**Mini-project KTMT**

**Bài 3 + Bài 10**

**Đào Minh Nhật – 20215107**

**Nguyễn Trọng Nghĩa - 20215101**

Bài 3:

Code:

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, số

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động

Giải thích:

Khai báo cách đếm số từ 0 đến 19 và lưu trong mảng digits

Khai báo cách đếm của 20, 30, … 90 và lưu trong mảng ge20

Cách làm: tách lần lượt 3 số và đọc.

+ Kiểm tra phần tử triệu:

Lấy $a0 \ 1000000

Nếu thương = 0 thì chuyển sang kiểm tra phần tử nghìn

Ngược lại thương != 0 thì gọi tới hàm “print\_below\_thousand “ để đọc số 3 số hàng triệusau đó in thêm đơn vị triệu.

+Kiểm tra phần tử nghìn : tương tự phần tử triệu, lấy $a0 \ 1000

+ Hàm “print\_below\_thousand “: in cách đọc của phần tử < 1000.

Lấy $a0 \ 100 để đọc hàng trăm nếu có. Sau đó đọc hàng chục và hàng đơn vị.

+ Hàm “print\_digits”: In hàng chục, hàng đơn vị

Nếu $a0 < 20 thì xâu cần tìm ở mảng digits

Ngược lại $a0 >= 20 thì thực hiện hàm “print\_dozen”, lấy xâu cần tìm ở mảng ge20

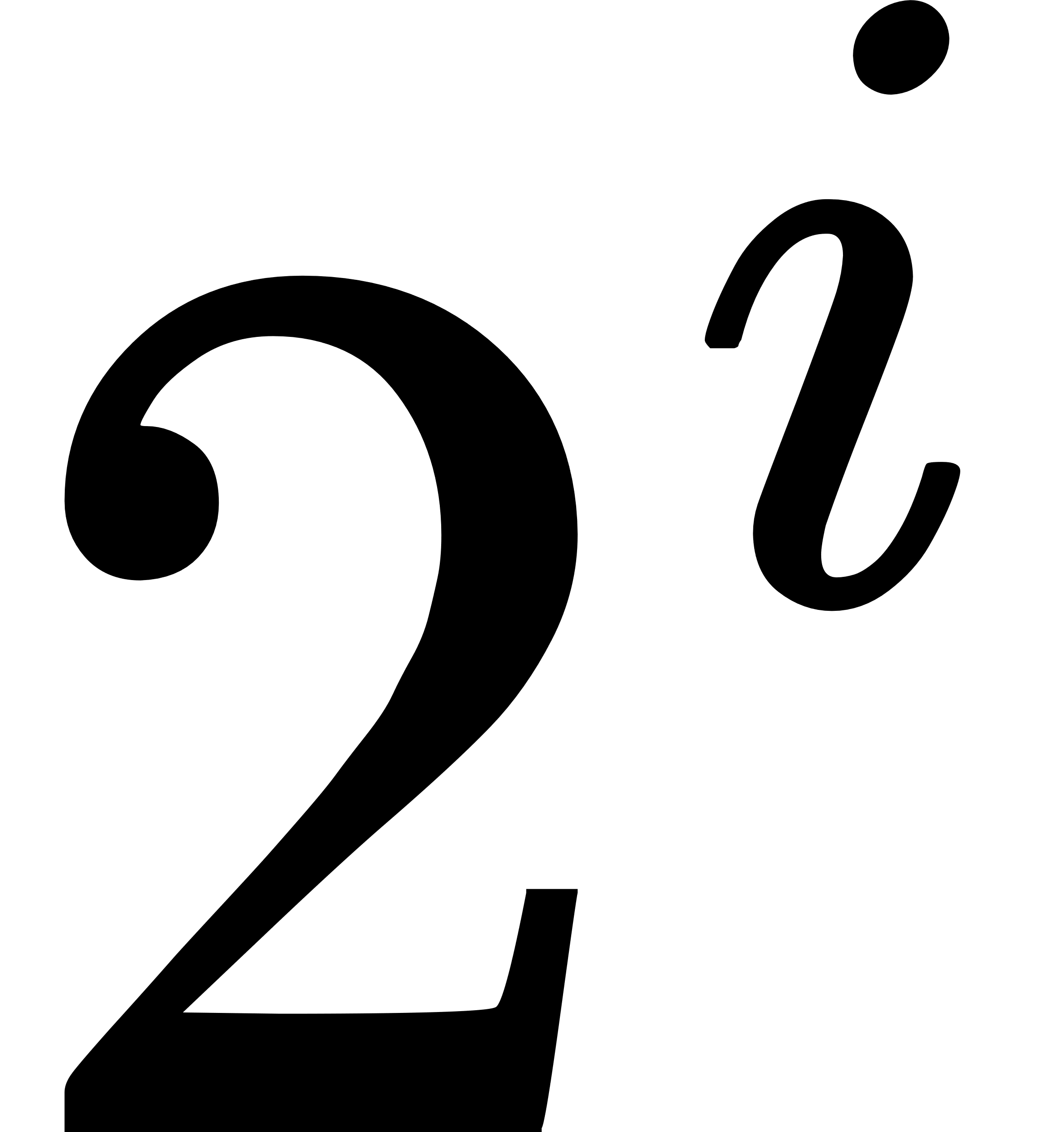
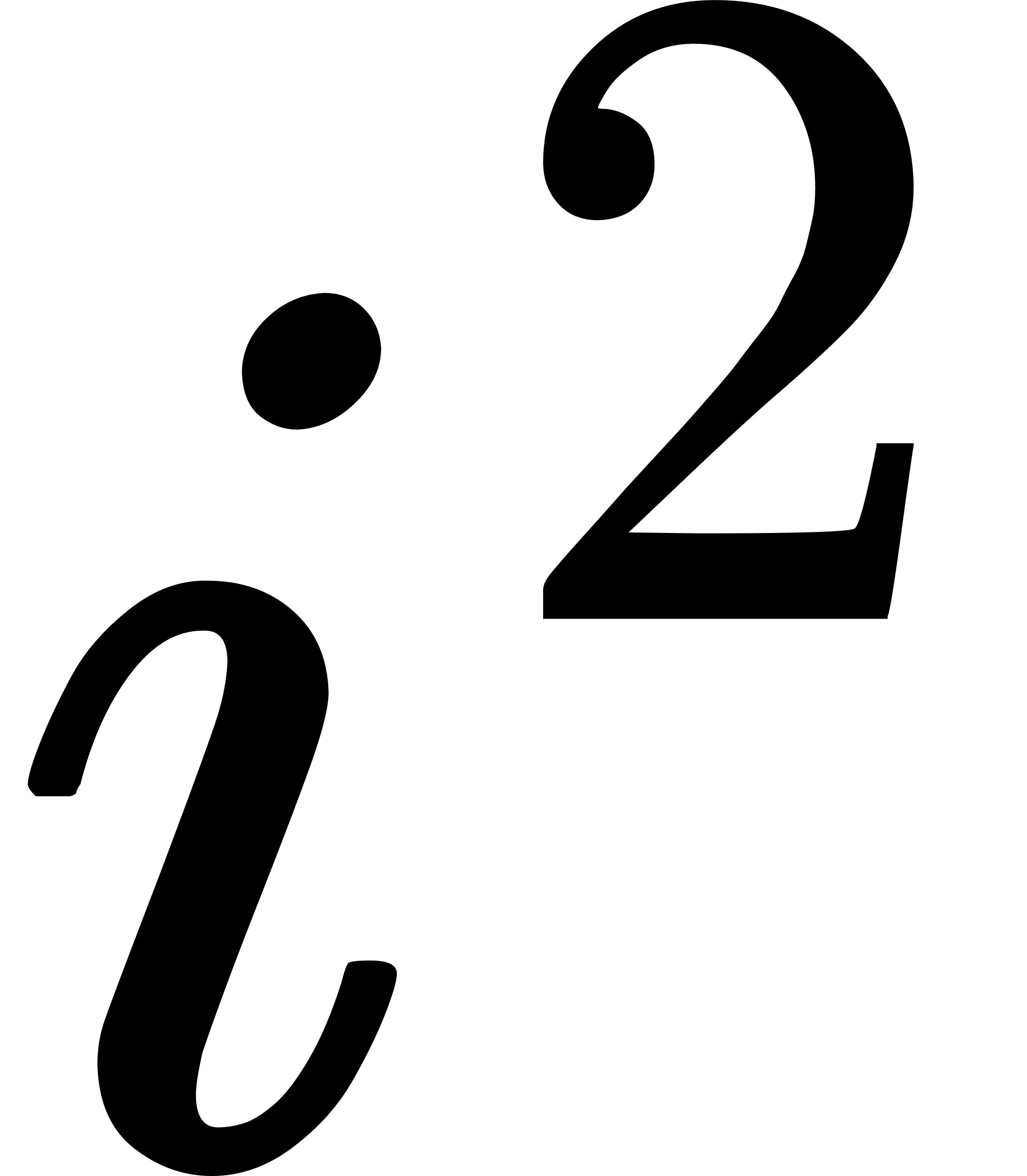
+ Hàm “print\_dozen”: in cách đọc của các số hàng chục từ 20 đến 90.

Nếu có hàng đơn vị thì quay lại hàm print\_digits để đọc số hàng đơn vị.

Kết quả: Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

Bài 10:

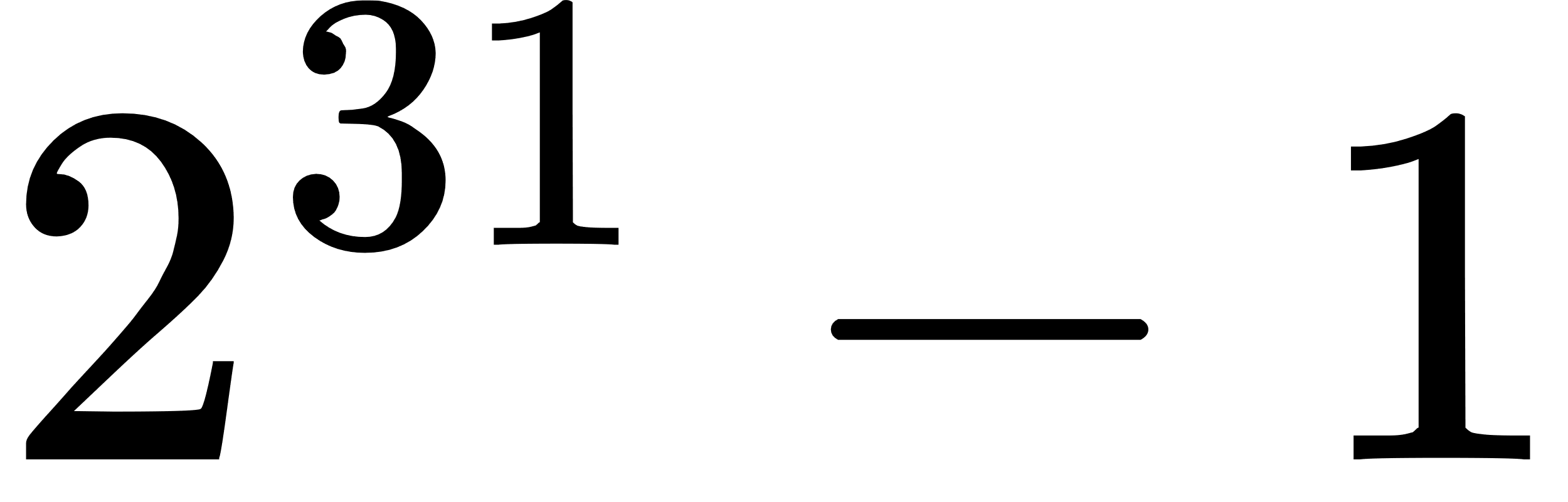
Ở project 10 có 3 functions chính cần phải làm là: hàm mũ  ,  và functions chuyển từ số thập phân sang thập lục phân với mà dạng thập lục phân phải ở dạng đơn giản nhất.

Với hàm mũ wps, ta khởi tạo một thanh ghi có giá trị ban đầu là 1, một thanh ghi lưu lại giá trị i. Sau đó, ta sử dụng vòng lặp i vòng để dịch trái thanh ghi có giá trị ban đầu là 1, đồng thời cũng giảm giá trị i sau mỗi vòng lặp đến khi i = 0 thì dừng.

Trong không gian 32-bit, ta có thể dịch tối đa 31 lần để tránh hiện tượng tràn. Vì vậy số i nhập vào có giá trị trong khoảng [0; 31] để chương trình được thực hiện. Với i<0 thì chưa nghĩ ra giải pháp.

Với hàm mũ wps, ta không thể dùng các phép toán logic để tính toán mà phải dùng phép nhân số học (mul) để thực hiện tính toán.

Vì wps nên nếu người dùng nhập vào một số nguyên âm thì có thể dùng một hàm tính trị tuyệt đối bằng cách lấy số bù 2.

Trong không gian 32-bit, số dương lớn nhất có thể có là . Vì vậy nên số i sẽ nằm trong khoảng wps tức (-65536, 65536). Vì có số dương lớn nên khi nhân với nhau phải dùng lệnh mulou.

Với hàm chuyển đổi số thập phân sang số thập lục phân ở dạng tối giản, ta sử dụng vòng lặp do-while lưu kết quả vào mảng 8 ô nhớ lưu trữ các ký tự sau khi chuyển đổi và dùng vòng lặp in ra màn hình từng ô nhớ để hiển thị kết quả.

Ta khởi tạo một thanh ghi có giá trị từ 0-32 để chỉ vị trí ô nhớ trong mảng kết quả.

Ta sử dụng một thanh ghi có giá trị 0xF rồi sử dụng phép toán and với i để lấy nội dung 4 bit cuối, lưu vào một thanh ghi khác. Thanh ghi này sẽ cập nhật giá trị của mình sao cho đúng với ký tự trên bảng mã ASCII. Nếu thanh ghi có giá trị từ 48 đến 57 thì là số từ ‘0’-’9’, nếu không thì là chữ cái, cộng thêm 7 để có giá trị từ 65 đến 70. Sau đó lưu giá trị vào mảng kết quả, tăng giá trị thanh ghi trỏ đến vị trí ô nhớ trong mảng thêm 4 để sang ô nhớ tiếp theo đồng thời cũng dịch phải thanh nhớ chứa giá trị i.

Đến khi i = 0 hoặc thanh ghi trỏ đến vị trí ô nhớ đạt giá trị 32 thì dừng vòng lặp do-while. Nếu người dùng nhập i=0 thì chương trình vẫn hiện ra ‘0x0’.

Vòng lặp in kết quả sau đó sẽ lấy kết quả từ thanh ghi trỏ đến vị trí ô nhớ trong mảng, từ đó truy ngược lại đến ô nhớ đầu tiên, mỗi lần truy ngược lại sẽ in ra màn hình ký tự trong bảng ASCII.

Source code:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Hệ điều hành

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, tài liệu

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, số

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động