Đây là một chương trình mã nguồn tiếng Assembly cho mô hình "tiny". Dưới đây là giải thích từng dòng mã:

```

.model tiny

.stack 100h

```

Dòng đầu tiên `.model tiny` chỉ định rằng chương trình sẽ được biên dịch và chạy trong mô hình "tiny". Dòng tiếp theo `.stack 100h` xác định kích thước của ngăn xếp (stack) là 100h (256 byte).

```

.data

tb1 db 'Nhap so thu nhat : $'

tb2 db 13,10,'Nhap so thu hai : $'

tb3 db 13,10,'Tong hai so la : $'

so1 db 0

so2 db 0

```

Phần `.data` chứa các biến và chuỗi dữ liệu. `tb1`, `tb2`, và `tb3` là các chuỗi thông báo hiển thị cho người dùng. `so1` và `so2` là các biến byte để lưu giá trị của số thứ nhất và số thứ hai.

```

.code

main proc

mov ax,@data

mov ds,ax

```

Phần `.code` bắt đầu mã chương trình. Dòng `main proc` khai báo một thủ tục (procedure) có tên là `main`. Các dòng tiếp theo `mov ax,@data` và `mov ds,ax` đặt thanh ghi `DS` bằng giá trị của thanh ghi `AX`, nơi chứa địa chỉ của phần dữ liệu (`.data`) đã được khai báo.

```

lea dx,tb1

mov ah,9

int 21h

```

Dòng `lea dx,tb1` tải địa chỉ của chuỗi `tb1` vào thanh ghi `DX`. Dòng `mov ah,9` thiết lập thanh ghi `AH` để gọi hàm xuất chuỗi. Dòng `int 21h` gọi hàm hệ thống để hiển thị chuỗi thông báo `tb1` lên màn hình.

```

nhapso1:

mov ah,1

int 21h

cmp al,13

je hientb2

sub al,30h

mov cl,al

mov al,so1

mov bl,10

mul bl

add al,cl

mov so1,al

jmp nhapso1

```

Nhãn `nhapso1` là một điểm nhảy để thực hiện việc nhập số thứ nhất từ bàn phím. Dòng `mov ah,1` chuẩn bị thanh ghi `AH` để gọi hàm đọc một ký tự từ bàn phím. Dòng `int 21h` gọi hàm hệ thống để đọc một ký tự từ bàn phím và lưu giá trị ASCII của ký tự đó vào thanh ghi `AL`. Dòng `cmp al,13` so sánh giá trị của `AL` với 13 (ký tự xuống dòng). Nếu hai giá trị bằng nhau (`je hientb2`), chương trình nhảy đến nhãn `hientb2` để hiển thị thông báo cho người dùng nhập số thứ hai. Ngược lại, chương trình tiếp tục thực hiện các dòng lệnh để chuyển đổi ký tự sang số và cộng dồn vào biến `so1`, sau đó nhảy lại `nhapso1` để nhập tiếp.

```

hientb2:

lea dx,tb2

mov ah,9

int 21h

```

Sau khi nhập xong số thứ nhất, chương trình nhảy đến nhãn `hientb2`. Dòng `lea dx,tb2` tải địa chỉ của chuỗi `tb2` vào thanh ghi `DX`. Dòng `mov ah,9` thiết lập thanh ghi `AH` để gọi hàm xuất chuỗi. Dòng `int 21h` gọi hàm hệ thống để hiển thị chuỗi thông báo `tb2` lên màn hình.

```

nhapso2:

mov ah,1

int 21h

cmp al,13

je hientb3

sub al,30h

mov cl,al

mov al,so2

mov bl,10

mul bl

add al,cl

mov so2,al

jmp nhapso2

```

Tương tự như việc nhập số thứ nhất, nhãn `nhapso2` cho phép người dùng nhập số thứ hai từ bàn phím. Dòng `mov ah,1` và `int 21h` đọc một ký tự từ bàn phím và lưu giá trị ASCII của ký tự đó vào thanh ghi `AL`. Nếu giá trị của `AL` là 13 (ký tự xuống dòng), chương trình nhảy đến nhãn `hientb3` để hiển thị kết quả. Ngược lại, chương trình thực hiện các dòng lệnh để chuyển đổi ký tự sang số và cộng dồn vào biến `so2`, sau đó nhảy lại `nhapso2` để nhập tiếp.

```

hientb3:

lea dx,tb3

mov ah,9

int 21h

```

Sau khi nhập xong số thứ hai, chương trình nhảy đến nhãn `hientb3`. Dòng `lea dx,tb3` tải địa chỉ của chuỗi `tb3` vào thanh ghi `DX`. Dòng `mov ah,9` thiết lập thanh ghi `AH` để gọi hàm xuất chuỗi. Dòng `int 21h` gọi hàm hệ thống để hiển thị chuỗi thông báo `tb3` lên m

àn hình.

```

mov al,so1

add al,so2

mov ah,0

mov cx,0

mov bl,10

```

Dòng `mov al,so1` di chuyển giá trị của biến `so1` vào thanh ghi `AL`. Dòng `add al,so2` thực hiện phép cộng giữa `AL` và giá trị của biến `so2`, kết quả được lưu trong `AL`. Dòng `mov ah,0` đặt giá trị 0 vào thanh ghi `AH`. Dòng `mov cx,0` và `mov bl,10` đặt giá trị 0 vào thanh ghi `CX` và giá trị 10 vào thanh ghi `BL`.

```

chia:

div bl

push ax

inc cx

cmp al,0

je hien

mov ah,0

jmp chia

```

Nhãn `chia` bắt đầu một vòng lặp để chia liên tiếp cho đến khi giá trị của `AL` là 0. Dòng `div bl` thực hiện phép chia giữa giá trị của `AX` (được tạo bởi `AL` và `AH`) và giá trị của `BL`. Kết quả của phép chia được lưu trong `AL`, phần dư được lưu trong `AH`. Dòng `push ax` đẩy giá trị của `AX` vào ngăn xếp. Dòng `inc cx` tăng giá trị của `CX` lên 1. Dòng `cmp al,0` so sánh giá trị của `AL` với 0. Nếu giá trị bằng nhau (`je hien`), chương trình nhảy đến nhãn `hien` để hiển thị kết quả. Ngược lại, dòng `mov ah,0` đặt giá trị 0 vào thanh ghi `AH` và chương trình nhảy lại `chia` để tiếp tục vòng lặp.

```

hien:

pop ax

mov dl,ah

add dl,30h

mov ah,2

int 21h

loop hien

```

Nhãn `hien` là điểm nhảy để hiển thị kết quả trên màn hình. Dòng `pop ax` lấy giá trị từ đỉnh ngăn xếp và đặt vào thanh ghi `AX`. Dòng `mov dl,ah` di chuyển giá trị của `AH` vào thanh ghi `DL`. Dòng `add dl,30h` thêm 30h vào giá trị của `DL` để chuyển đổi từ giá trị ASCII sang giá trị thập phân. Dòng `mov ah,2` thiết lập thanh ghi `AH` để gọi hàm xuất ký tự. Dòng `int 21h` gọi hàm hệ thống để hiển thị ký tự. Dòng `loop hien` lặp lại quá trình hiển thị ký tự trong vòng lặp cho đến khi giá trị của `CX` bằng 0.

```

thoat:

mov ah,76

int 21h

main endp

```

Nhãn `thoat` là một điểm nhảy để thoát khỏi chương trình. Dòng `mov ah,76` thiết lập giá trị 76 (4Chh) vào thanh ghi `AH`, đại diện cho hàm thoát chương trình. Dòng `int 21h` gọi hàm hệ thống để thoát khỏi chương trình. Dòng `main endp` kết thúc thủ tục `main`.

Đây là một chương trình đơn giản để nhập hai số từ bàn phím, tính tổng và hiển thị kết quả trên màn hình.