**EJERCICIO 8**

Inicio del programa en la posición de memoria: 2269

Inicio de área de datos en la posición de memoria: 4165

**1. Una descripción de lo que el programa realiza**

Sumará valores enteros de 2 bytes cada uno, siendo elementos de una lista

**2. Para qué se usa cada registro:**

**SI**: es el registro en donde al elemento siguiente a sumar

**CL**: Es la parte baja del CX y representa los n elementos de un vector

**AX**: es el registro en donde se irá guardando los resultados de las sumas sucesivas

**3. Dónde se ubican las variables y qué valores les has asignado**:

4165 y 4166: dirección donde se guarda el primer elemento del vector

4167 y 4168: dirección donde se guarda el segundo elemento del vector

4169 y 416A: tendrá el valor del resultado cuando no de overflow en la suma de elementos del

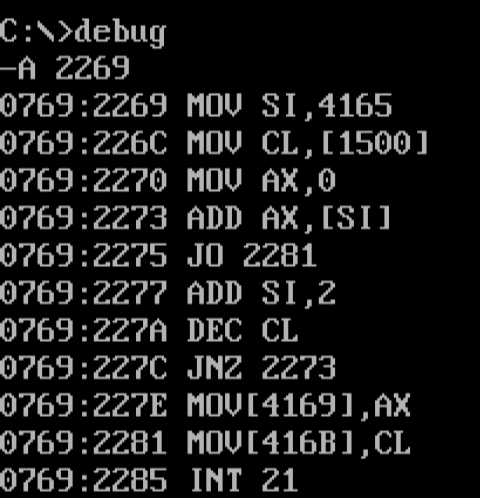
vector

416B y 416C: tendrá el valor del resultado habiendo dado overflow en la suma de elementos del

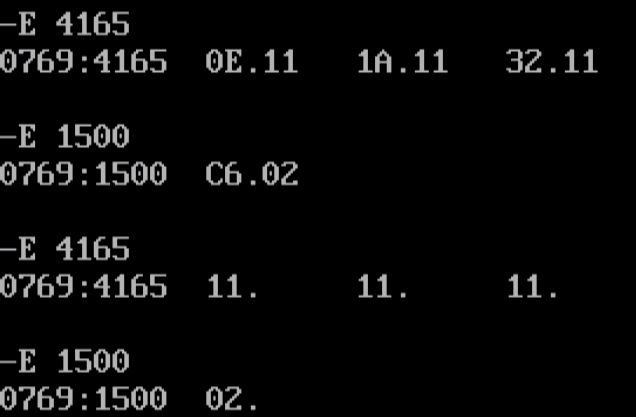
vector

1500: n elementos de un vector

**CAPTURAS DOSBOX**

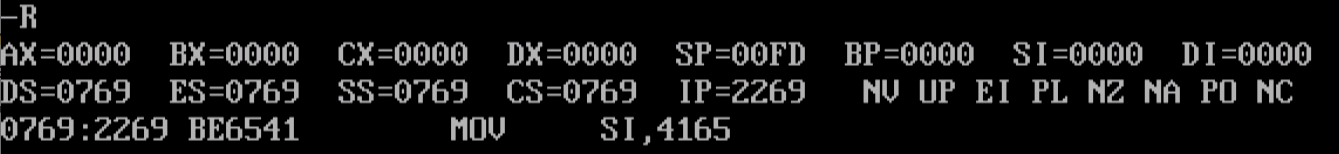


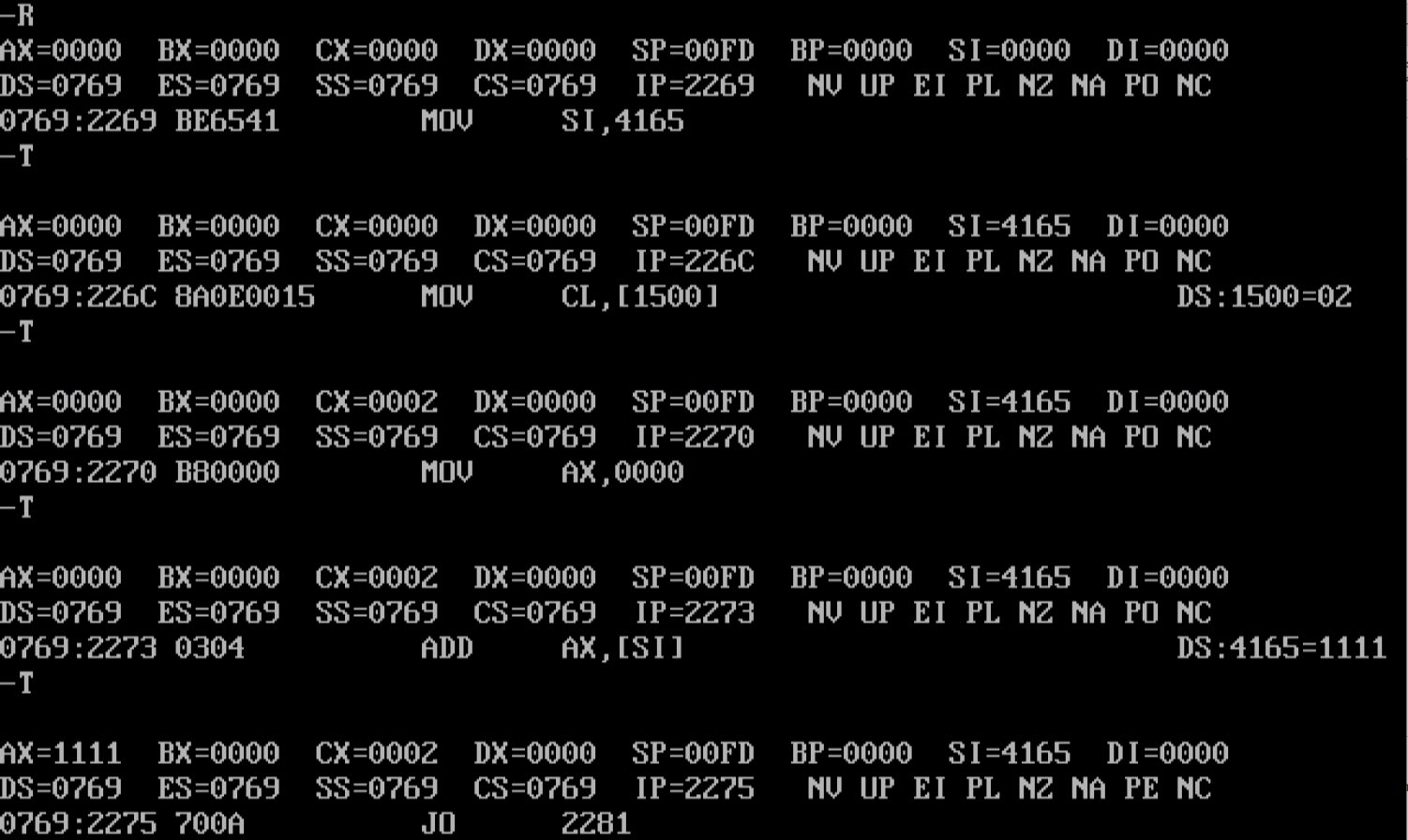
Desde 4165 se cargan los datos.

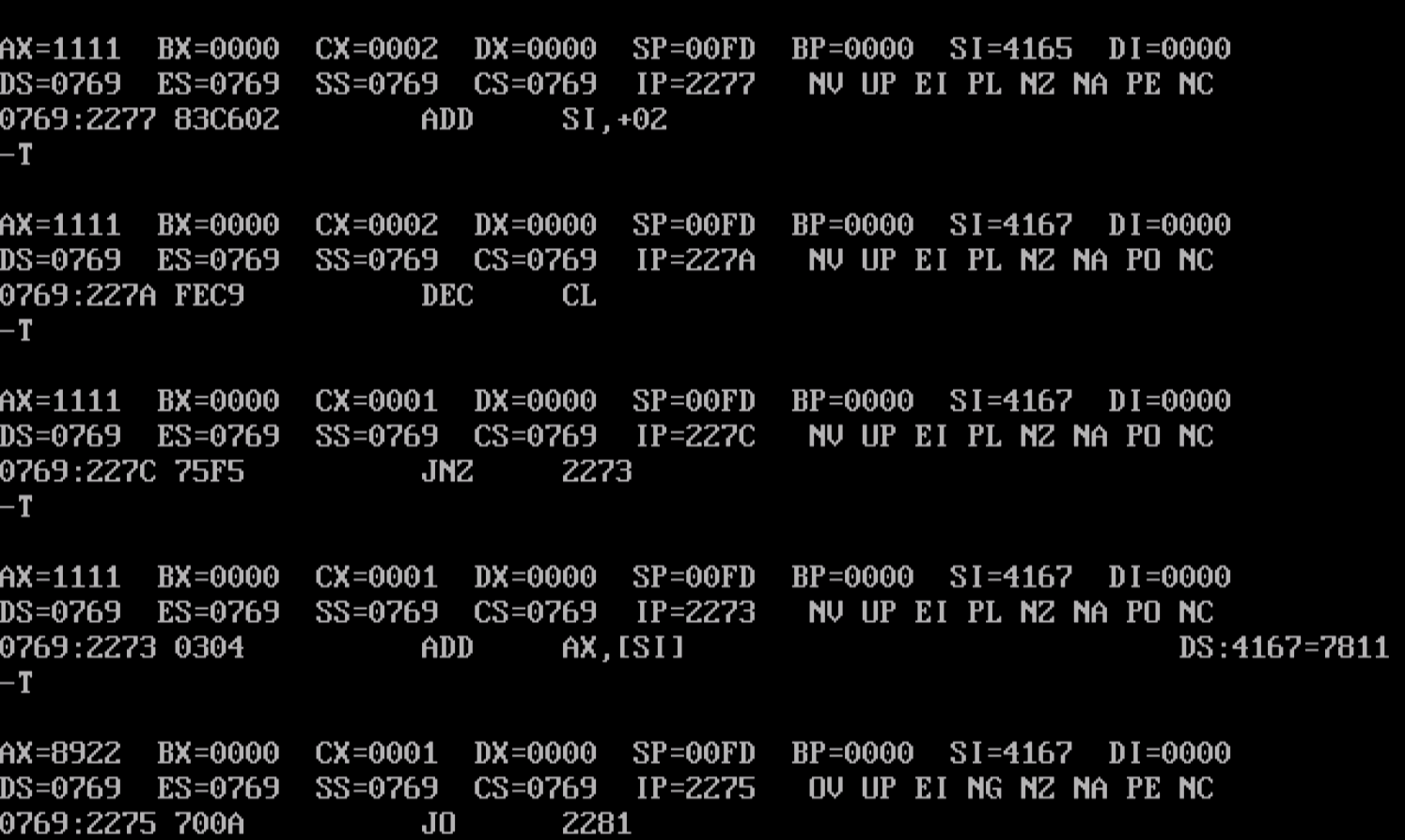


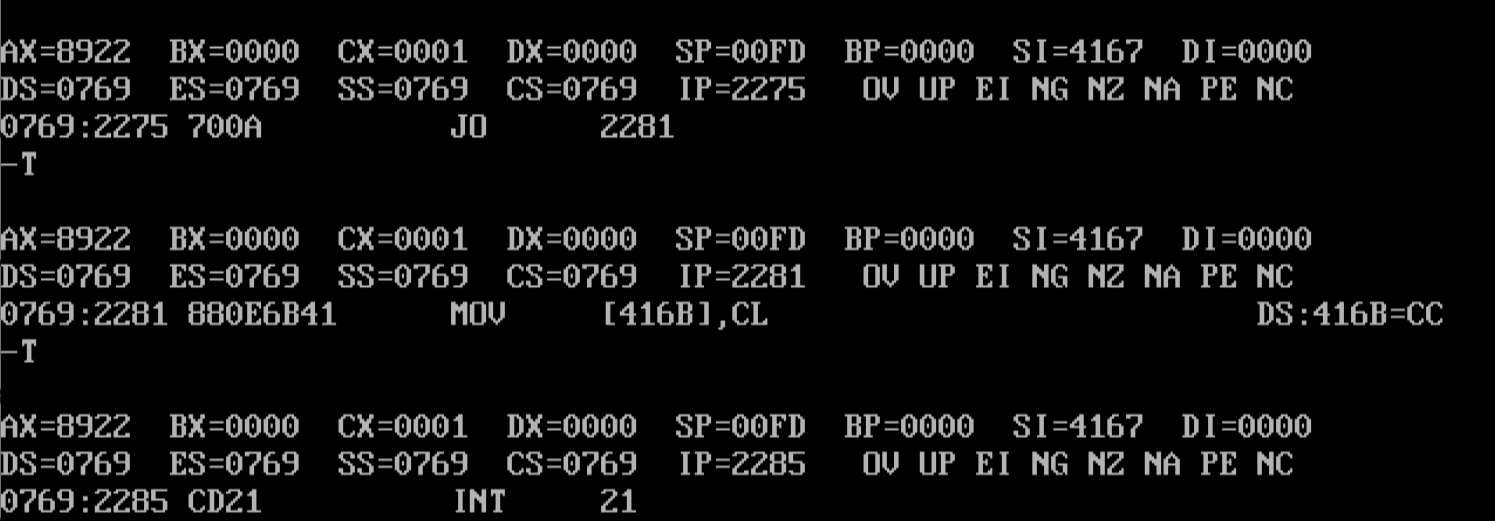
RIP

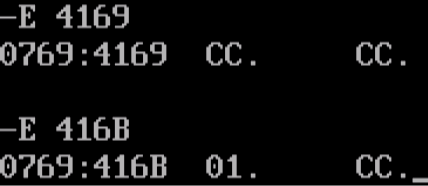












**EJERCICIO 9**

Inicio del programa en la posición de memoria: 2269

Inicio de área de datos en la posición de memoria: 4165

**1. Una descripción de lo que el programa realiza**

Se realizará una búsqueda de un elemento en una lista (en este caso la lista tendrá caracteres

codificados en ASCII de 8 bits y se buscará el carácter E, que es el 45h) y se lo compara con el que

se está buscando encontrar a través de una resta sin asignación del resultado, llevando un conteo

de las veces que se encontró en la lista. Además, dicho elemento se guardará en otra lista junto

con la posición en donde estaba.

**2. Para qué se usa cada registro:**

**SI**: es el registro que tendrá la dirección inicial de la lista de caracteres

**CL**: es el registro que guardara los n caracteres de la lista

**DX**: es el registro que tendrá la dirección inicial de la segunda lista que guardara las direcciones

donde se encontró el elemento a buscar

**AX**: dentro de este registro en AH, se guardará las veces que se encontró el elemento que

buscaremos en la lista y en AL se guardará el valor de la lista que está apuntando SI, es decir, se le

enviará cada uno de los valores de la lista (paso previo haciendo la comparación sin asignación del

resultado)

**3. Dónde se ubican las variables y qué valores les has asignado:**

4165: dirección donde se guarda el primer elemento del vector

4166: dirección donde se guarda el segundo elemento del vector

4167: dirección donde se guarda el tercer elemento del vector

4168: dirección donde se guarda el cuarto elemento del vector

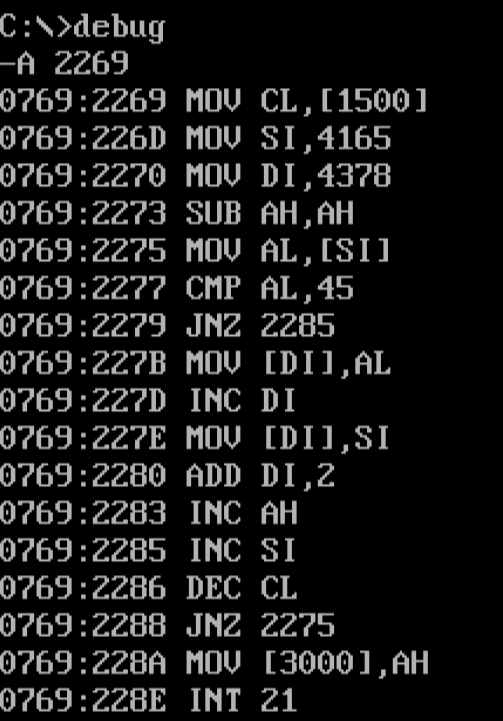
4169: dirección donde se guarda el quinto elemento del vector

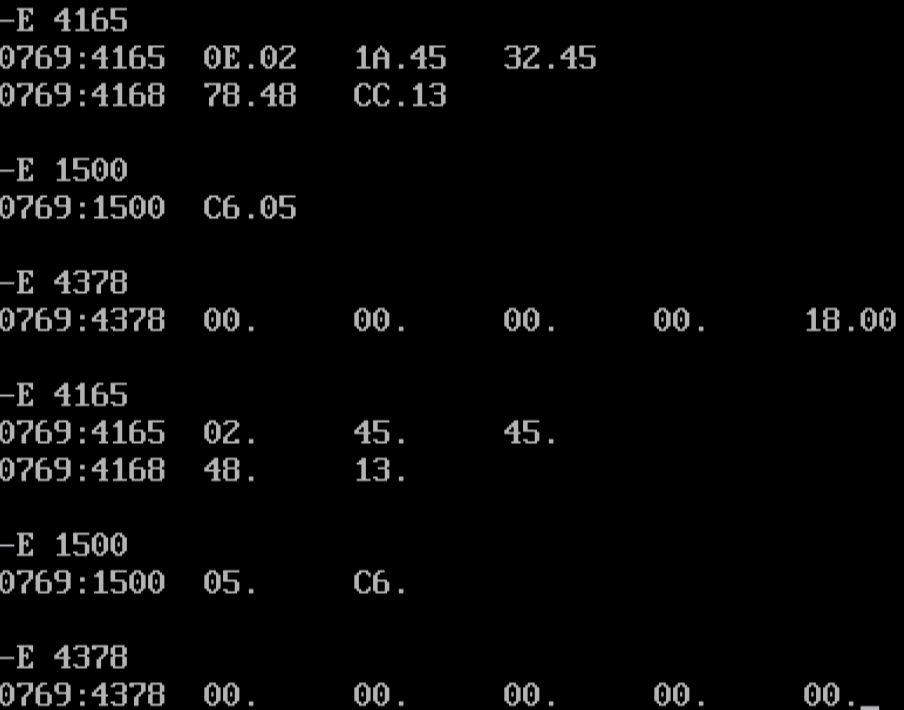
1500: n elementos del vector

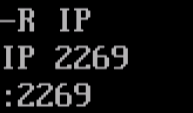
3000: el número k de veces que se encontró el elemento buscado. En nuestro ejemplo el elemento a buscar será la letra E y esta posición la cargará el programa con el valor dependiendo de cuantas E haya en la lista.

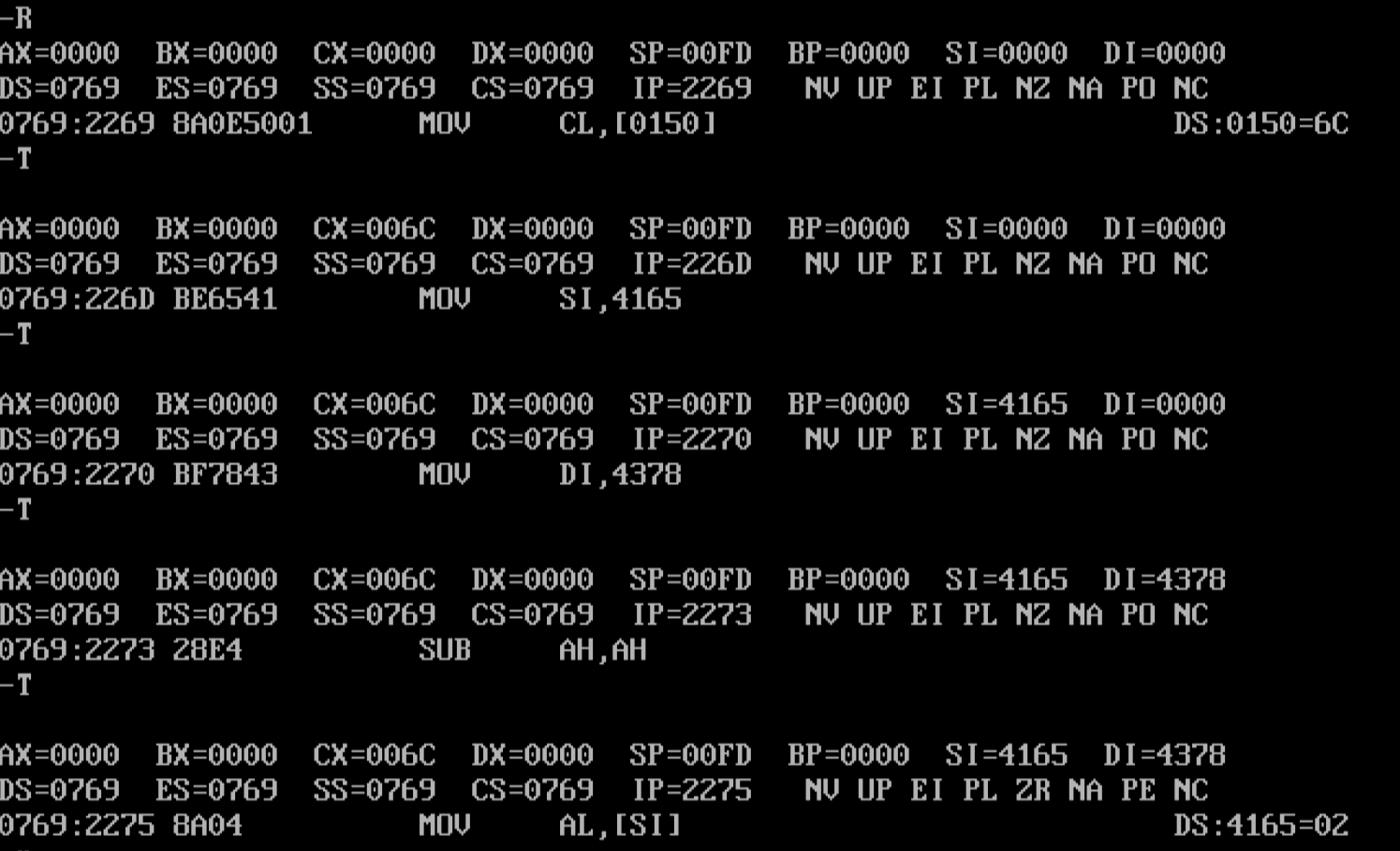
4378: desde esta posición, se guardará la segunda lista.

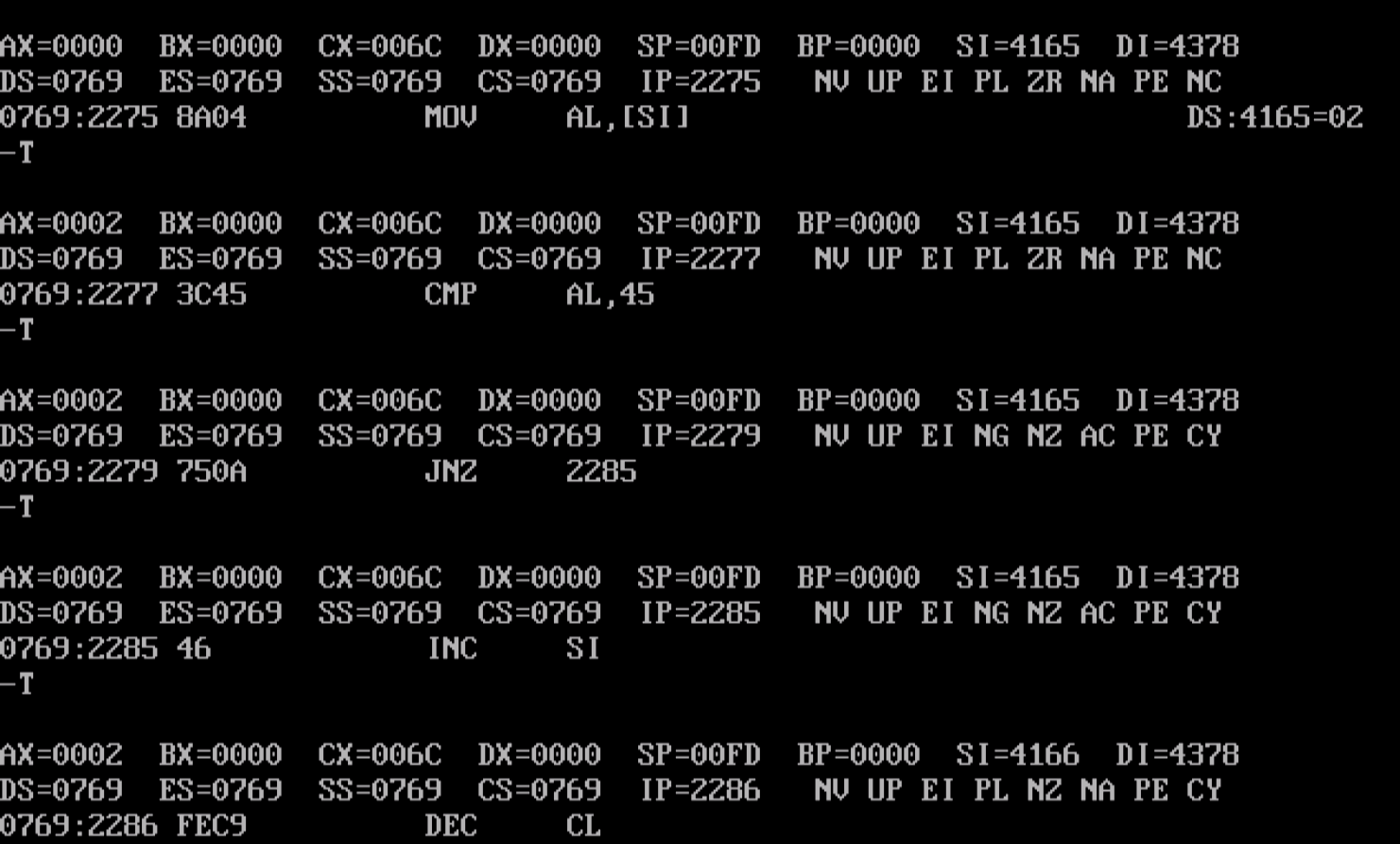
**CAPTURAS DOSBOX**

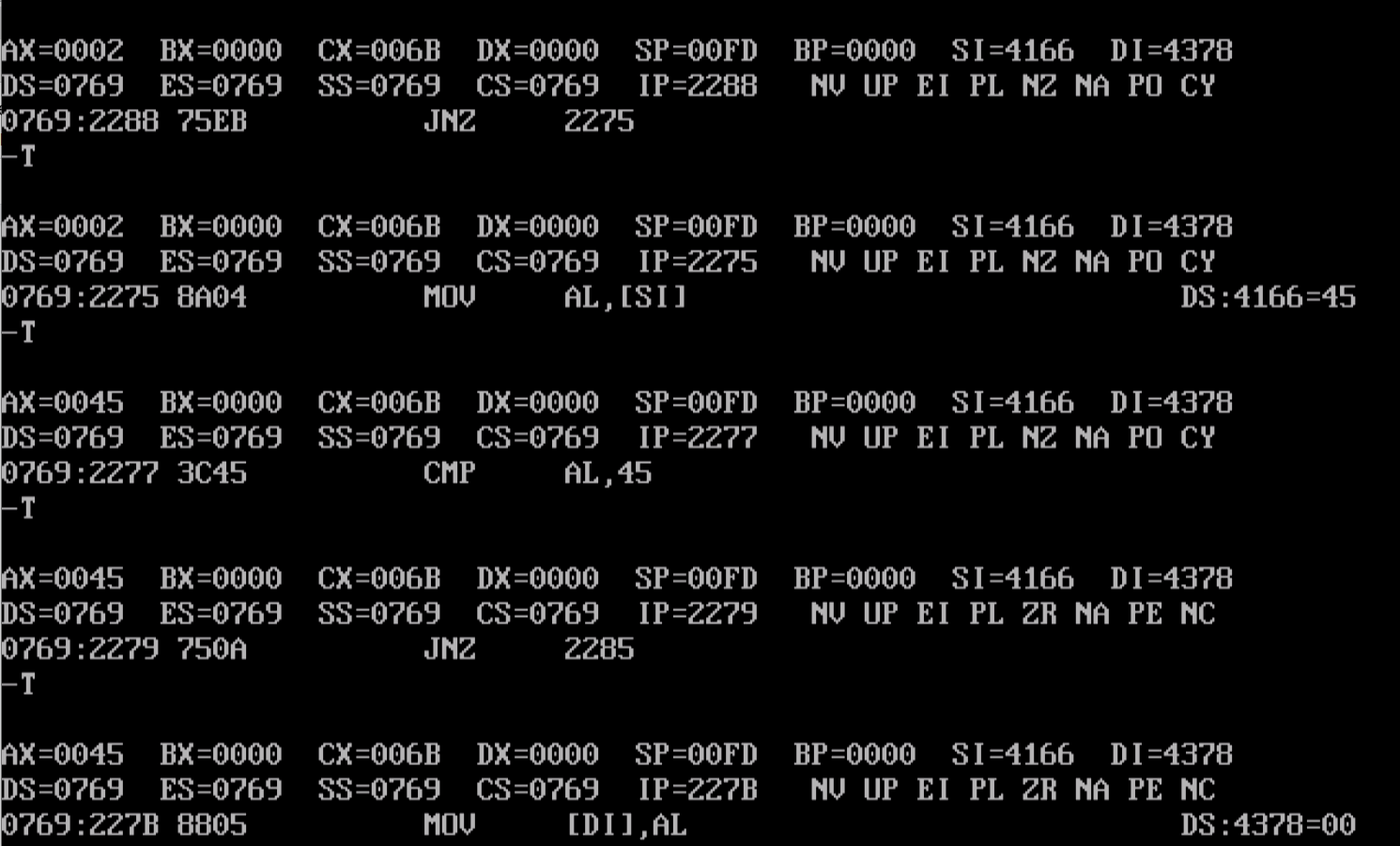
****

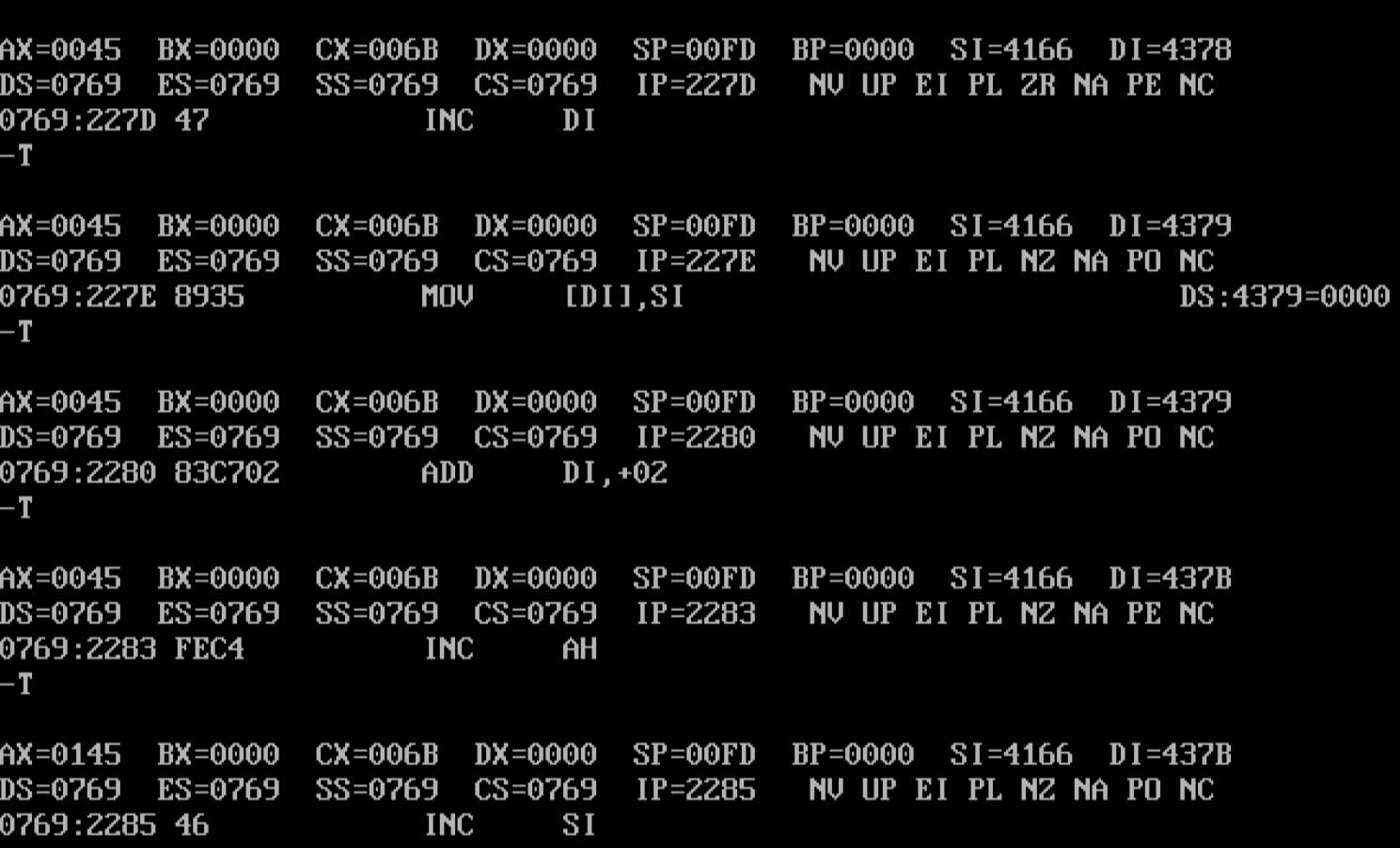
****



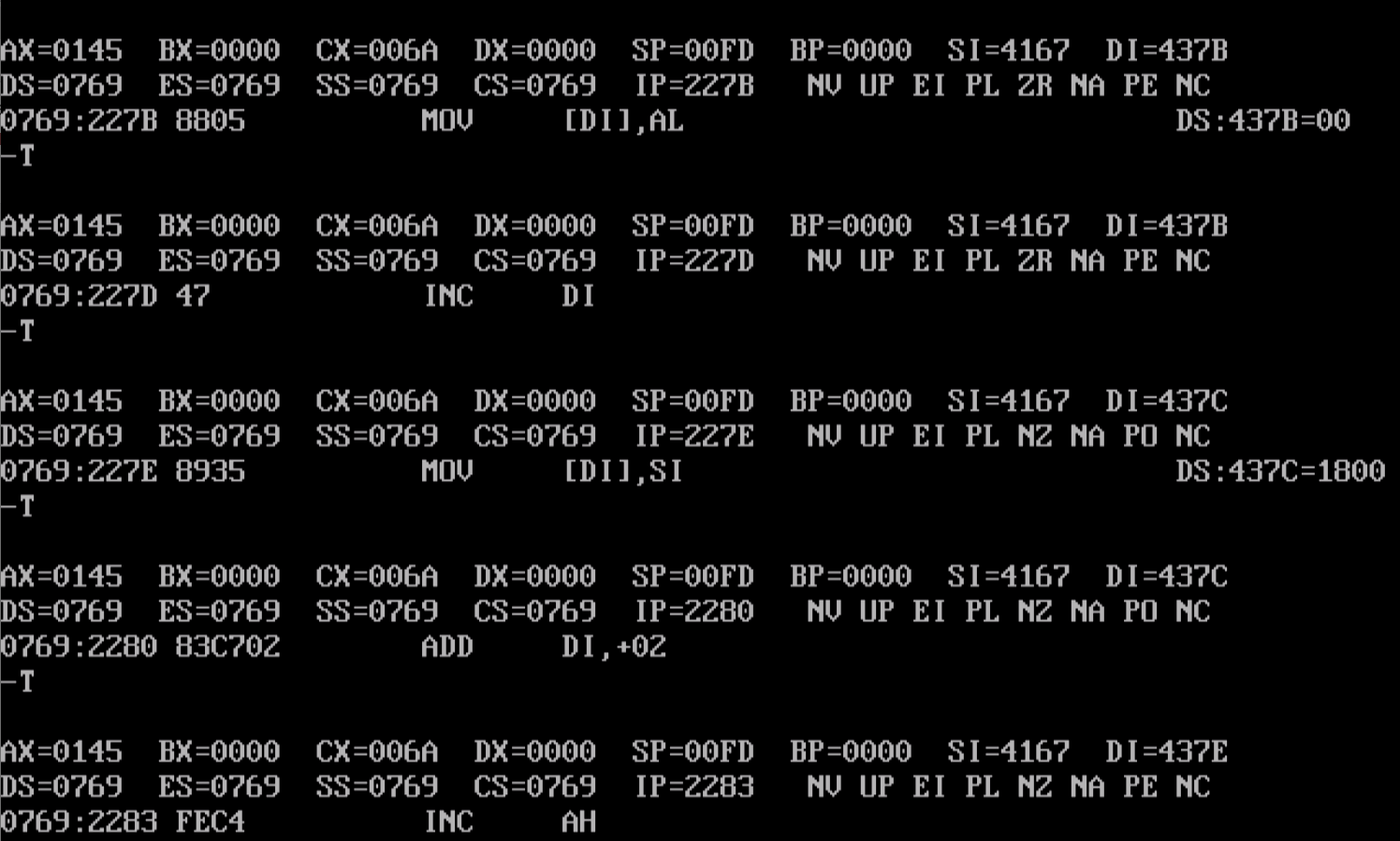


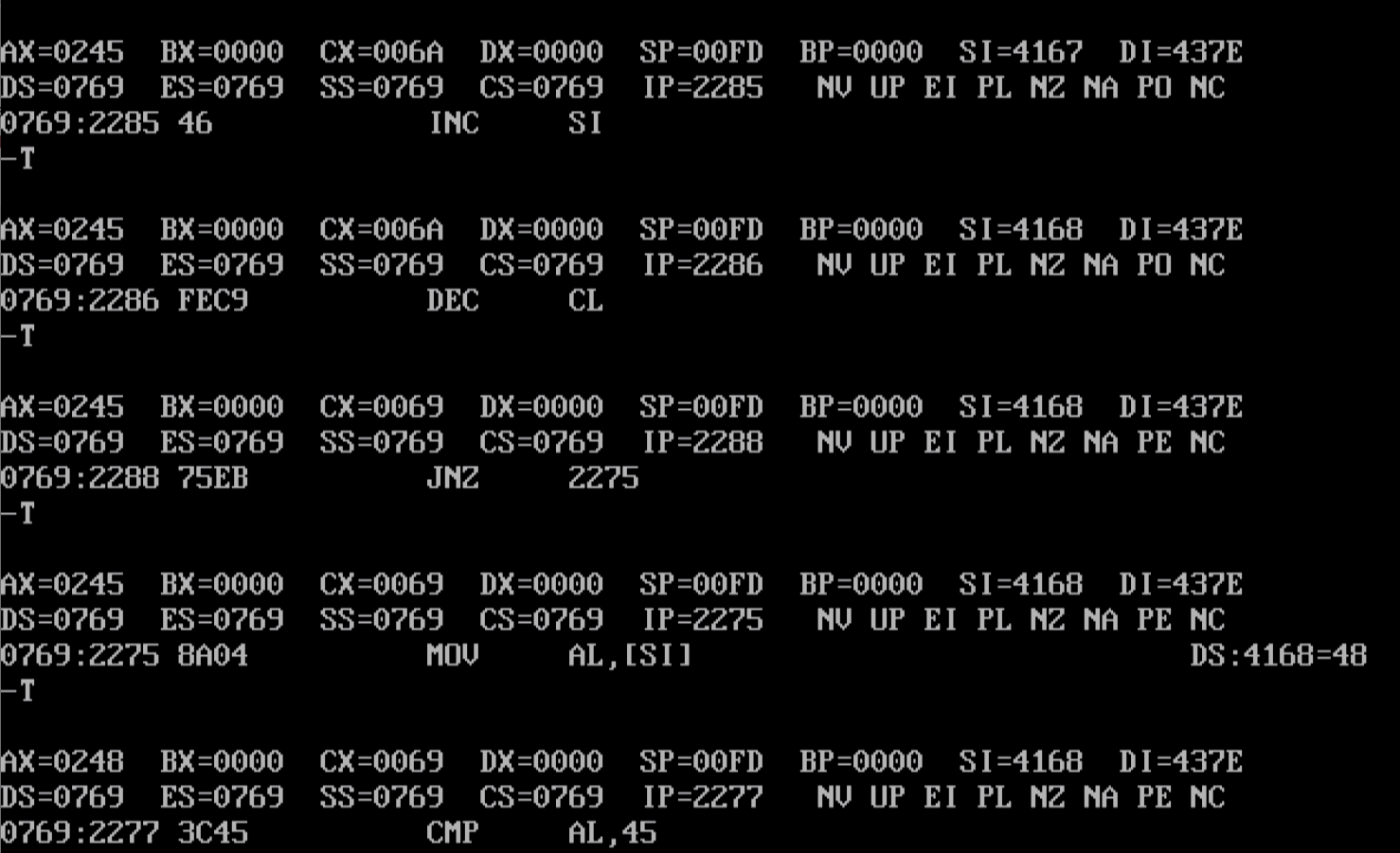


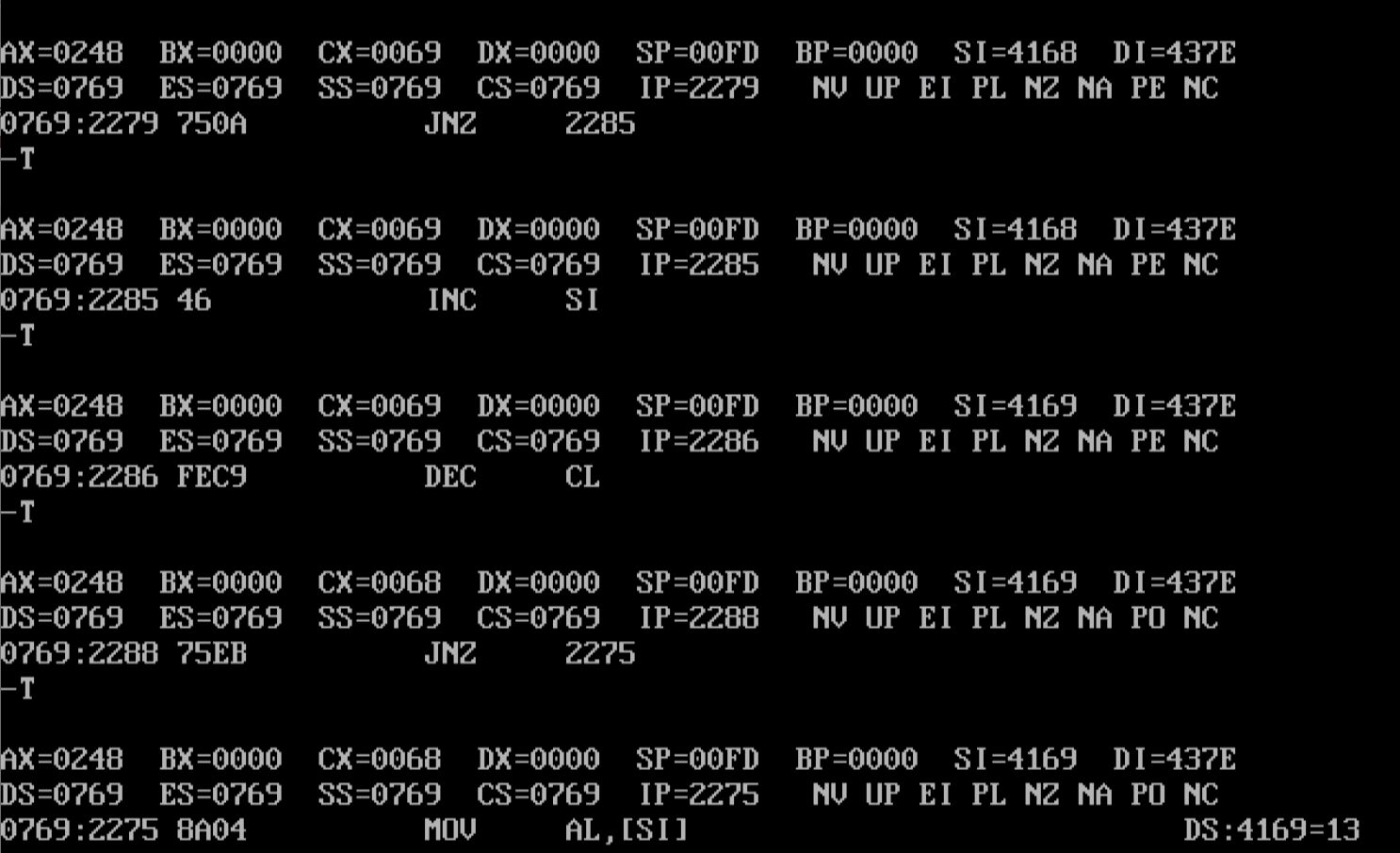


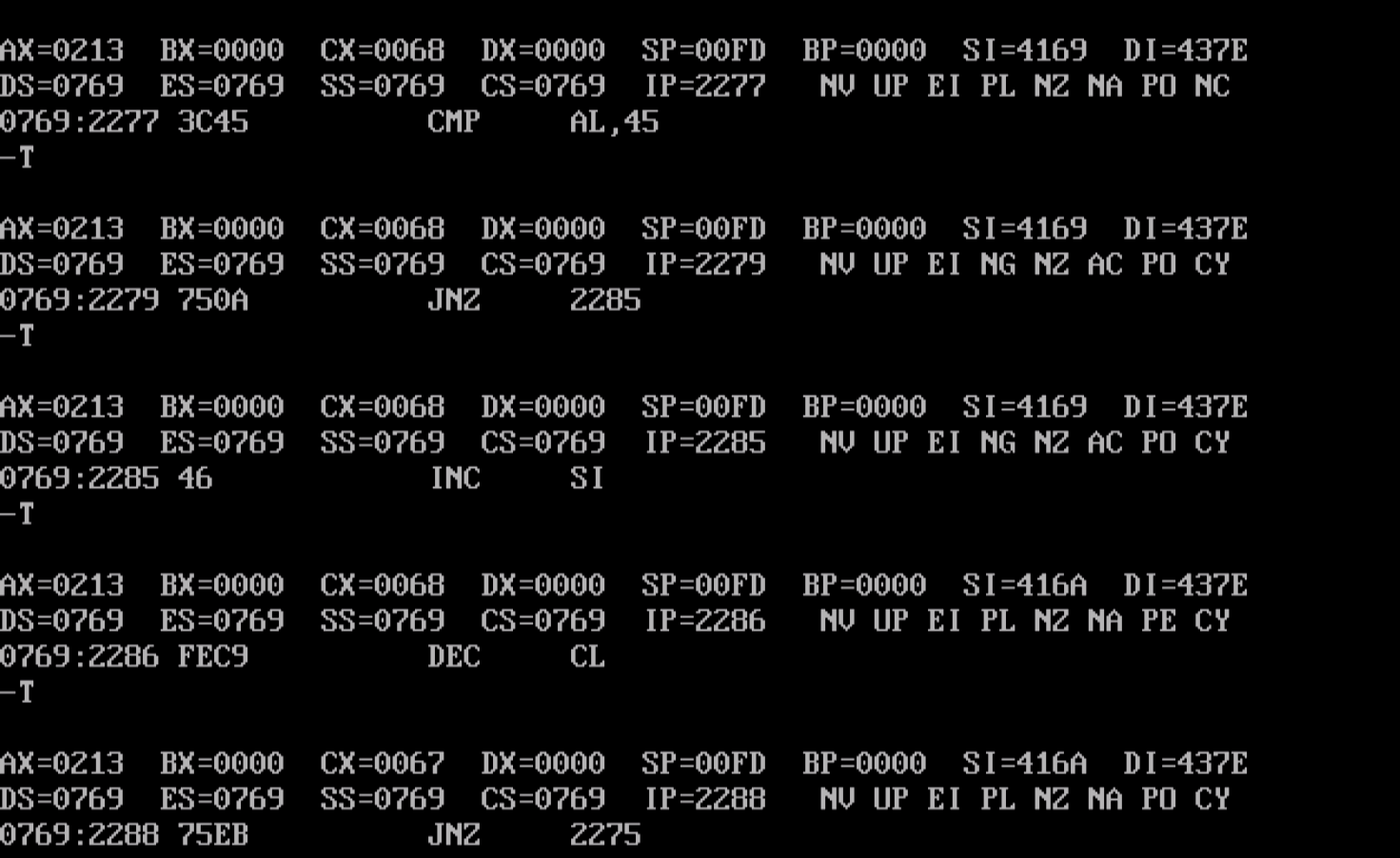


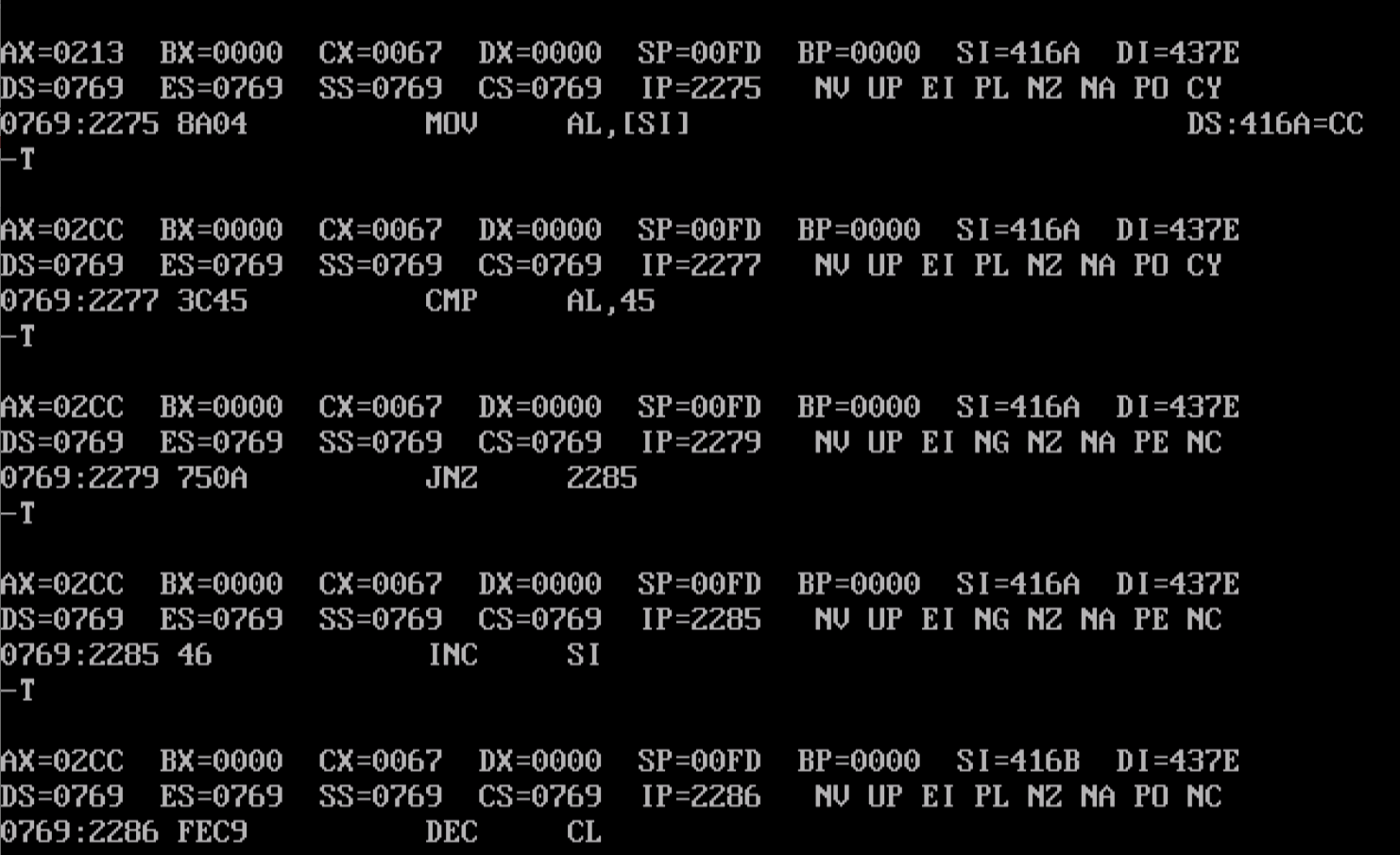












**EJERCICIO 12**

Inicio del programa en la posición de memoria: 2269

Inicio de área de datos en la posición de memoria: 4165

**1. Una descripción de lo que el programa realiza**

Se buscará en una lista de números naturales (de 2 bytes cada uno), cuál es el mayor de ellos

utilizando la instrucción JA (jump of above) saltando de instrucción si está por arriba.

**2. Para qué se usa cada registro:**

**AX**: registro que guardará el número mayor de la lista de números a medida que se la recorra,

hasta que ya no queda más números que recorrer y quedándose aquí el número mayor a todos los

demás.

**SI**: apunta al comienzo de la lista de números

**CL**: es el registro que guardara los n números de la lista

**3. Dónde se ubican las variables y qué valores les has asignado:**

**4165 y 4166:** dirección donde se guarda el primer número de 2 bytes del vector

**4167 y 4168:** dirección donde se guarda el segundo número de 2 bytes del vector

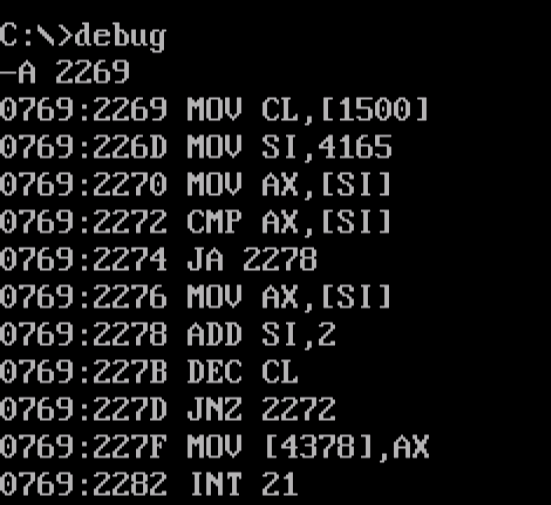
**4169 y 416A:** dirección donde se guarda el tercer número de 2 bytes del vector

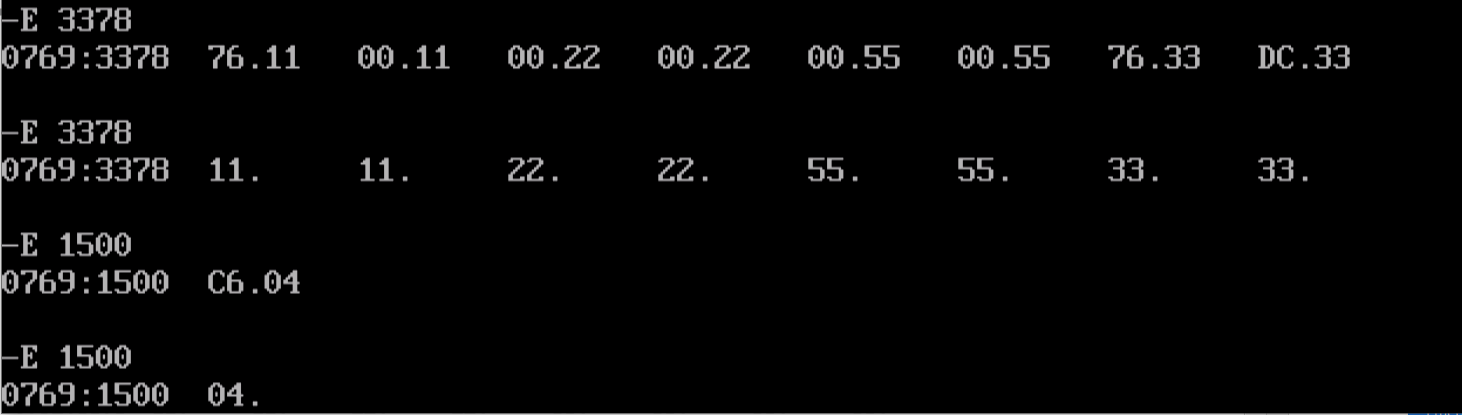
**416B y 416C:** dirección donde se guarda el cuarto número de 2 bytes del vector

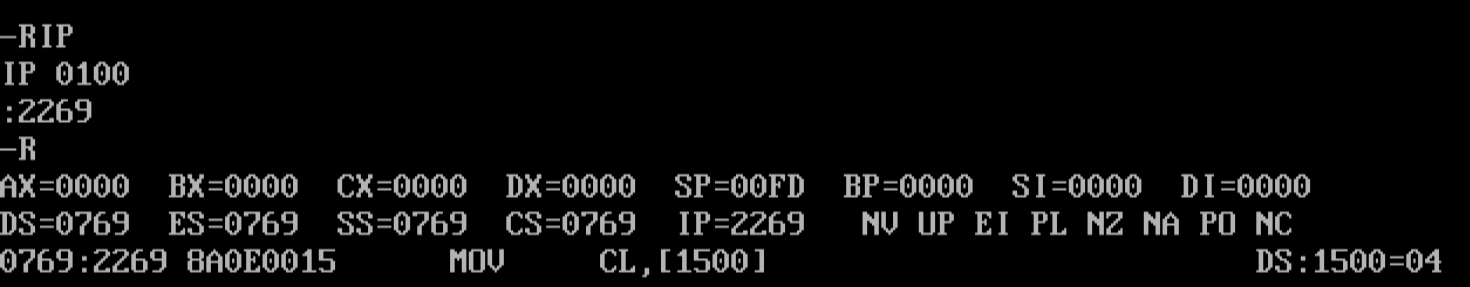
**1500:** n elementos del vector, la cantidad de números que tendrá el vector

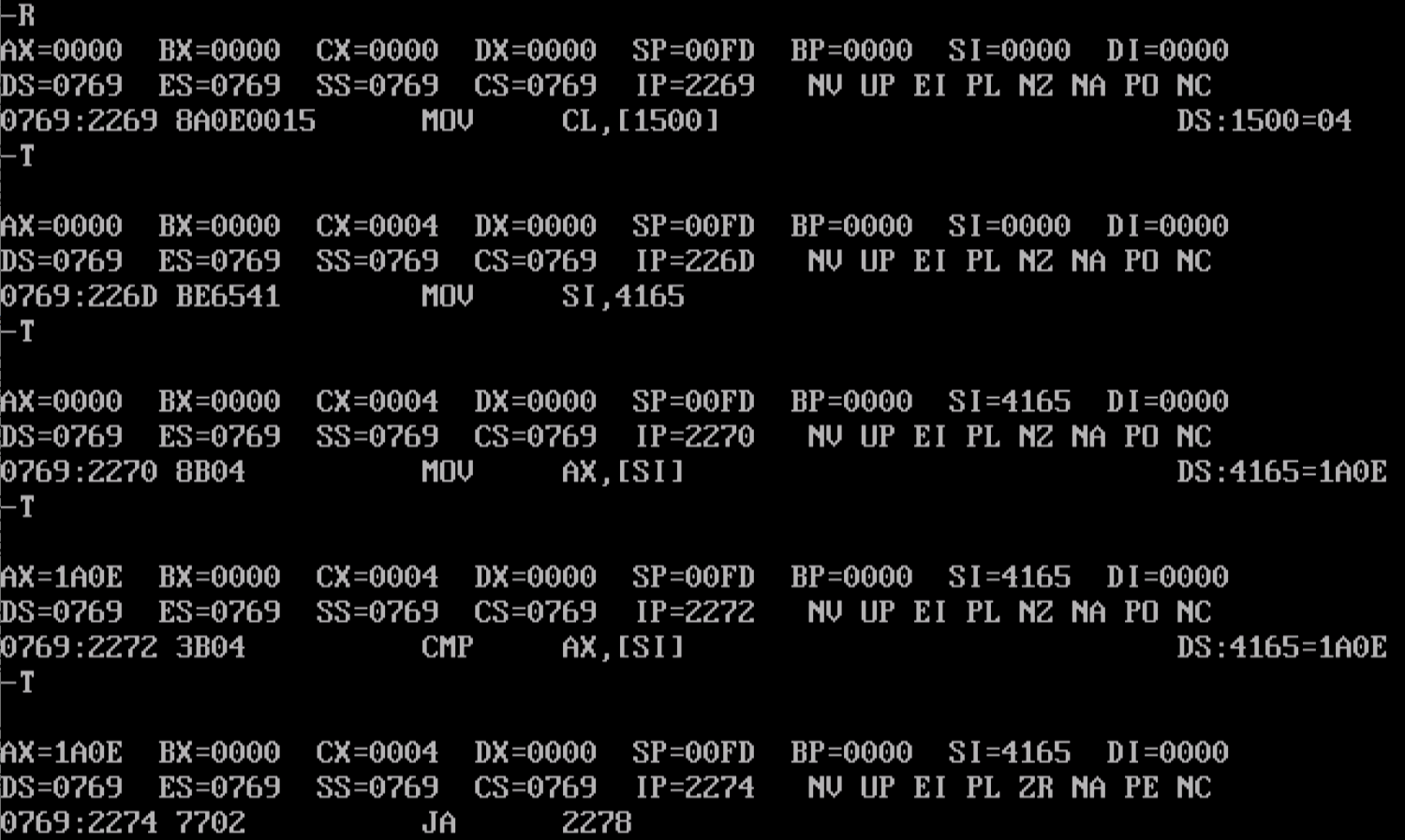
**4378:** dirección donde se guardará el número mayor de la lista cuando se haya terminado de recorrerla

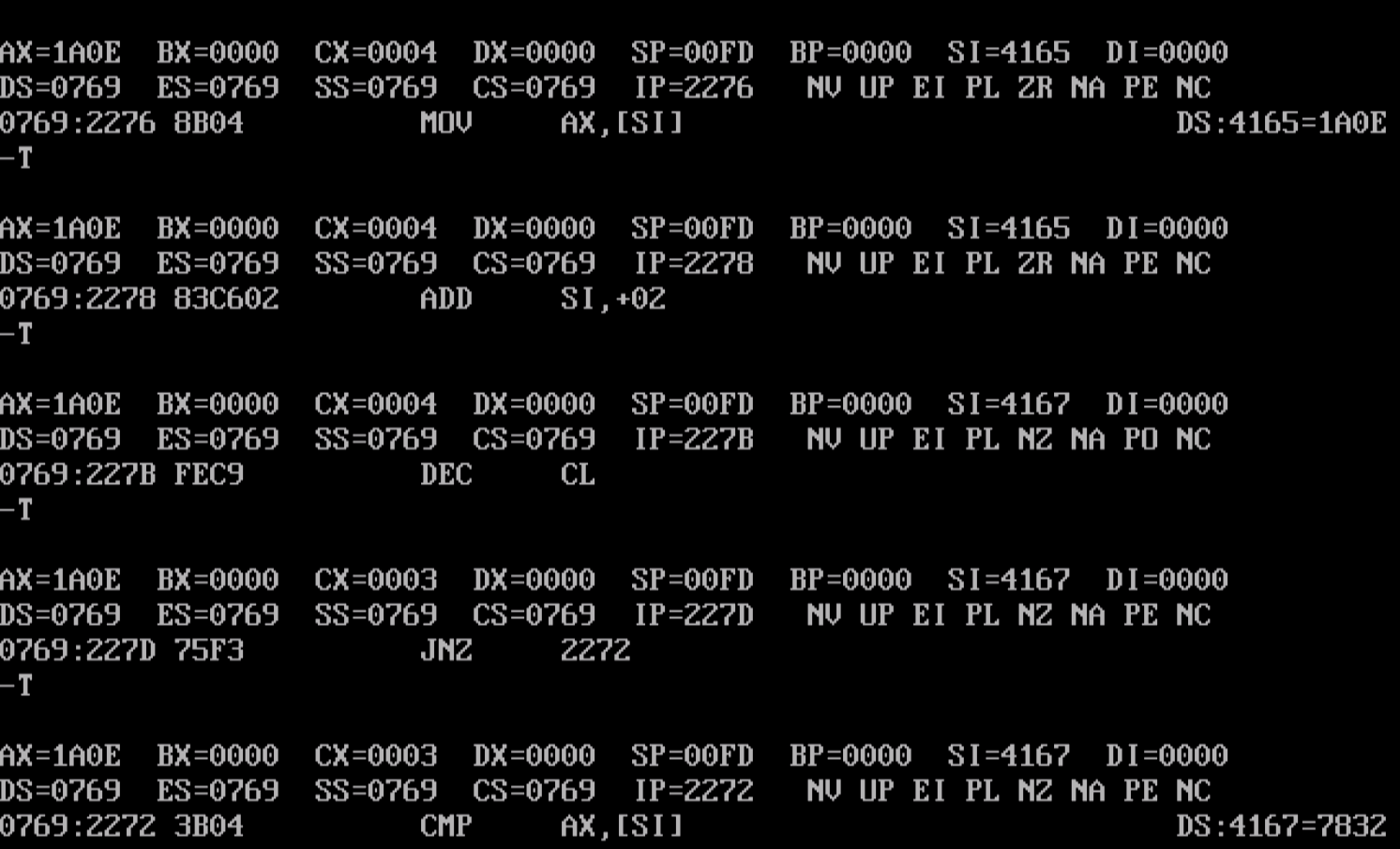
**CAPTURAS DOSBOX**

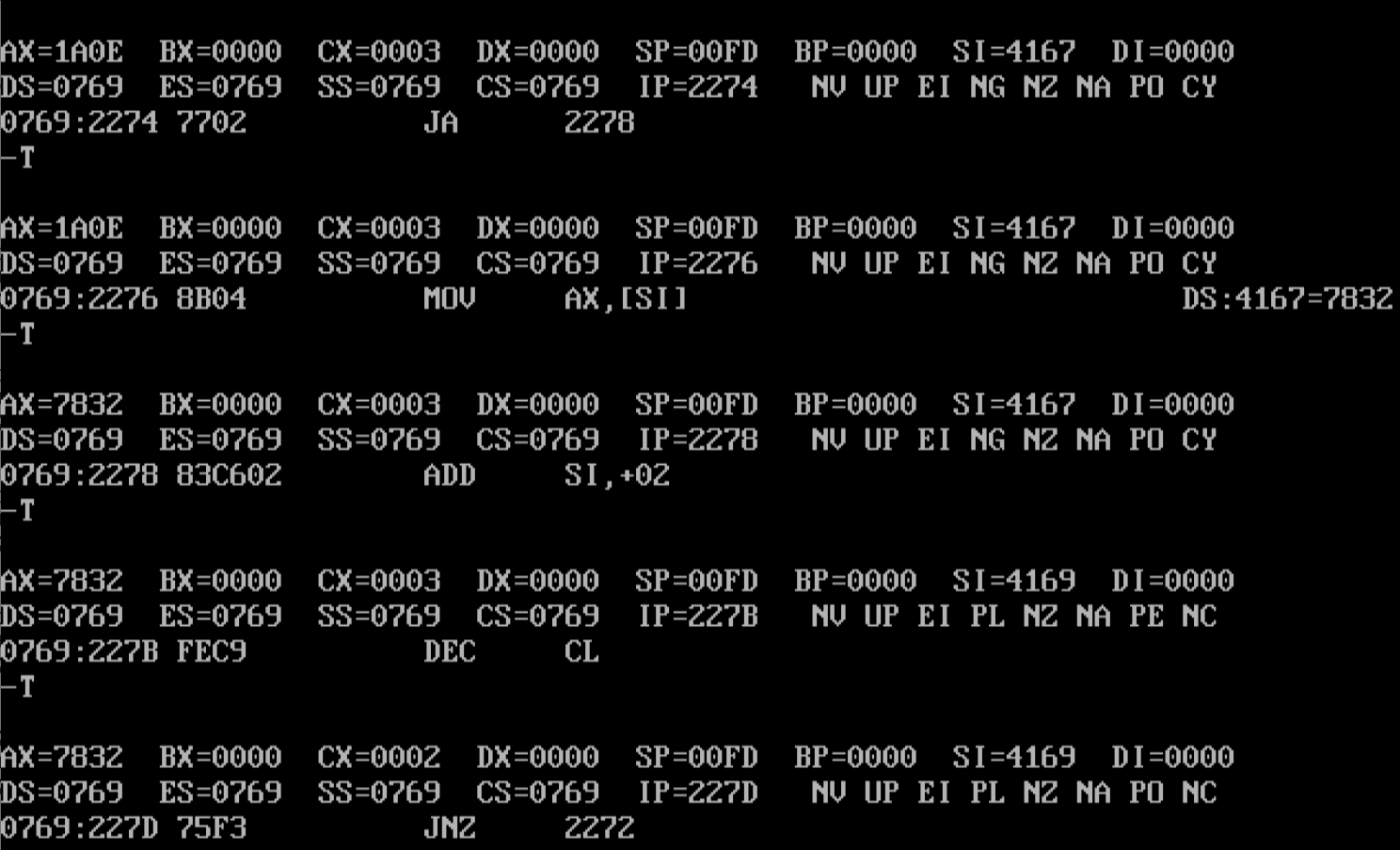
****

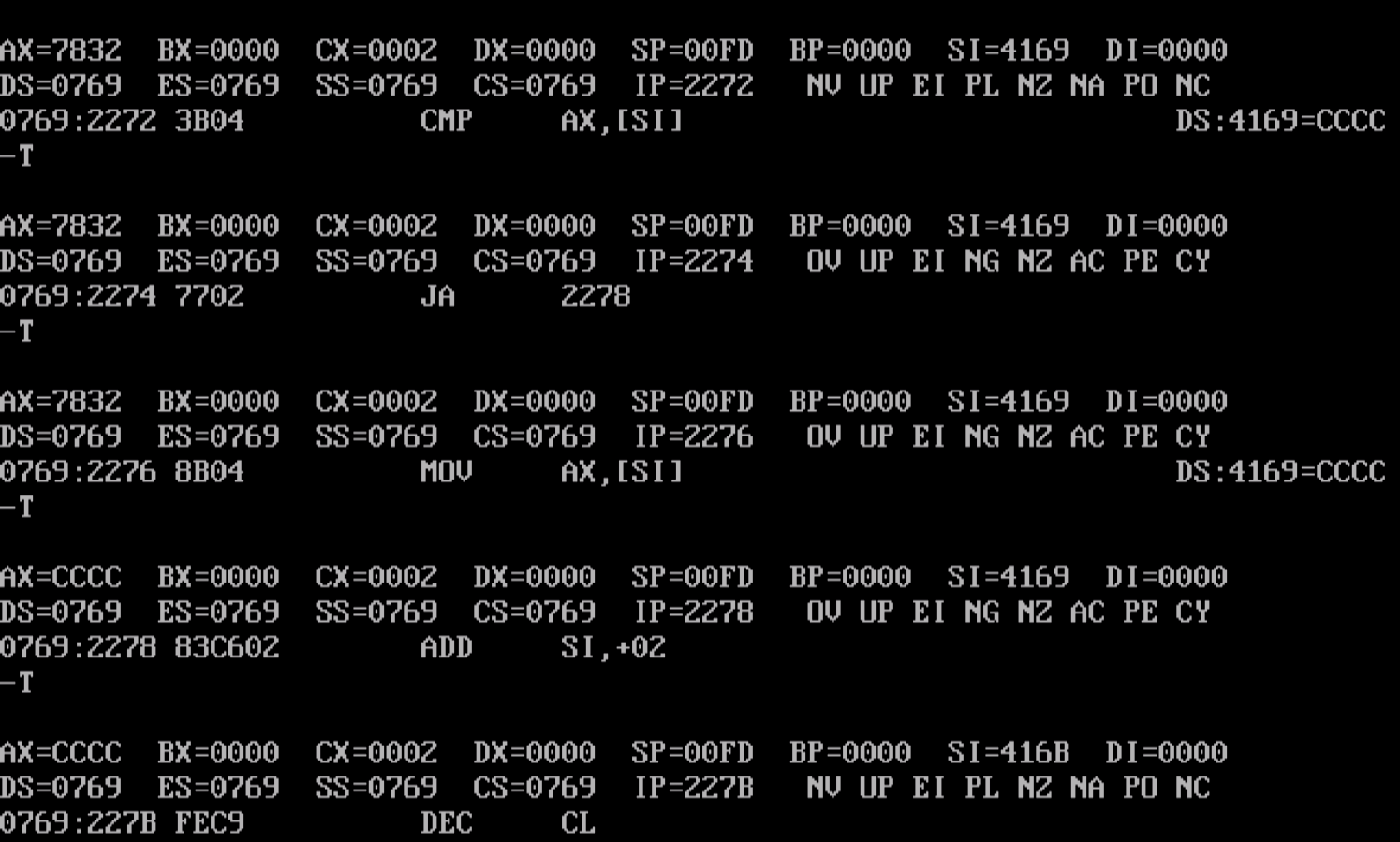
****

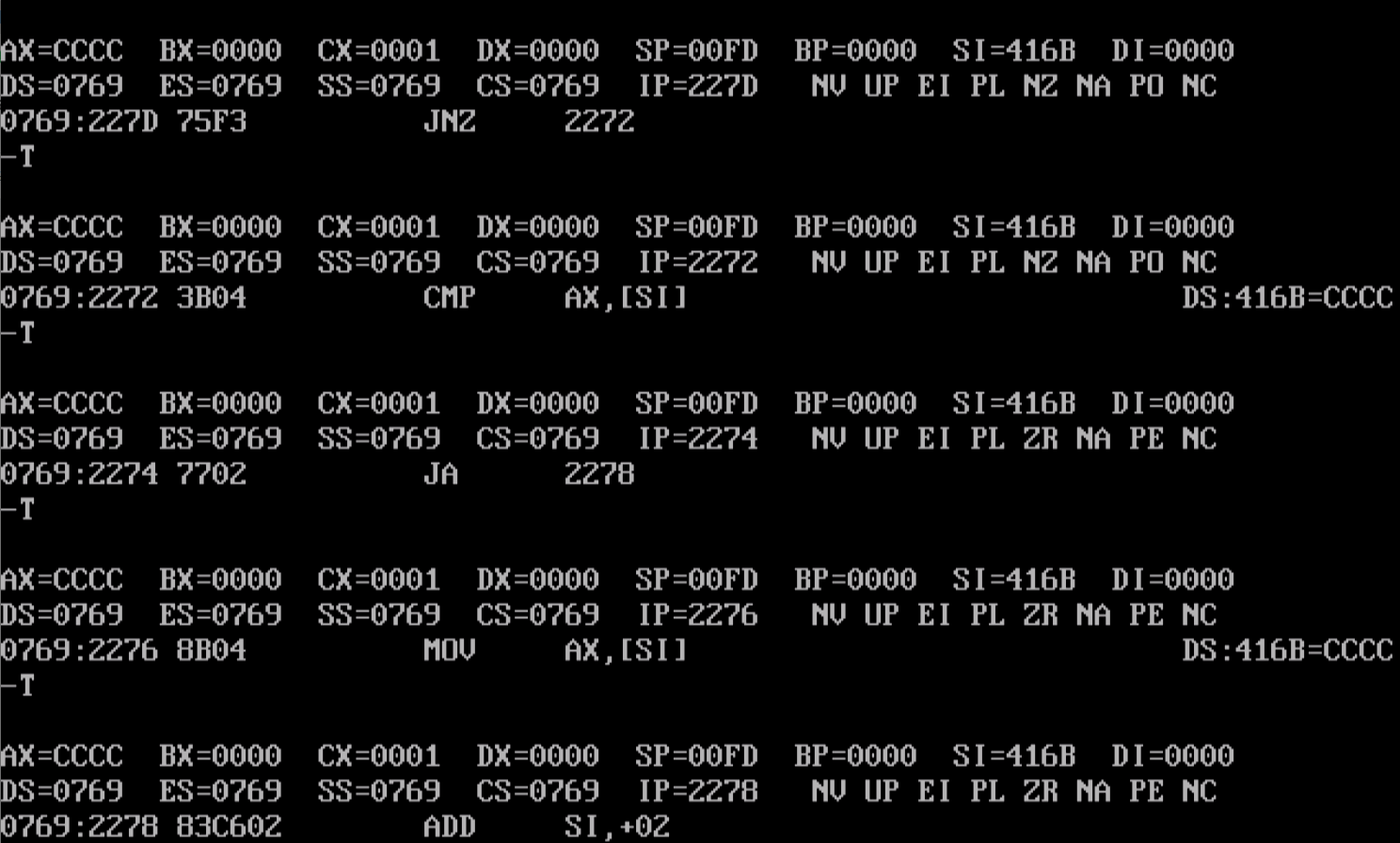
****

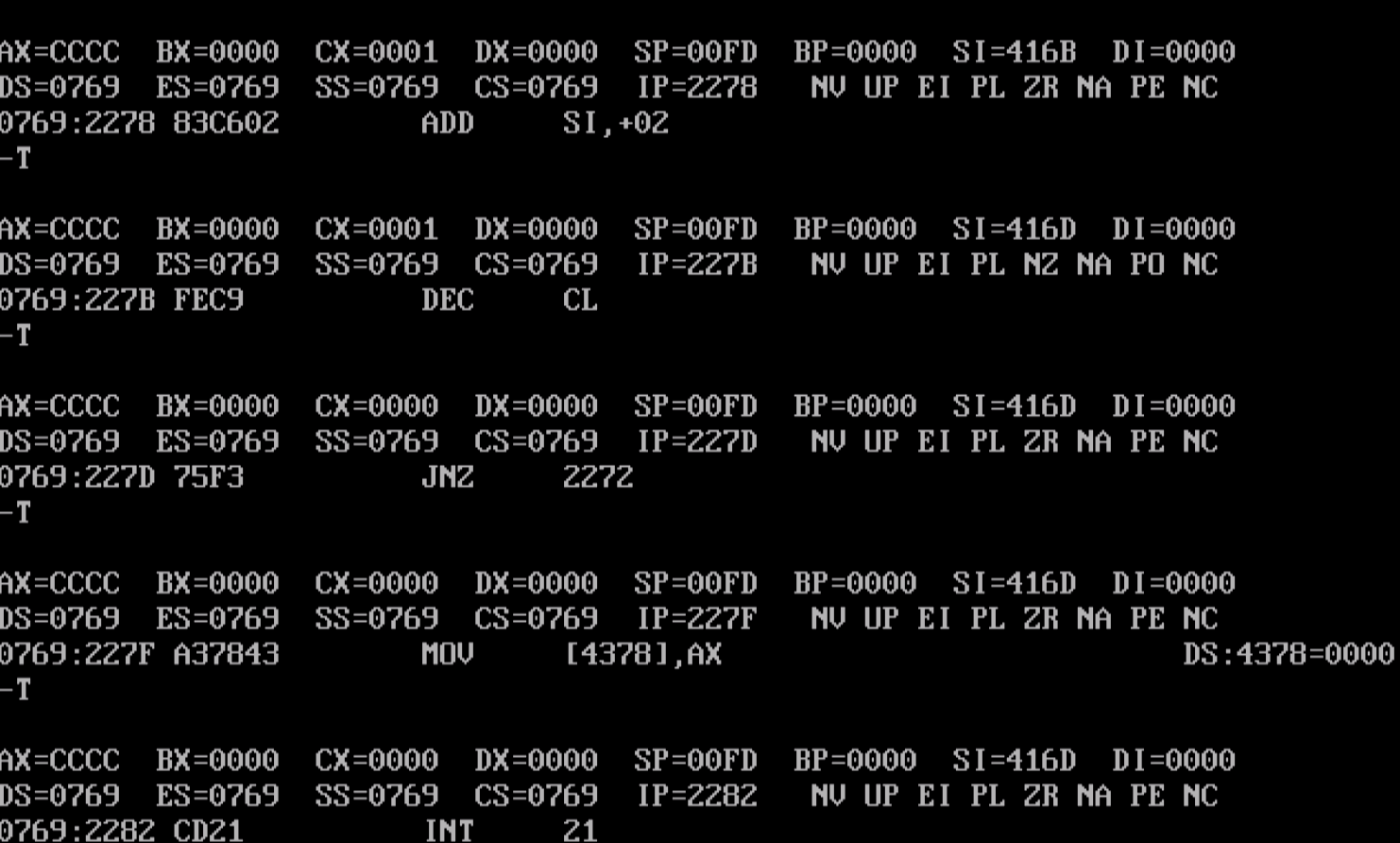
****

****

****

****

****

****

****

**EJERCICIO 12**

Inicio del programa en la posición de memoria: 2269

Inicio de área de datos en la posición de memoria: 4165

**1. Una descripción de lo que el programa realiza**

El programa separara en dos listas, los números positivos y negativos (los ceros no se cuentan)

enteros pertenecientes a una lista principal y guardara la cantidad hallada de ambos signos en dos

variables.

**2. Para qué se usa cada registro:**

**CL**: es el registro que guardara los n números de la lista

**SI**: apunta al comienzo de la lista principal de números

**DI**: apunta al comienzo de la lista de números negativos

**BX**: apunta al comienzo de la lista de números positivos

**DX**: será el registro que lleve a cabo el contador del total de números positivos (DH) y negativos

(**DL**) hallados en cada lista respectivamente.

**AX**: es el registro donde se guardarán los números de la lista principal para ser comparados.

**3. Dónde se ubican las variables y qué valores les has asignado:**

**4165** y **4166**: dirección donde se guarda un número positivo (0025)

**4167** y **4168**: dirección donde se guarda un cero

**4169** y **416A**: dirección donde se guarda un número negativo (8033)

**416B** y **416C**: dirección donde se guarda un número positivo (0042)

**1500**: n elementos del vector, la cantidad de números que tendrá el vector

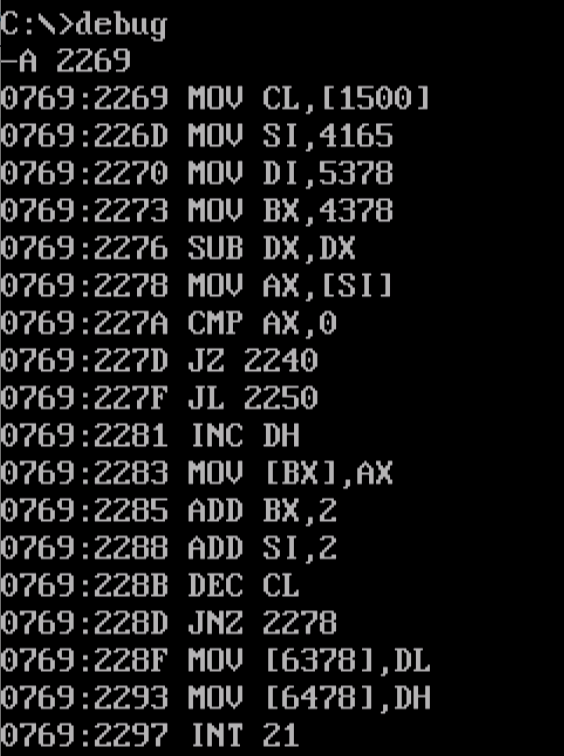
**4378**: dirección donde se guardará el primer elemento de la lista de número positivos

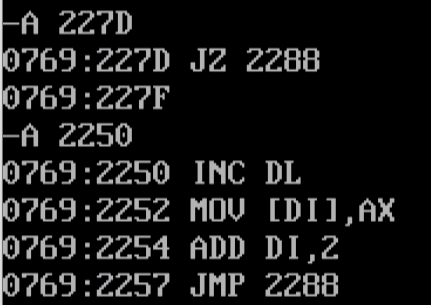
**5378**: dirección donde se guardará el primer elemento de la lista de número negativos

**6378**: dirección donde se guardará el total de negativos encontrados

**6478**: dirección donde se guardará el total de positivos encontrados

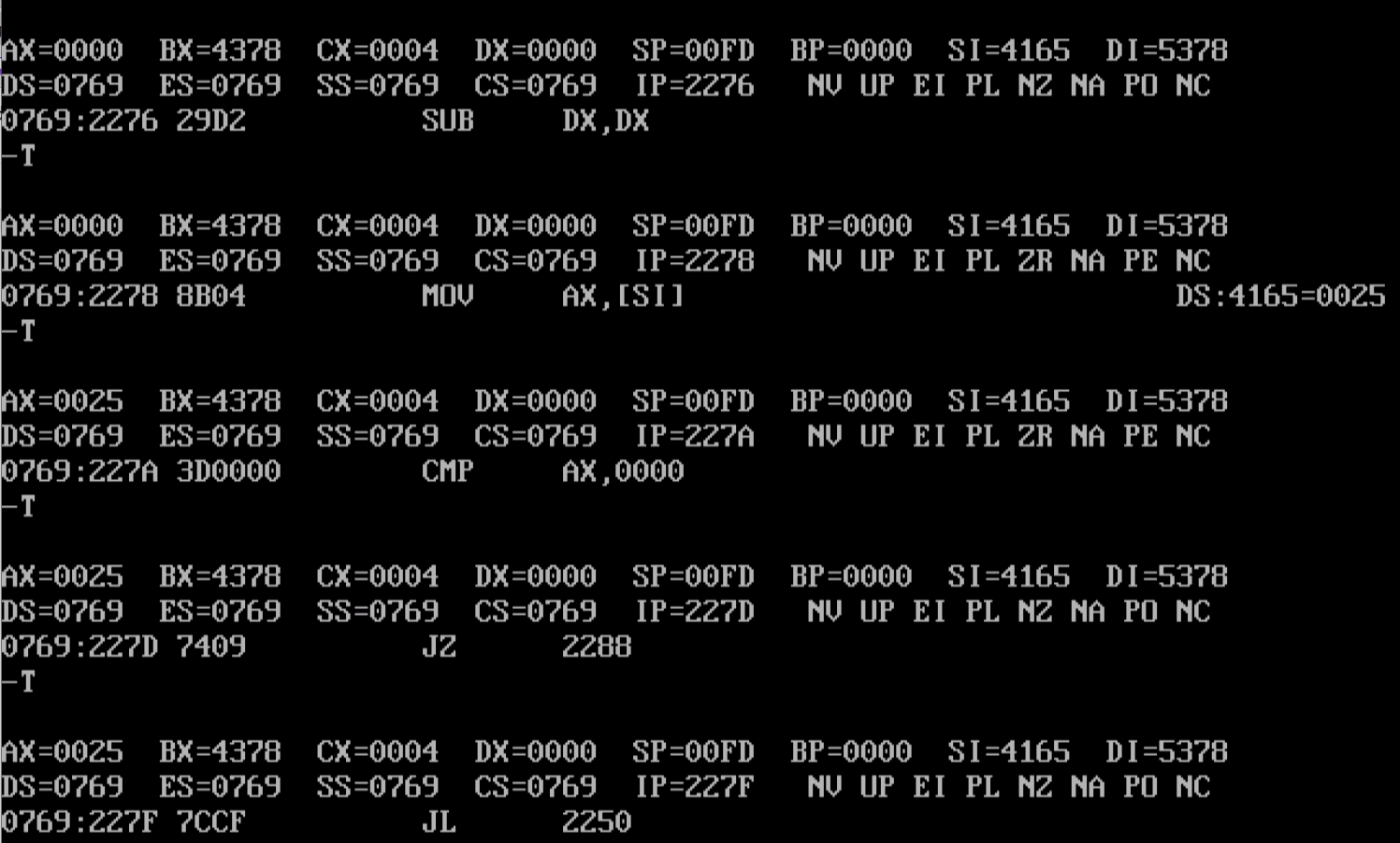
**CAPTURAS DOSBOX**

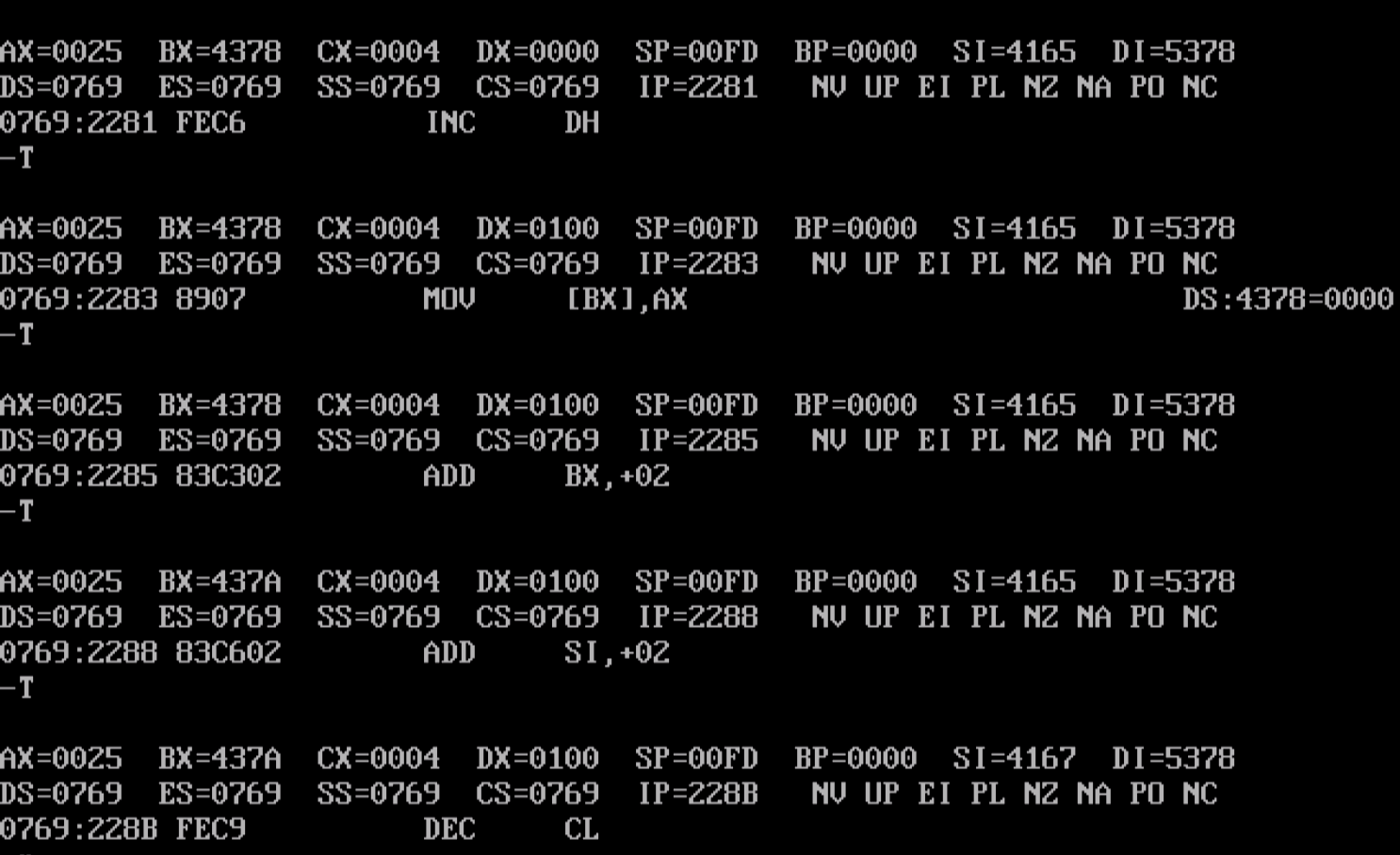


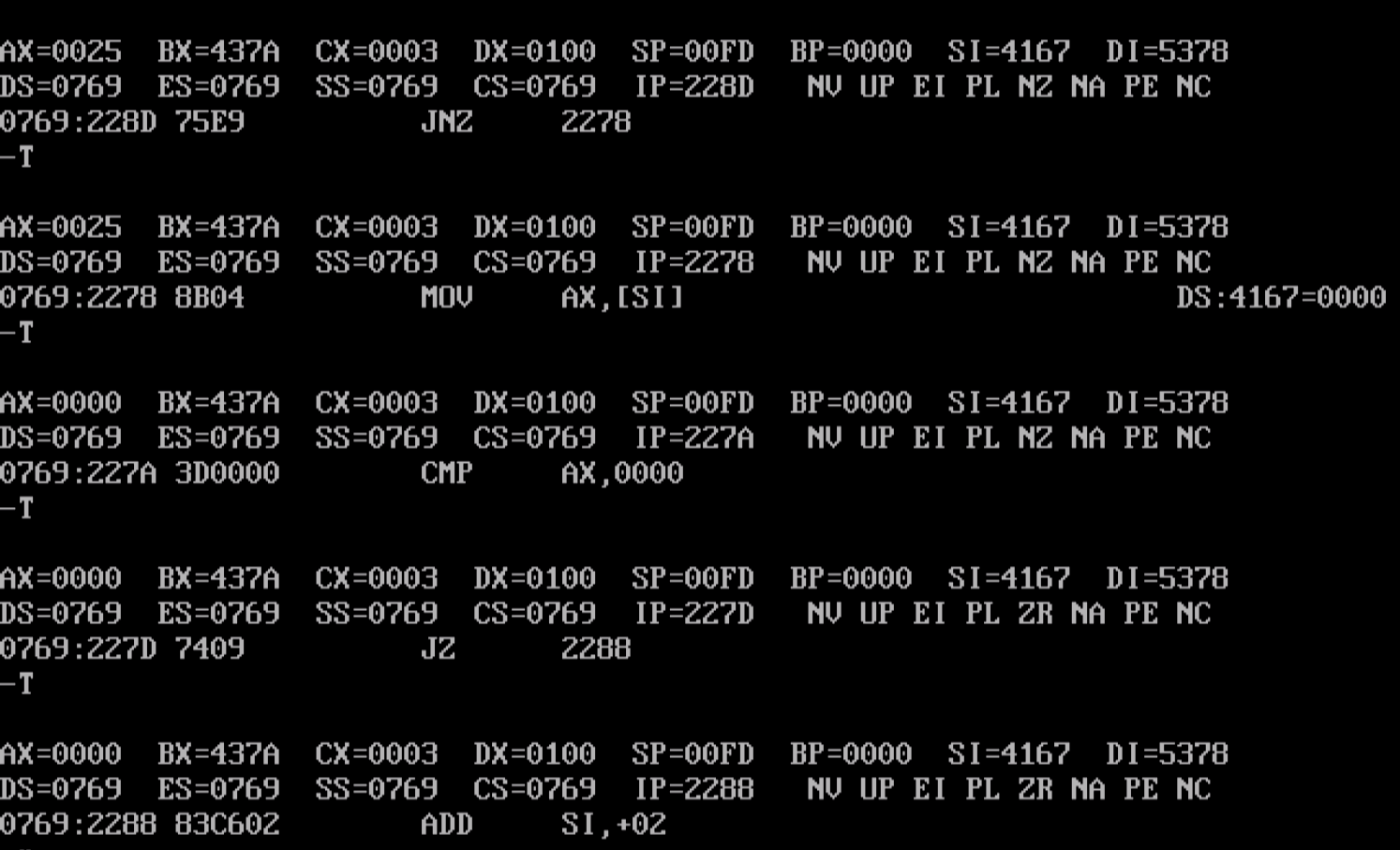


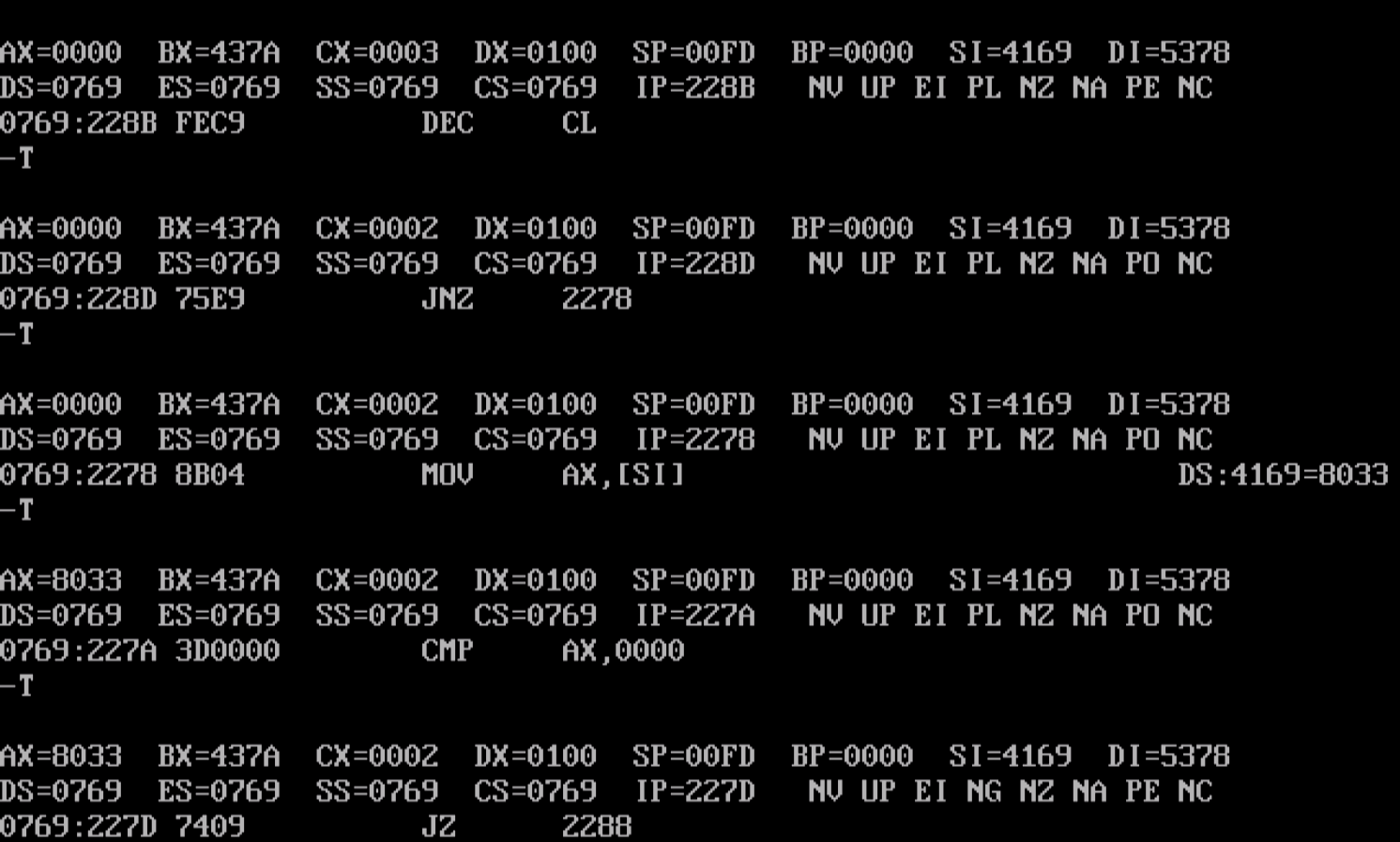




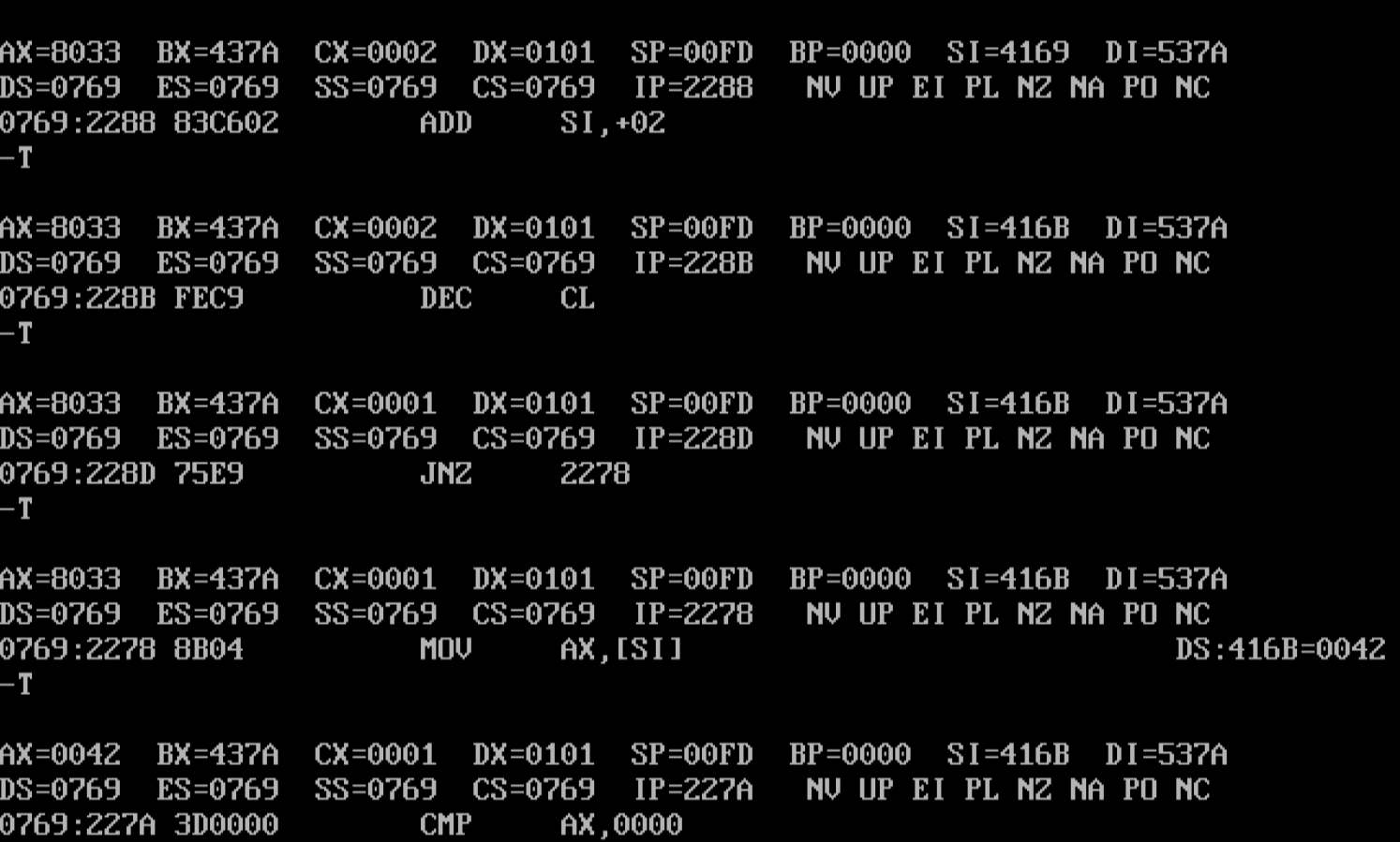




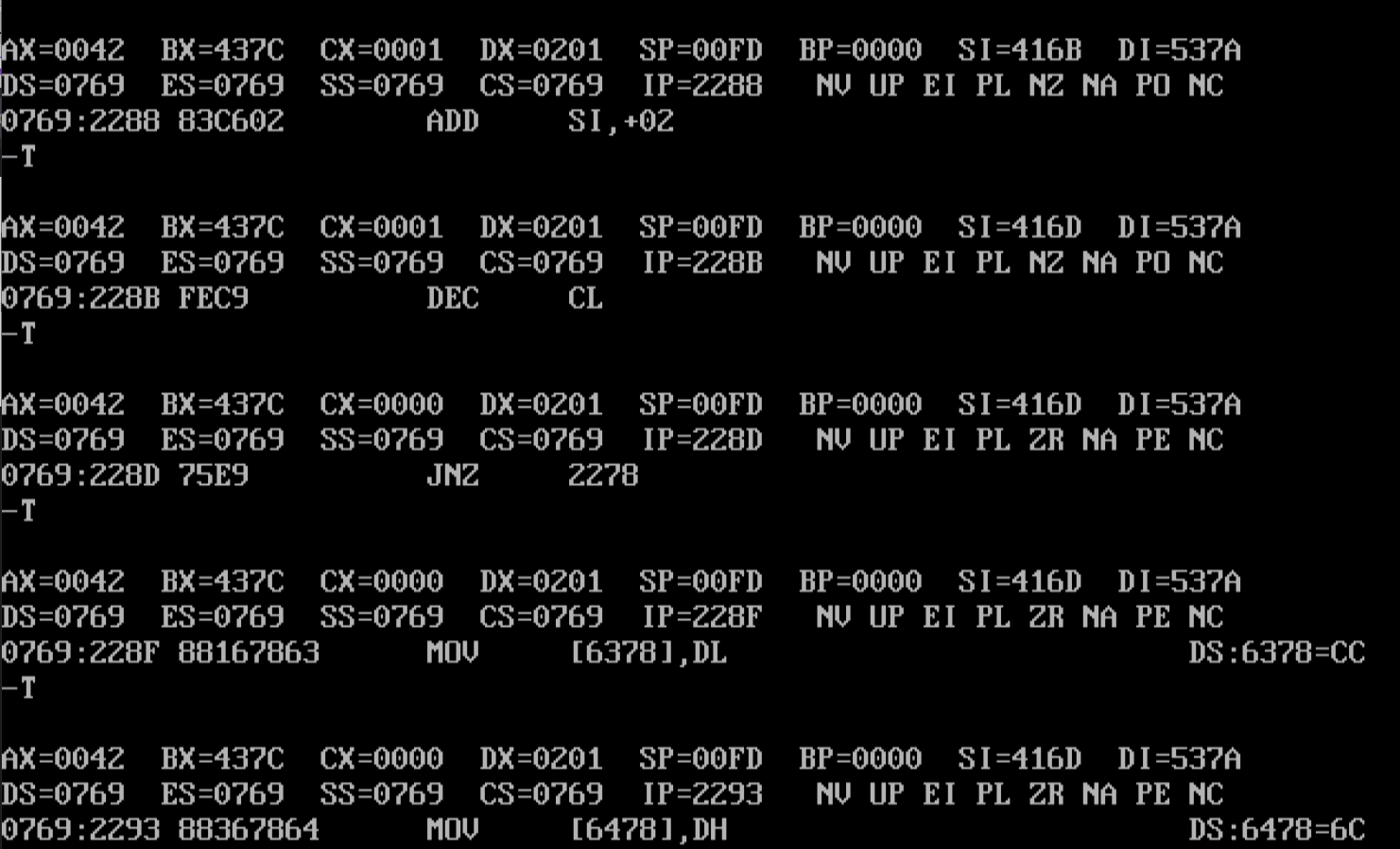


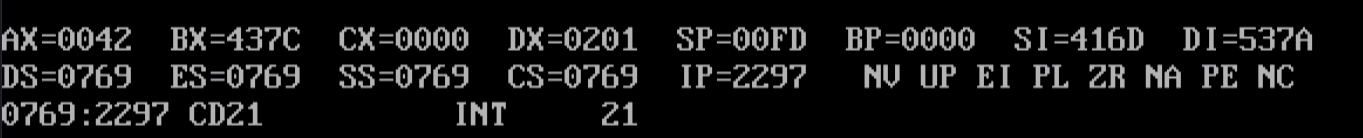


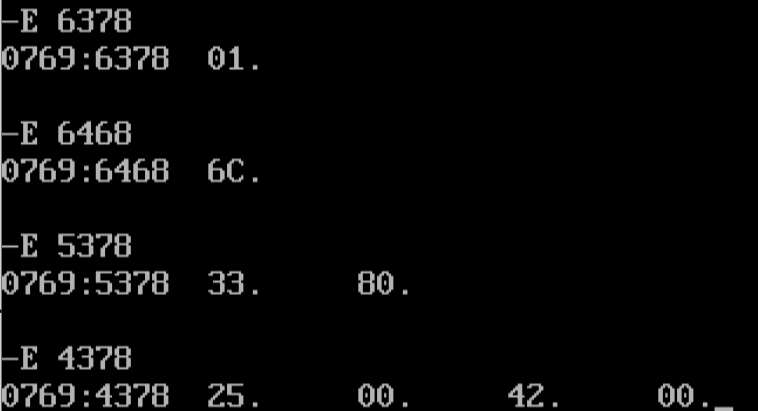












**EJERCICIO 14**

Inicio del programa en la posición de memoria: 2269

Inicio de área de datos en la posición de memoria: 4165

**1. Una descripción de lo que el programa realiza**

En este programa se tendrá dos listas conformadas por caracteres en código ASCII de mayúsculas

correspondientes a nombres de personas, siendo de igual o distinta longitud, donde cada lista terminará con el carácter ASCII correspondiente al Enter. Ambas listas se las compara y se determinará cuál de ellas va primero por orden alfabético, escribiendo en la memoria la dirección de dicha lista. Mientras que no haya el código ASCII del enter en la comparación, se seguirá con el proceso.

**2. Para qué se usa cada registro:**

**BX**: es el registro que apunta al comienzo de la primera lista de caracteres. Se usará para indicar al final del programa que esta lista es la que corresponde primera por orden alfabético, en caso que así sea.

**DX**: es el registro que apunta al comienzo de la segunda lista de caracteres. Se usará para indicar al final del programa que esta lista es la que corresponde primera por orden alfabético, en caso que así sea.

**SI**: es el registro que se inicializa con un valor anterior a la dirección de memoria donde está la primera lista de caracteres.

**DI**: es el registro que se inicializa con un valor anterior a la dirección de memoria donde está la segunda lista de caracteres.

**AL**: registro que recibirá carácter por carácter de la primera lista y se usara para comparar y

determinar si es mayor, menor o igual al código Enter y posteriormente, al carácter de la segunda lista.

**AH**: registro que recibirá carácter por carácter de la segunda lista y se usara para comparar y determinar si es mayor, menor o igual al código Enter y posteriormente, al carácter de la primera lista.

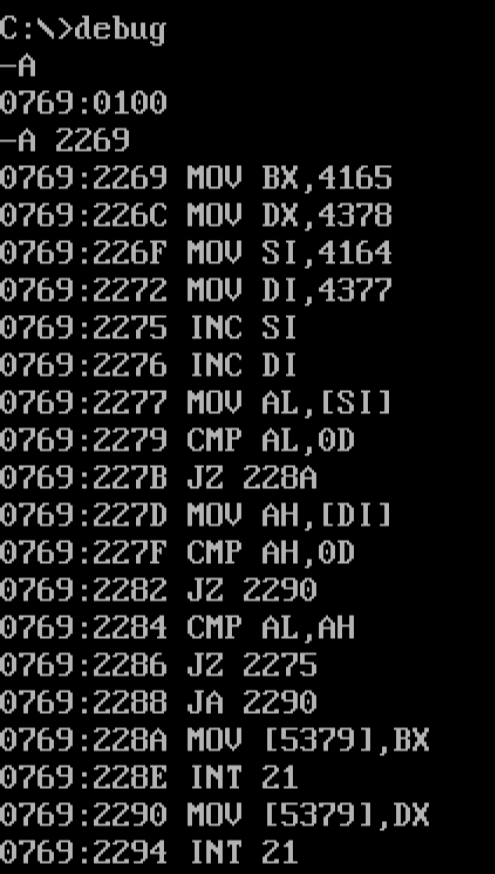
**3. Dónde se ubican las variables y qué valores les has asignado:**

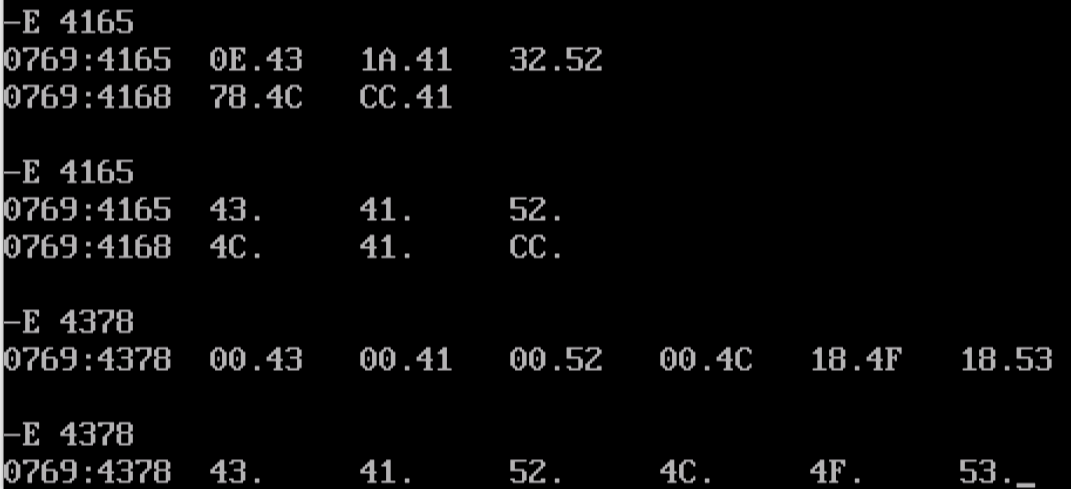
**4165**: dirección donde comenzará el primer elemento de la primera lista

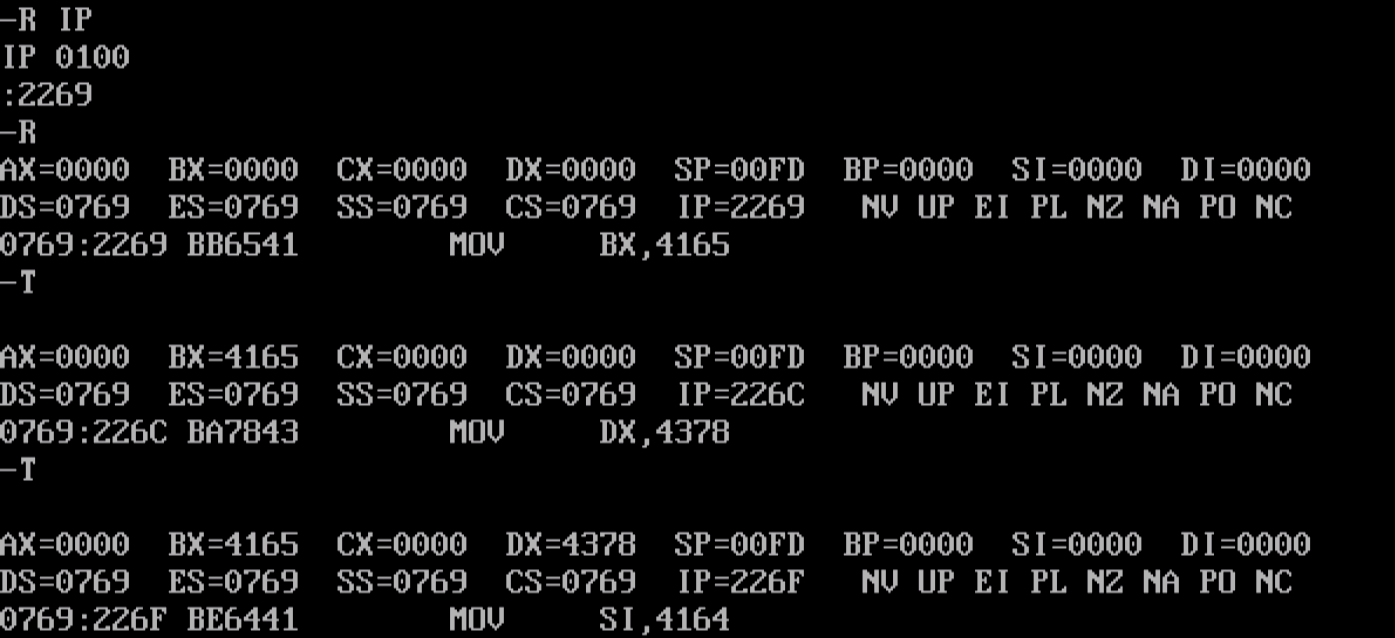
**4378**: dirección donde comenzará el primer elemento de la segunda lista

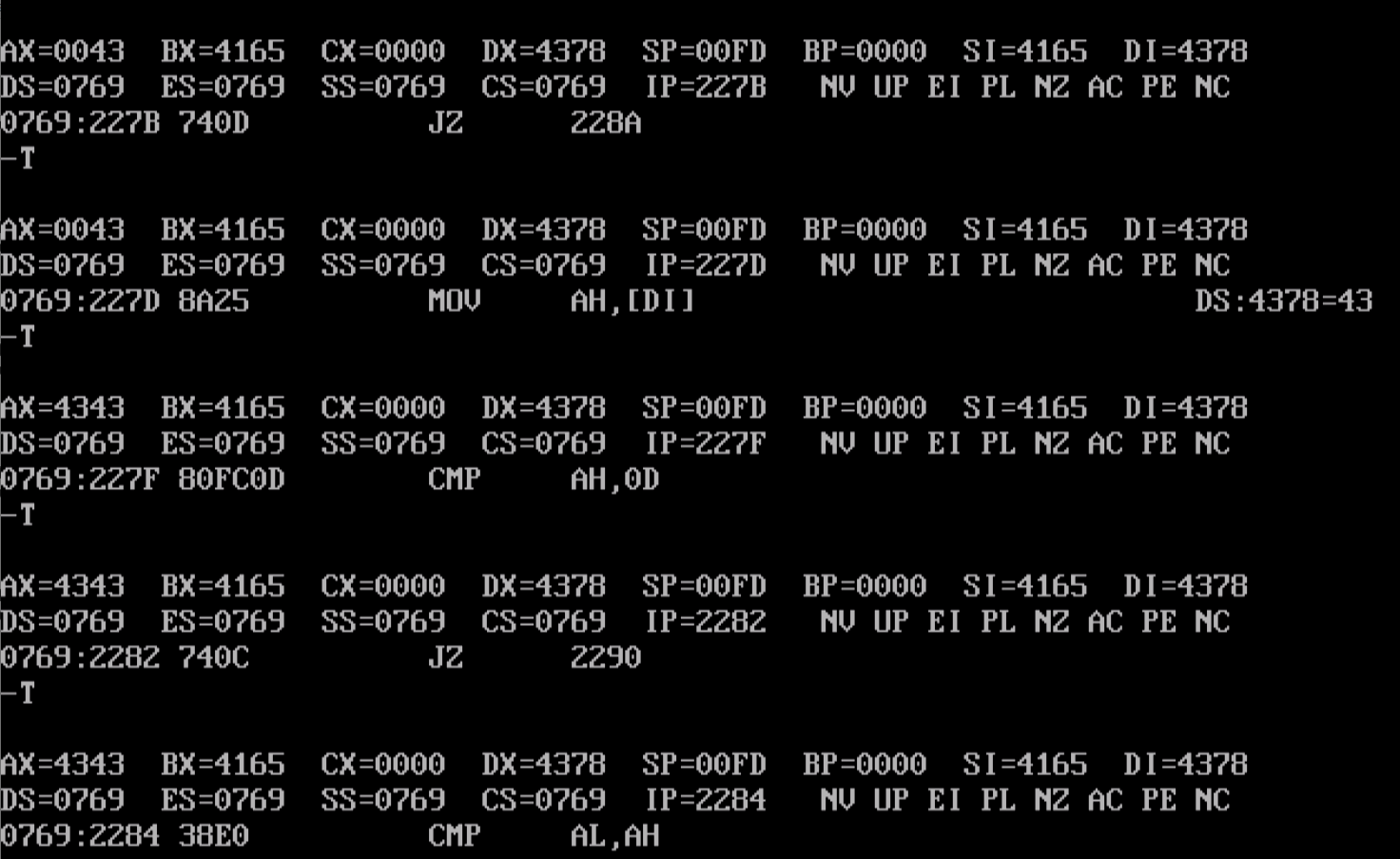
**5379**: dirección donde comenzará el primer elemento de la lista resultante tras la comparación.

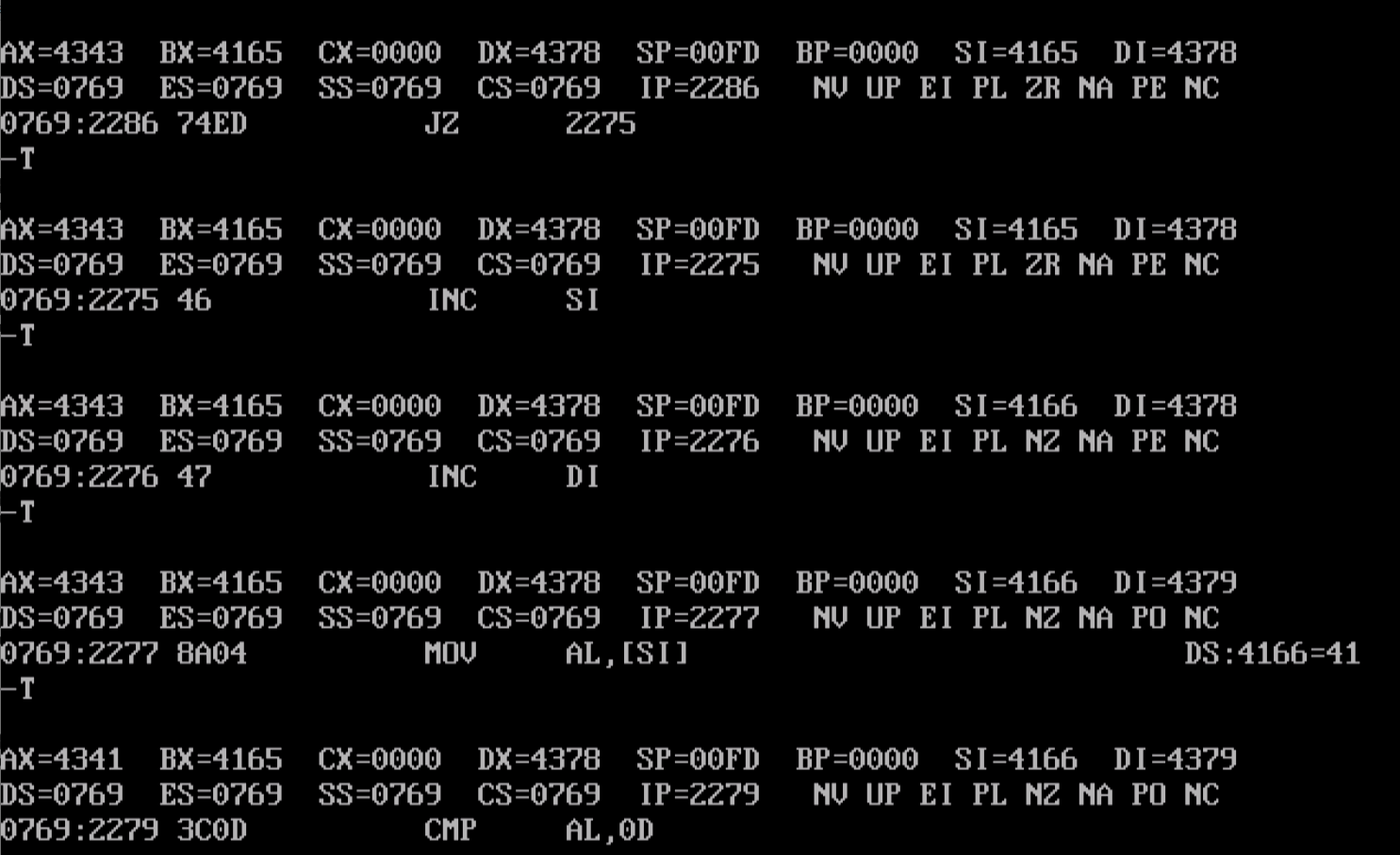
**CAPTURAS DOSBOX**

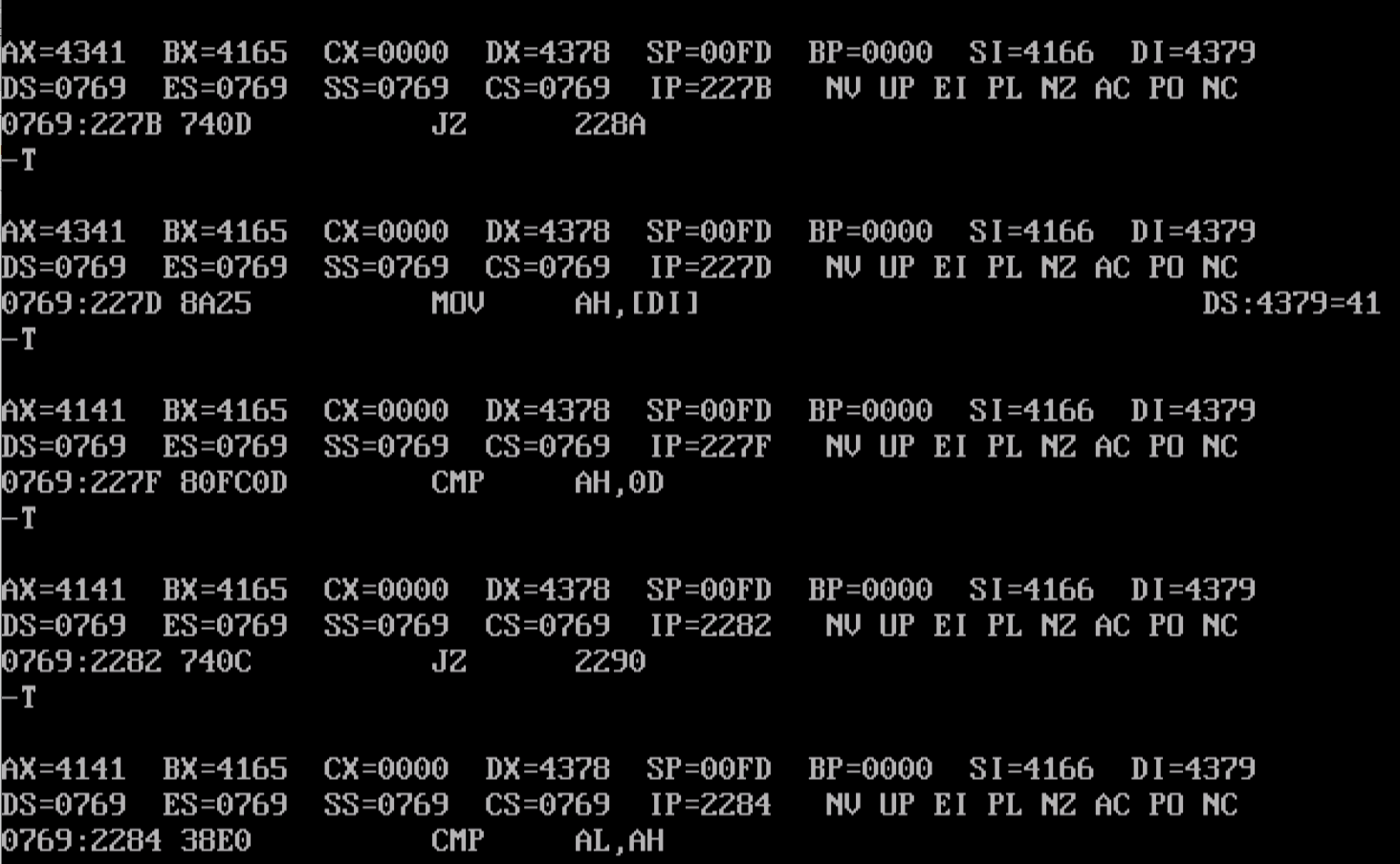
****

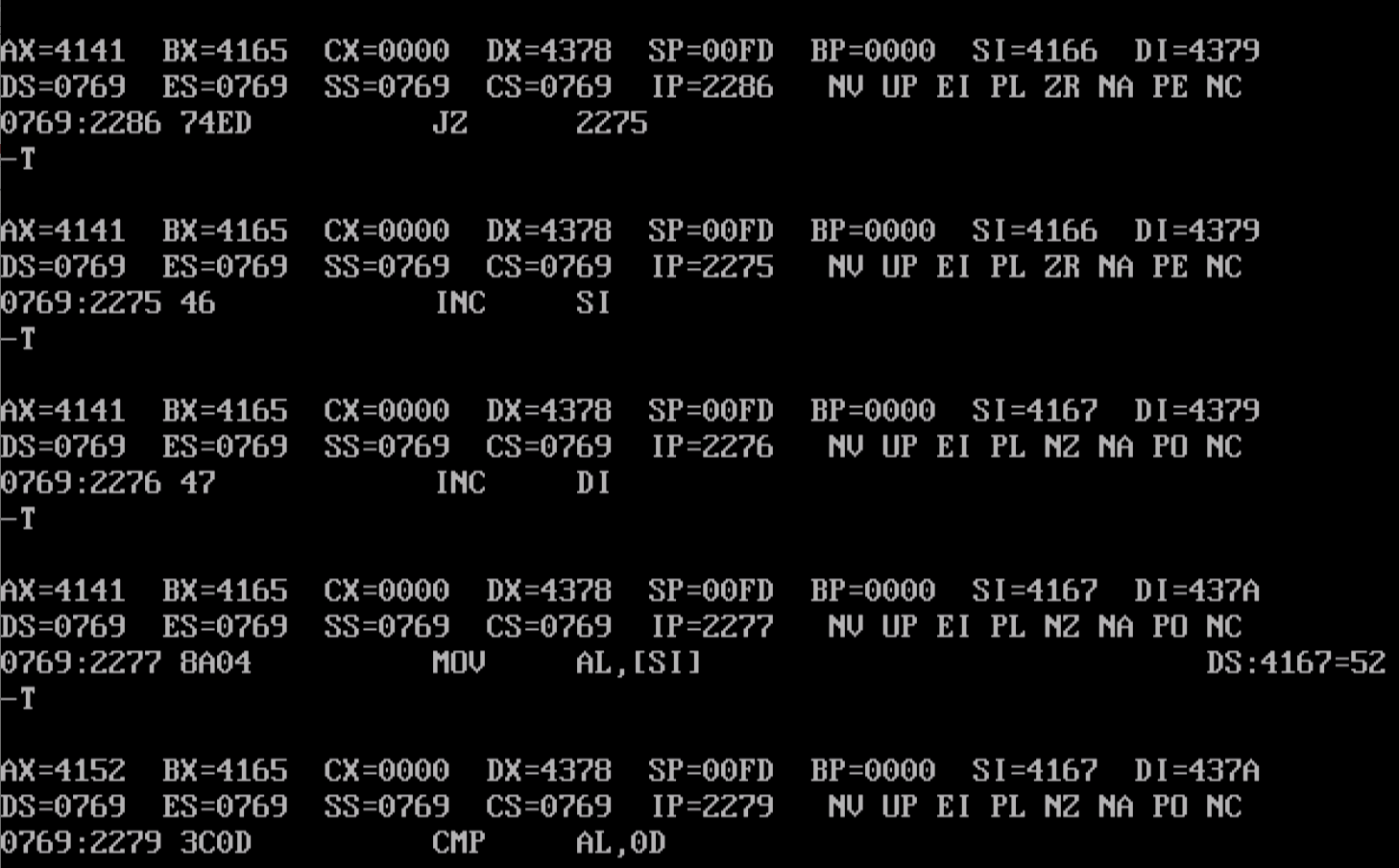
****

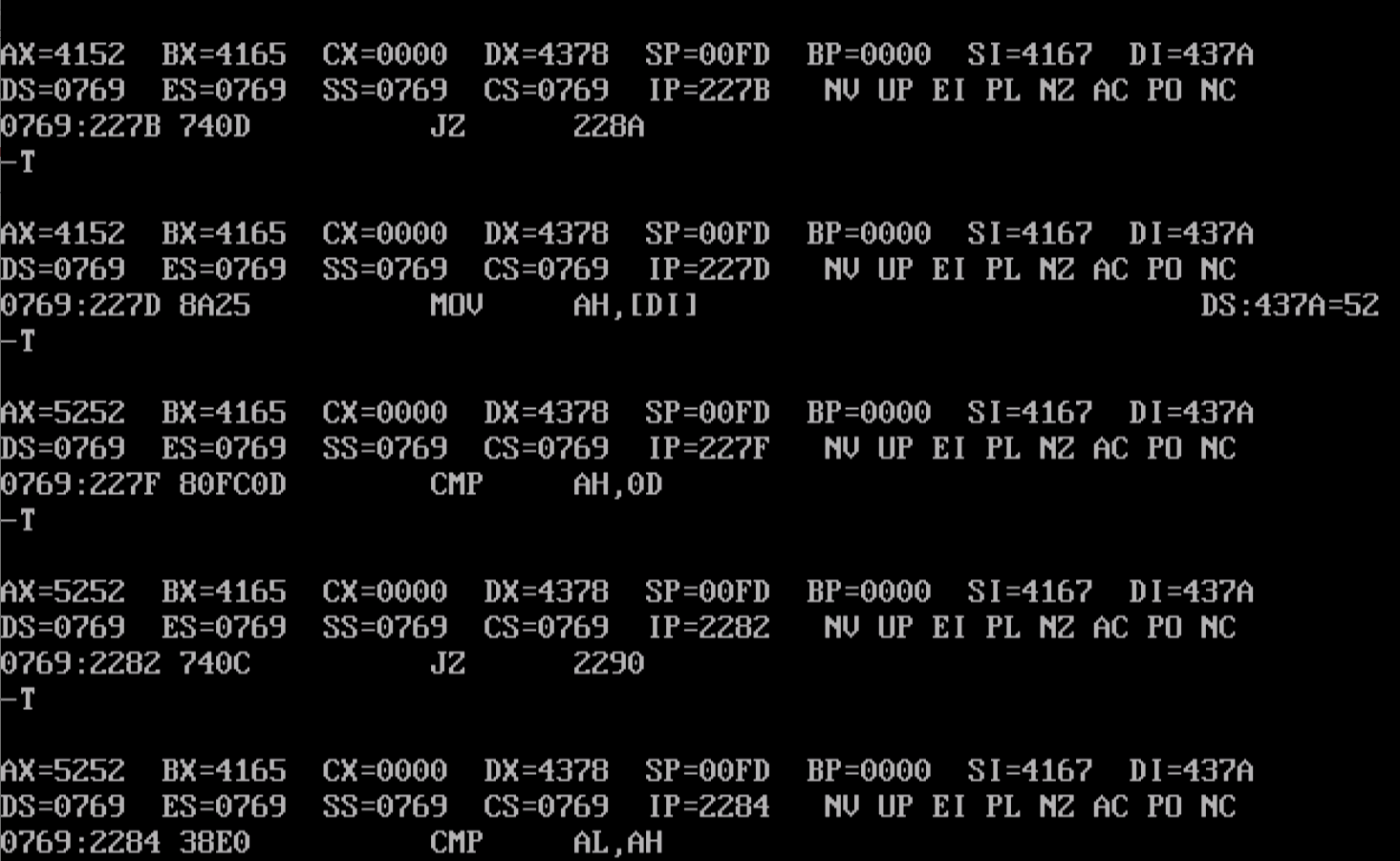
****

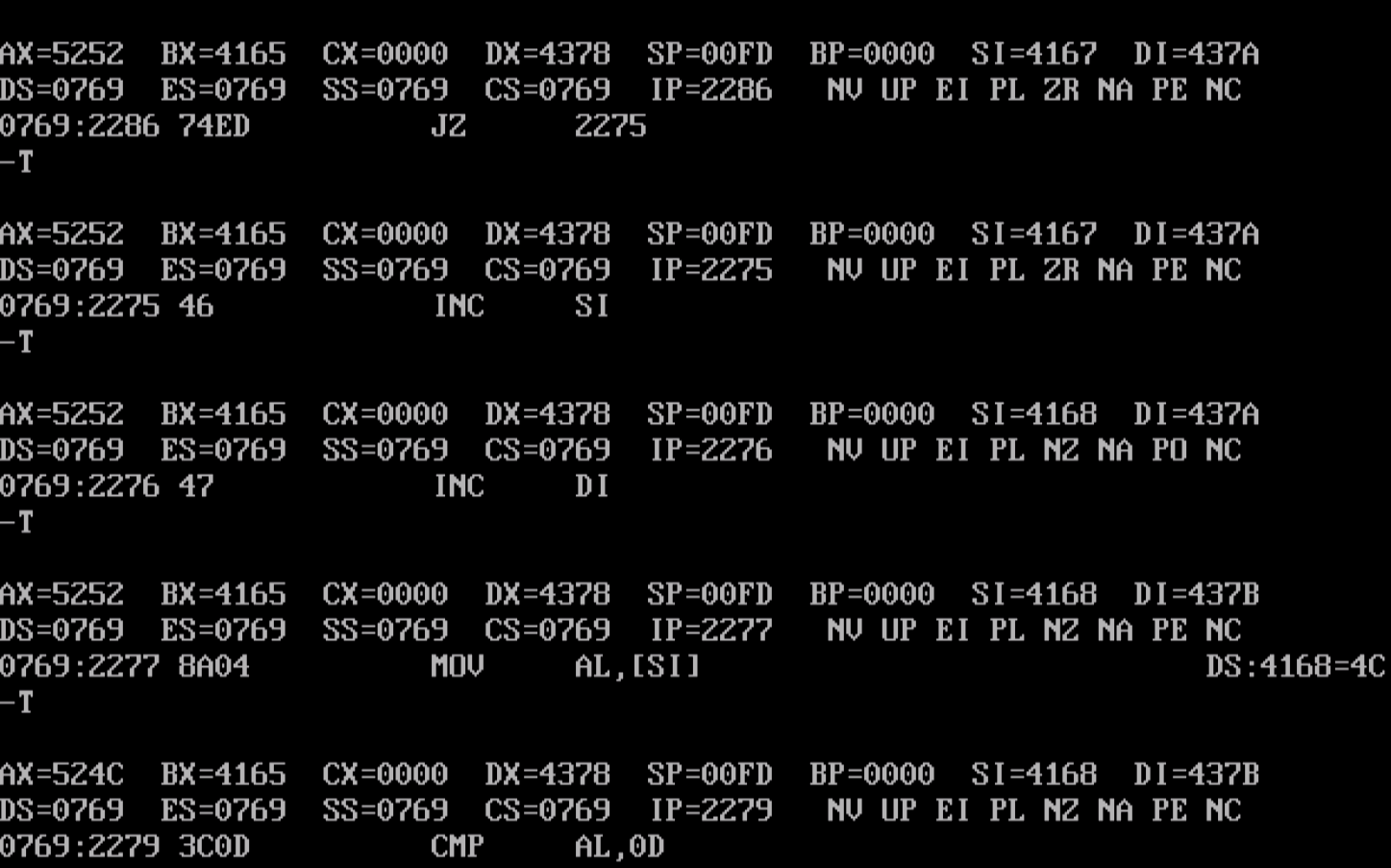
****

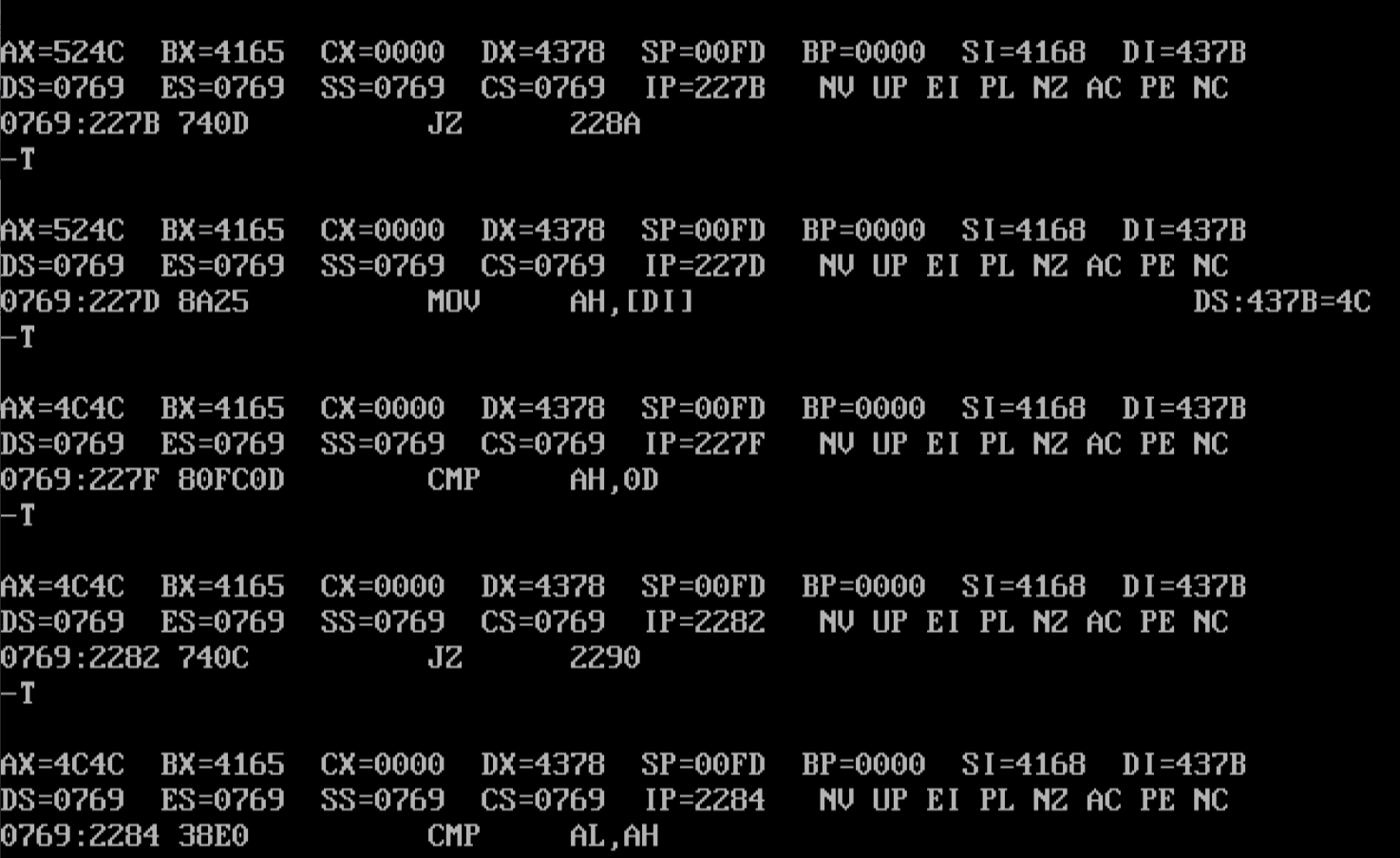
****

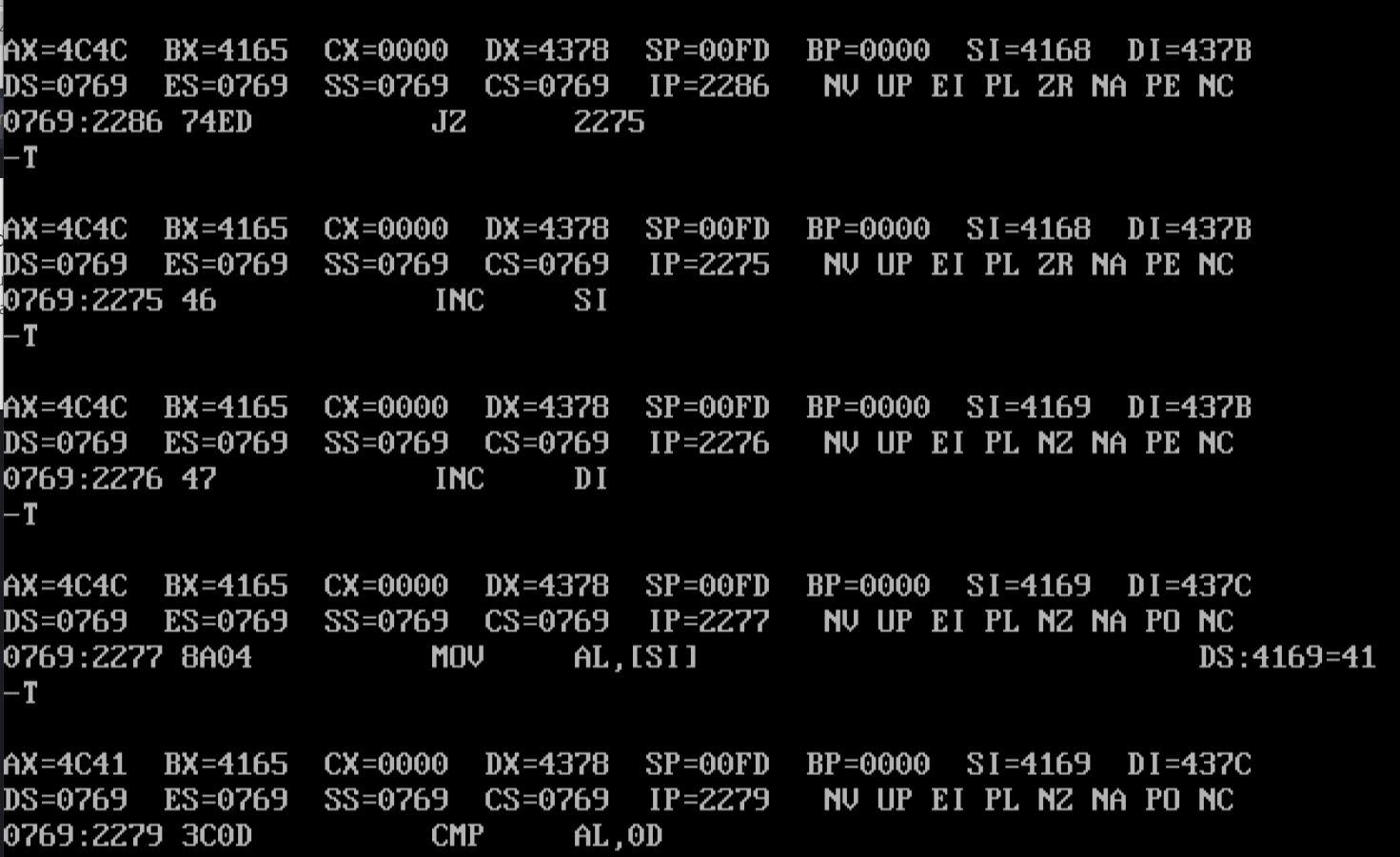
****

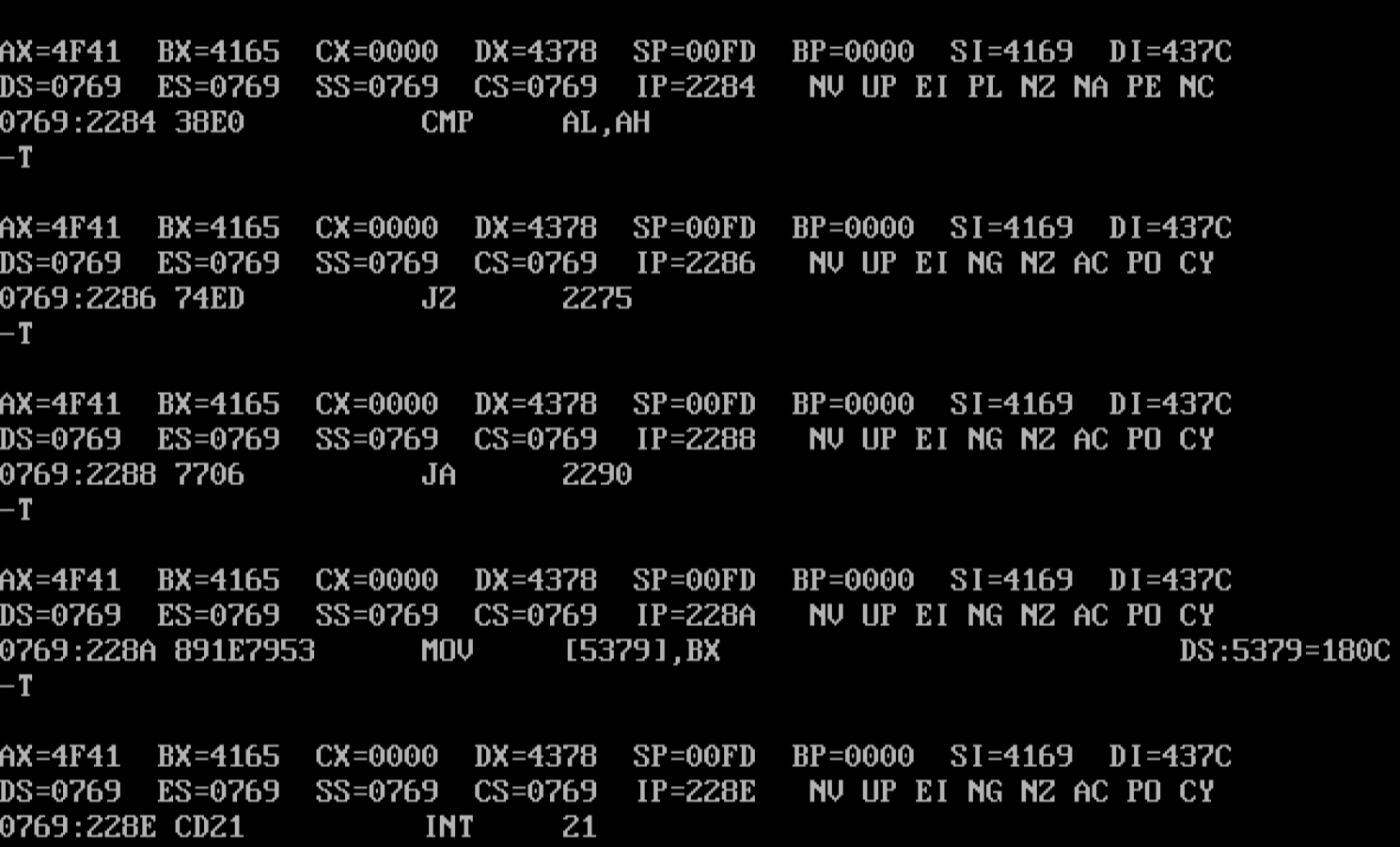
****

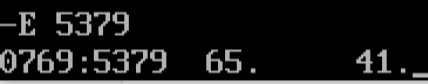
****

****

****

****

****

****

**EJERCICIO 22**

Inicio del programa en la posición de memoria: 2269

Inicio de área de datos en la posición de memoria: 4165

**1. Una descripción de lo que el programa realiza**

Con una secuencia de código, se verifica el estado de la memoria de un computador. Para ello se escribe valores en determinadas posiciones de memoria, luego se verifica que realmente existan esos valores en cada una de las posiciones escritas y detectar en cuales de ellas no lo están (posiciones defectuosas) las cuales se guardaran en una lista.

**2. Para qué se usa cada registro:**

**SI**: registro que apunta al comienzo de las posiciones de memoria que se verificará su estado

**DI**: registro que apunta al comienzo de la lista de posiciones defectuosas.

**AL**: registro que se le carga con 8 unos para luego ser grabados en las posiciones de memoria elegidas

**3. Dónde se ubican las variables y qué valores les has asignado:**

**4165**: posición de memoria donde se guarda el primer valor

**1378**: posición de memoria donde se guardarán las posiciones defectuosas.

**CAPTURAS DOSBOX**

