**BÀI 31. THỤC HÀNH VIẾT CHƯƠNG TRÌNH ĐƠN GIẢN**

**Nhiệm vụ 1.**Viết chương trình nhập từ bàn phím số tự nhiên n, kiểm tra n có phải là số nguyên tố hay không. Nếu n là hợp số thì in ra kết quả phân tích n thành tích các thừa số nguyên tố. Chú ý số 1 không là nguyên tố và cũng không là hợp số.

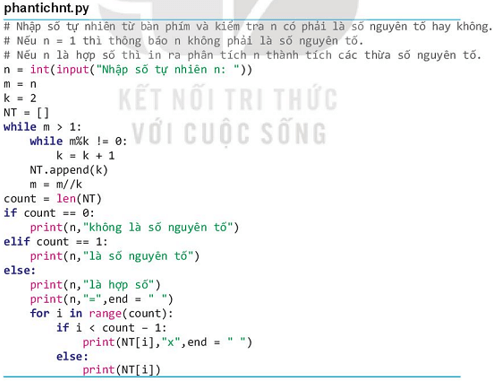
**Hướng dẫn**:

Sử dụng biến danh sách NT để lưu các thửa số nguyên tố của n. Chương trình sẽ thiết lập danh sách NT chỉ khi n > 1. Kết quả của chương trình sẽ như sau:

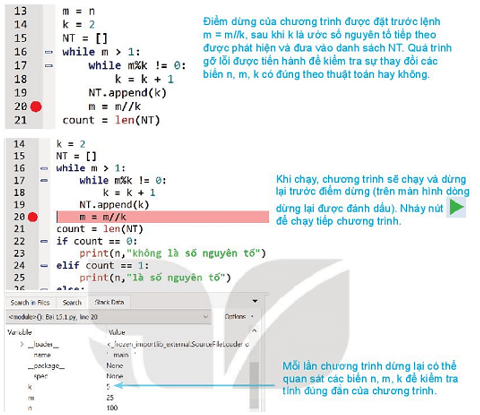
- Nếu n = 1 thị danh sách NT sẽ rỗng.

- Nếu n > 1 thị danh sách NT không rỗng. Độ dài danh sách len (NT) sẽ bằng 1 khi và chỉ khi n là số nguyên tố.

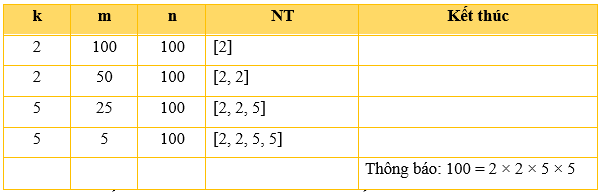
Nếu len(NT) > 1 thị chương trình sẽ in ra khai triển n thành tích các thửa số nguyên tố, khai triển này sẽ có dạng: n = p1 × p2 × …× px.



Chạy chương trình với công cụ gỡ lỗi của phần mềm lập trình. Thiết lập một điểm dừng tại dòng 20 của chương trình như sau:



Thiết lập bảng theo dõi các giá trị trung bình k, m, n, NT sẽ như sau, giả sử giá trị nhập ban đầu của n = 100.

****

**Nhiệm vụ 2.** Viết chương trình nhập từ bàn phím ba số thực a, b, c và tìm nghiệm phương trình bậc hai: ax2 + bx + c = 0.

Chương trình cần xét đầy đủ các trường hợp xảy ra.

**Hướng dẫn:**

Với bộ dữ liệu a, b, c đã nhập (là các số thực), chúng ta cần xét đầy đủ các trường hợp sau:

- Nếu a = b = c = 0 phương trình có vô số nghiệm.

- Nếu a = b = 0; c ≠ 0, phương trình vô nghiệm.

- Nếu a = 0; b ≠ 0 phương trình là bậc nhất và có nghiệm duy nhất.

- Nếu a ≠ 0, giải phương trình bậc hai. Nghiệm sẽ phụ thuộc vào giá trị delta = b2 – 4ac.

Phương trình vô nghiệm, có một nghiệm kép hoặc hai nghiệm phân biệt phụ thuộc vào giá trị delta là nhỏ hơn 0, bằng 0 hay lớn hơn 0.

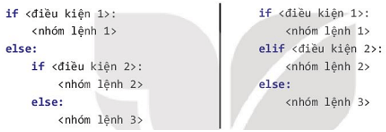
Chương trình được thiết kế thông qua các hàm sau:

- NhapDL(): hàm nhập ba số a, b, c từ bàn phím.

- GiaiPT1(b, c): hàm giải phương trình bậc nhất bx + c = 0

- Gia PT2a, b, c); hàm giải phương trình bậc hai: ax2 + bx + c = 0.

Trong bài thực hành chúng ta sử dụng cấu trúc mở rộng của lệnh rẽ nhánh **if … else** trong Python khi các lệnh này lồng nhau. Khi đó các lệnh rẽ nhánh lồng nhau trong mô hình bên trái sẽ được viết gọn hơn như mô hình bên phải.



**Chú ý:**Cấu trúc **if … elif …. else**có thể lồng nhau nhiều lần.

Chương trình đầy đủ như sau:

