

HÀM TRONG PYTHON (T1)

Lưu vào thư mục LUUPYTHON với tên hambt1,hambt2,...

1. Kiểm tra n là số nguyên tố nếu đúng in 1, sai in 0 (cho I chạy từ 1- \sqrt{n} để tối ưu hóa thuật toán thay vì chạy đến n-1 không cần thiết)
2. In tổng các chữ số của n
3. In tổng các chữ số chẵn của n
4. In tổng chữ số của n là số nguyên tố
5. In số đảo ngược của n. Ví dụ 123 in 312
6. In số lượng ước của n là số nguyên tố (ước của n là số nguyên tố chính là thừa số nguyên tố)
7. In ước nguyên tố lớn nhất của n
8. Nhập vào một số nguyên n. Kiểm tra xem n có phải là số hoàn hảo hay không? Nếu hoàn hảo in “yes”, không hoàn hảo in “no”
9. Nhập vào một số nguyên n. Kiểm tra xem n có phải là số chính phương hay không? Nếu là số chính phương in “yes”, không là số chính phương in “no”
10. Kiểm tra nếu n tồn tại ít nhất một số 6, nếu đúng in 1, sai in 0.
11. Tính tổng giai thừa các chữ số của n và in ra.

Ví dụ: n=123

Tổng=1! +2!+3!

12. Kiểm tra nếu tổng chữ số của n chia hết cho 8, nếu đúng in 1 sai in 0
13. Kiểm tra số n có phải chỉ được tạo từ 1 số hay không?

Ví dụ: 222, 333, 9999999. Đúng in ra yes, sai in ra no.

14. Kiểm tra n có phải có chữ số tận cùng là lớn nhất hay không, tức là không có chữ số nào của n lớn hơn chữ số hàng đơn vị của nó. Nếu đúng in 1 sai in 0

15. In tổng lũy thừa chữ số của n với số mũ là số chữ số.

Ví dụ:

$123=1^3+2^3+3^3$