”, “C”, “B”).

**Bước 3:** Gọi thap\_ha\_noi(2, “A”, “C”, “B”).

- Kiểm tra điều kiện n =2, không thỏa mãn.

- Thực hiện dòng lệnh else: thap\_ha\_noi(1, “A”, “B”, “C”).

**Bước 4:** Gọi thap\_ha\_noi(1, “A”, “B”, “C”).

- Kiểm tra điều kiện n =1, thỏa mãn.

- Thực hiện lệnh if: print(f”Chuyển đĩa từ cột {A} sang cột {B}”), in ra “Chuyển đĩa 1 từ cột A sang cột B”.

**Quay lại bước 3:** thap\_ha\_noi(2, “A”, “C”, “B”).

- Thực hiện in ra: “Chuyển đĩa 2 từ cột {A} sang cột {C}”.

- Thực hiện thap\_ha\_noi(1, “B”, “C”, “A”).

**Bước 5:** Gọi thap\_ha\_noi(1, “B”, “C”, “A”).

- Kiểm tra điều kiện n = 1, thỏa mãn.

- Thực hiện lệnh if: in ra “Chuyển đĩa 1 từ cột B sang cột C”.

**Quay lại bước 2:** thap\_ha\_noi(3, “A”, “B”, “C”).

- Thực hiện in ra “Chuyển đĩa 3 từ cột A sang cột B”.

- Thực hiện thap\_ha\_noi(2, “C”, “B”, “A”).

**Bước 6:** thap\_ha\_noi(2, “C”, “B”, “A”).

- Kiểm tra điều kiện n = 2, không thỏa mãn.

- Thực hiện lệnh else: thap\_ha\_noi(1, “C”, “A”, “B”).

**Bước 7**: thap\_ha\_noi(1, “C”, “A”, “B”).

- Kiểm tra điều kiện n =1, thỏa mãn.

- Thực hiện lệnh if : in ra “Chuyển đĩa 1 từ cột {C} sang cột {A}”.

**Quay lại bước 6:** thap\_ha\_noi(2, “C”, “B”, “A”).

- Thực hiện in ra: “Chuyển đĩa 2 từ cột C sang cột B”.

- Thực hiện thap\_ha\_noi(1, “A”, “B”, “C”).

**Bước 8:** thap\_ha\_noi(1, “A”, “B”, “C”)

- Kiểm tra điều kiện n =1, thỏa mãn.

- Thực hiện lệnh if: in ra “Chuyển đĩa 1 từ cột A sang cột B”.

**Quay lại bước 1**: thap\_ha\_noi(4, “A”, “C”, “B”).

- Thực hiện thap\_ha\_noi(3, “B”, “C”, “A”).

**Bước 9:** thap\_ha\_noi(3, “B”, “C”, “A”).

- Kiểm tra điều kiện n = 1, không thỏa mãn.

- Thực hiện lệnh else: thap\_ha\_noi(2, “B”, “A”, “C”).

**Bước 10:** thap\_ha\_noi(2, “B”, “A”, “C”).

- Kiểm tra điều kiện n = 2, không thỏa mãn.

- Thực hiện lệnh else: thap\_ha\_noi(1, “B”, “C”, “A”).

**Bước 11:** thap\_ha\_noi(1, “B”, “C”, “A”).

- Kiểm tra điều kiện n =1, thỏa mãn.

- Thực hiện lệnh if: in ra “Chuyển đĩa từ cột B sang cột C”.

**Quay lại bước 10**: thap\_ha\_noi(2, “B”, “A”, “C”).

- Thực hiện in ra “ Chuyển đĩa 2 từ cột B sang cột A”.

- Thực hiện thap\_ha\_noi(1, “C”, “A”, “B”).

**Bước 12:** thap\_ha\_noi(1, “C”, “A”, “B”).

- Kiểm tra điều kiện n =1, thỏa mãn.

- Thực hiện lệnh if in ra “ Chuyển 1 đĩa C sang cột A”.

**Quay lại bước 9:** thap\_ha\_noi(3, “B”, “C”, “A”).

- Thực hiện in ra “Chuyển từ đĩa 3”

Vậy kết quả là “Chuyển từ đĩa 3”