- **Número USP:** 11726111
- **Nome completo do aluno:** Santiago Quintero Hincapié

Primeira questão

A funcionalidade deste program é: mostrar na saída o código ASCII de uma sequência de letras, neste caso, os dados conseguidos com meu nome do item 2 da *preparação dos dados*, ou seja:

Hexa	letra	Decimal
53	S	83
41	A	65
4E	N	78
54	T	84
49	I	73
41	A	65
47	G	71
4F	O	79
53	S	83
41	A	65
4E	N	78
54	T	84
49	I	73
41	A	65
47	G	71
4F	O	79

O código objeto carregável é

Tabela do Programa:

```
IC   CO  OP   Rótulo
100  LD  f00  STAR
102  MM  201
104  LD  202  IN
106  MM  10c
108  LD  203
10a  MM  10d
10c  LD  f01  LDA
10e  PD  004
110  LD  201
112  -   200
114  MM  201
116  JZ  12a
118  LD  10d
11a  +   200
11c  MM  10d
11e  JZ  122
120  JP  10c
122  LD  10c  INCR
124  +   200
126  MM  10c
128  JP  10c
12a  HM  100  FORA
```

Iniciar o contador

Iniciar a instrução LD do endereço 10c

Mostrar o dado correspondente á letra na saída

Atualizar o contador e perguntar se é zero

Atualizar a instrução LD do endereço 10c

Jump do loop

Atualizar a instrução LD do endereço 10c

Terminar

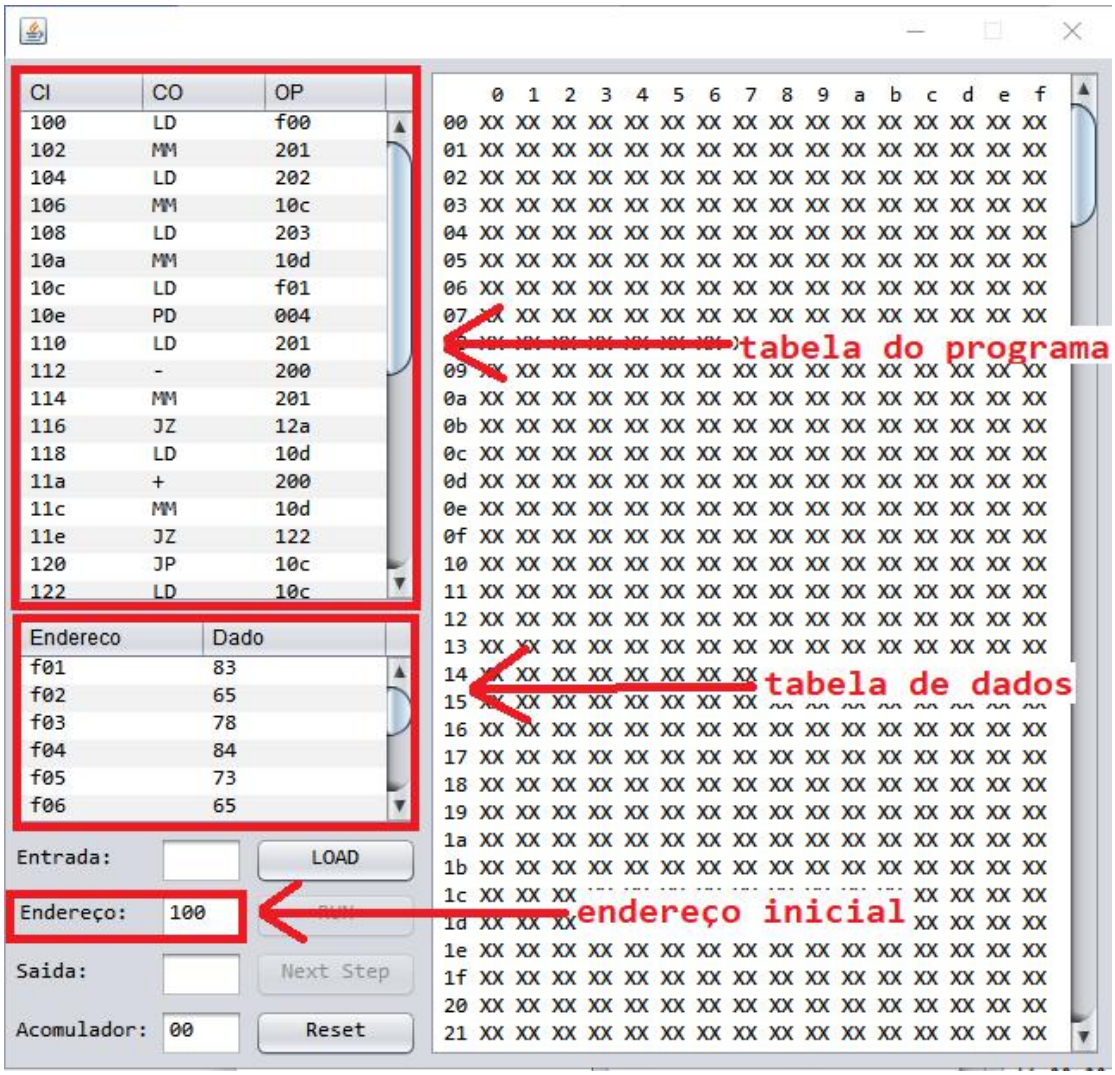
Tabela de dados: esta tabela trabalha com números decimais, por isso transformei os numero hexadecimais em números decimais.

```
200 1    //UM
201 0    //CONTADOR
202 143  //8f em hexa
203 1    //01 em hexa
f00 16   //comprimento do texto
f01 83   //53 em hexa “S”
f02 65   //41 em hexa “A”
f03 78   //4E em hexa “N”
f04 84   //54 em hexa “T”
f05 73   //49 em hexa “I”
f06 65   //41 em hexa “A”
f07 71   //47 em hexa “G”
f08 79   //4F em hexa “O”
f09 83   //53 em hexa “S”
f0a 65   //41 em hexa “A”
f0b 78   //4E em hexa “N”
f0c 84   //54 em hexa “T”
f0d 73   //49 em hexa “I”
f0e 65   //41 em hexa “A”
f10 71   //47 em hexa “G”
f11 79   //4F em hexa “O”
```

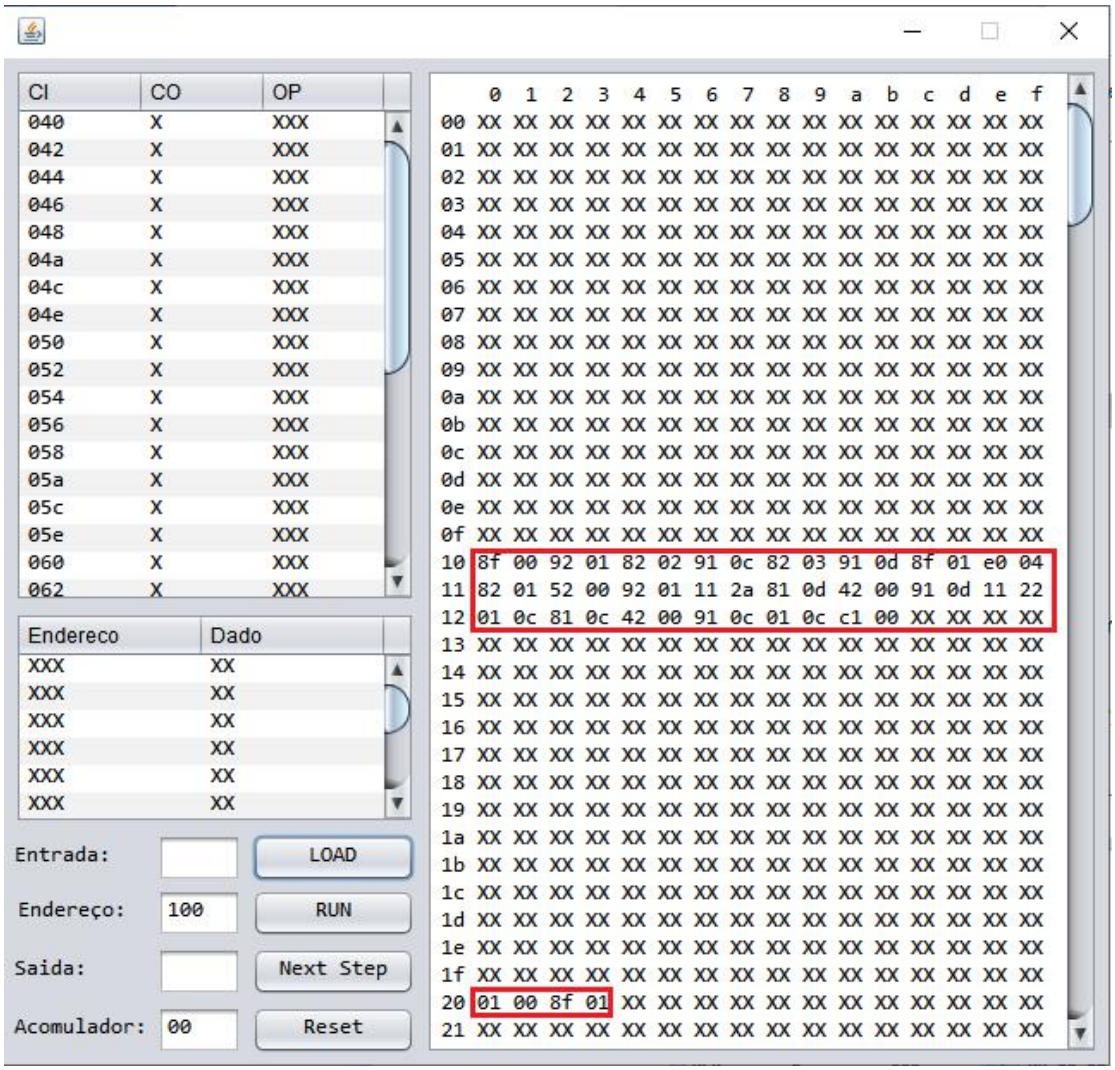
Instrução LDA INIC

Implementação do código objeto na MVN

Primeramente escrevemos os dados e as instruções no programa:

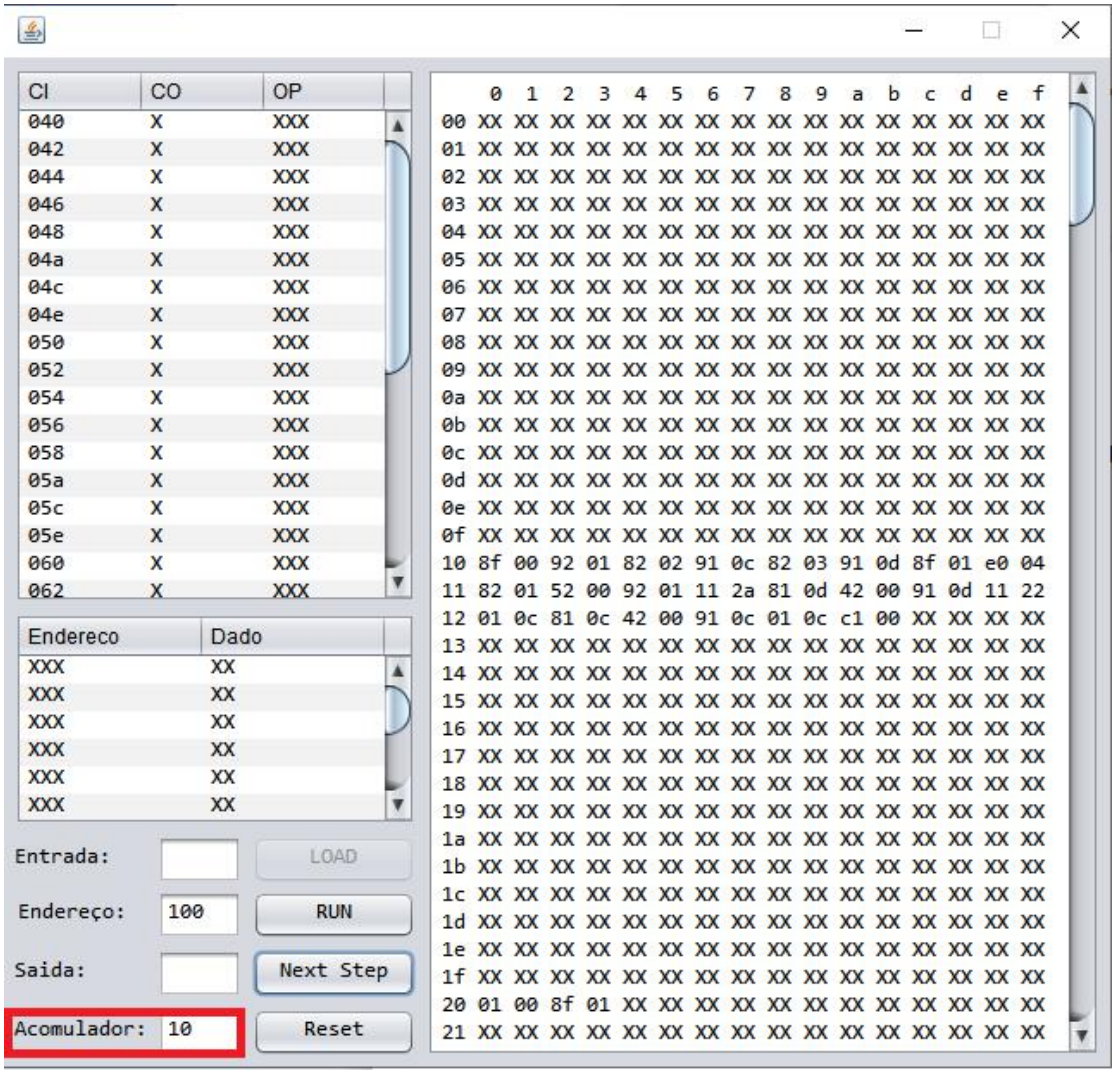


Logo damos click em *LOAD* para montar tudo na memoria:



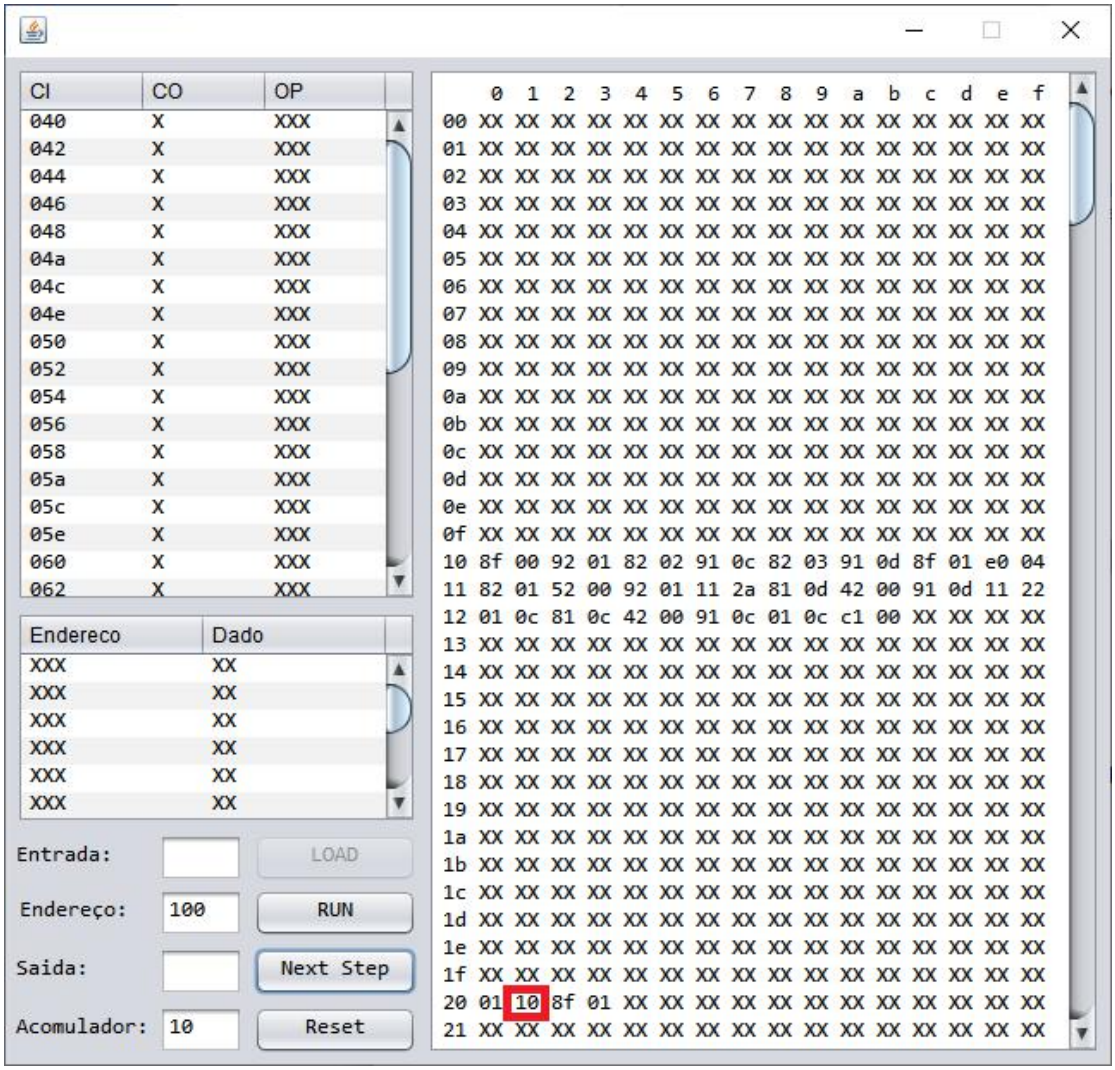
Pode-se ver parte do código de maquina na memoria.

Executando a primera linha (click em Next Step): **LD** **f00**



O acumulador pega o que esta no endereço f00, ou seja, o comprimento do vector de letras o qual é 16 (o acumulador sempre mostra tudo em código de maquina, ou seja, em hexadecimal).

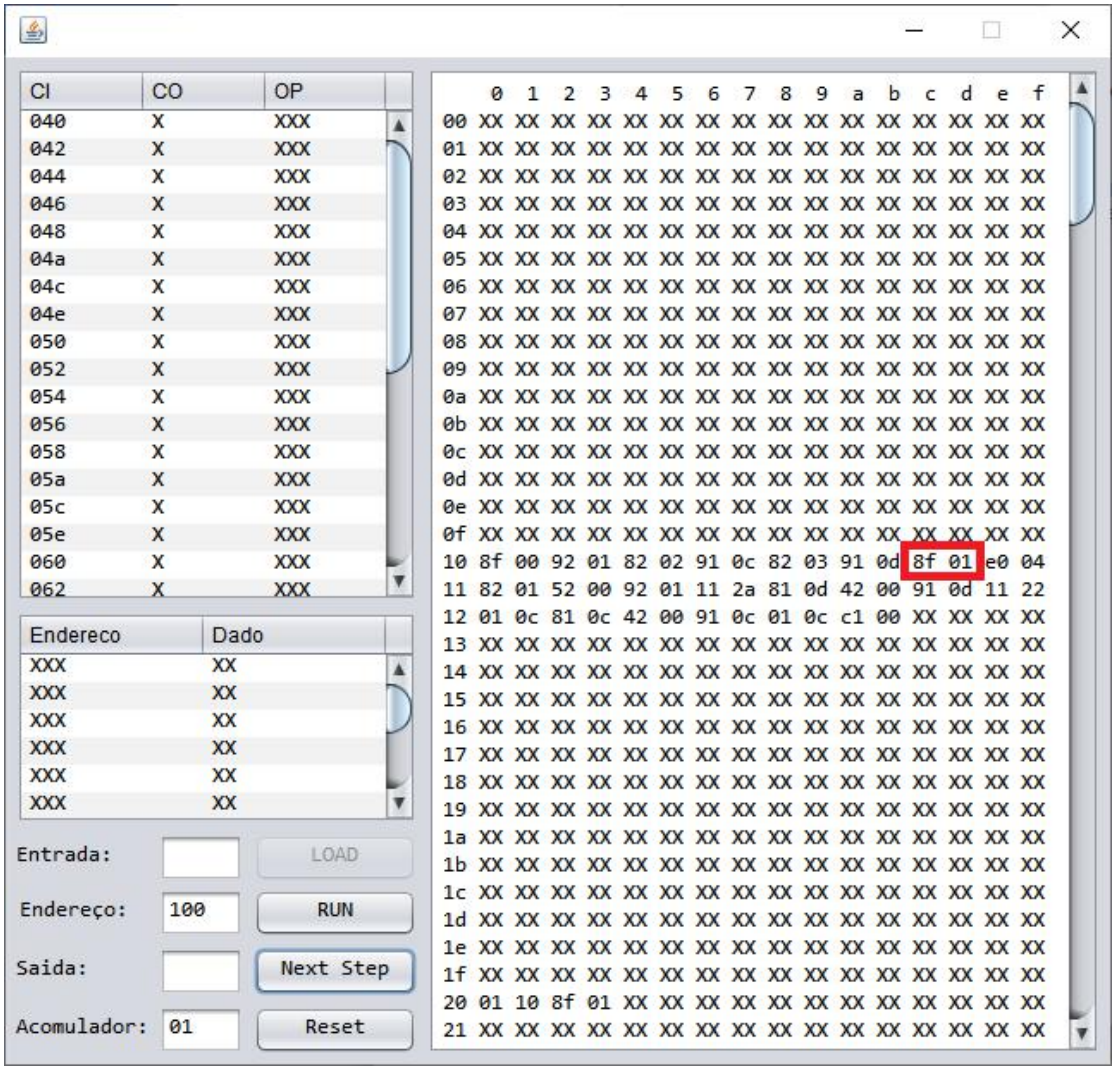
Executando a próxima linha (click em Next Step): **MM 201**



Escreve na memoria no endereço 201 o que tinha no acumulador (10).

Executando as próximas linhas:

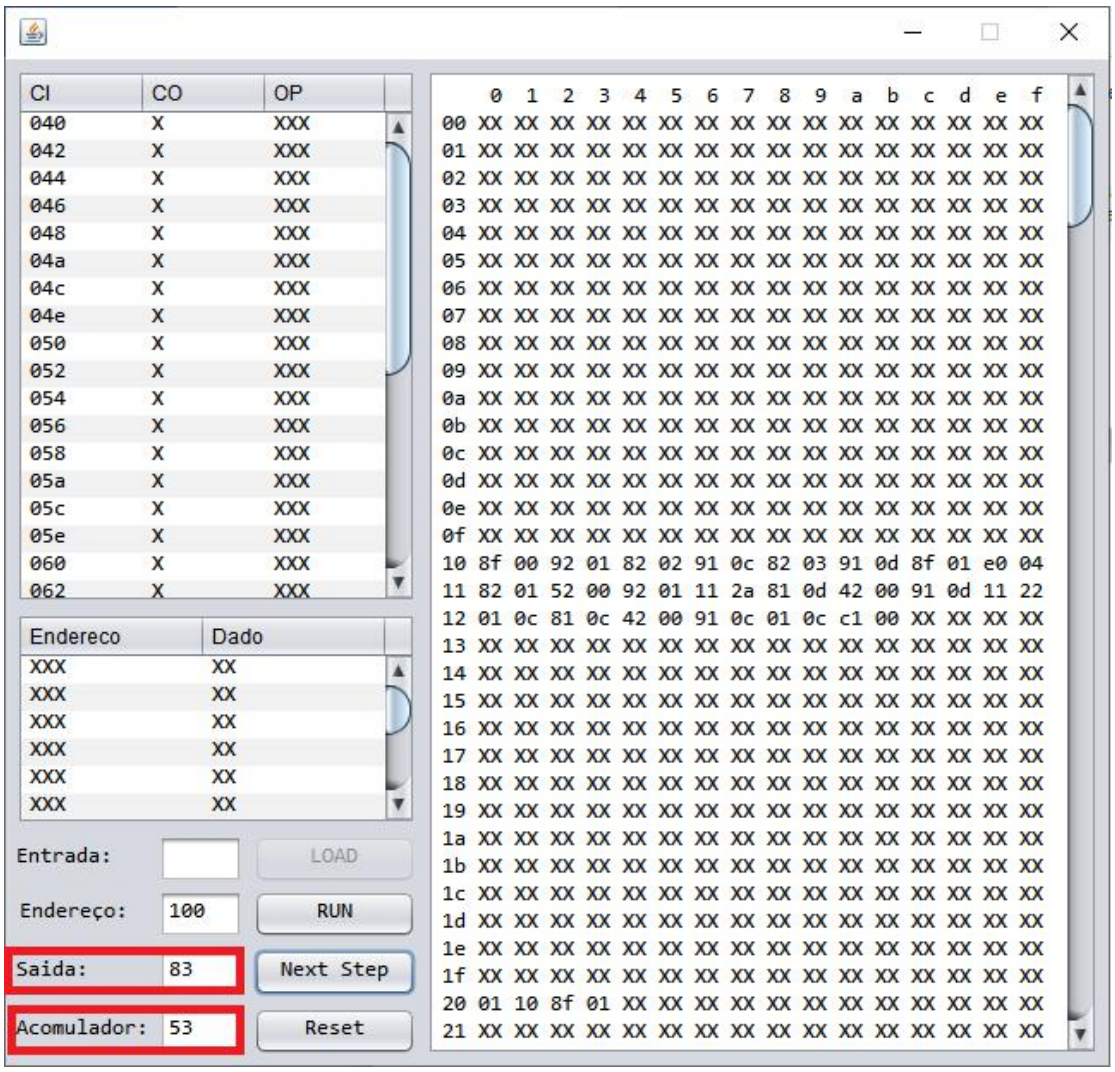
```
LD   202
MM   10c
LD   203
MM   10d
```



Atualizo a instrução LDA INIC no endereço 10c.

Executando as próximas linhas:

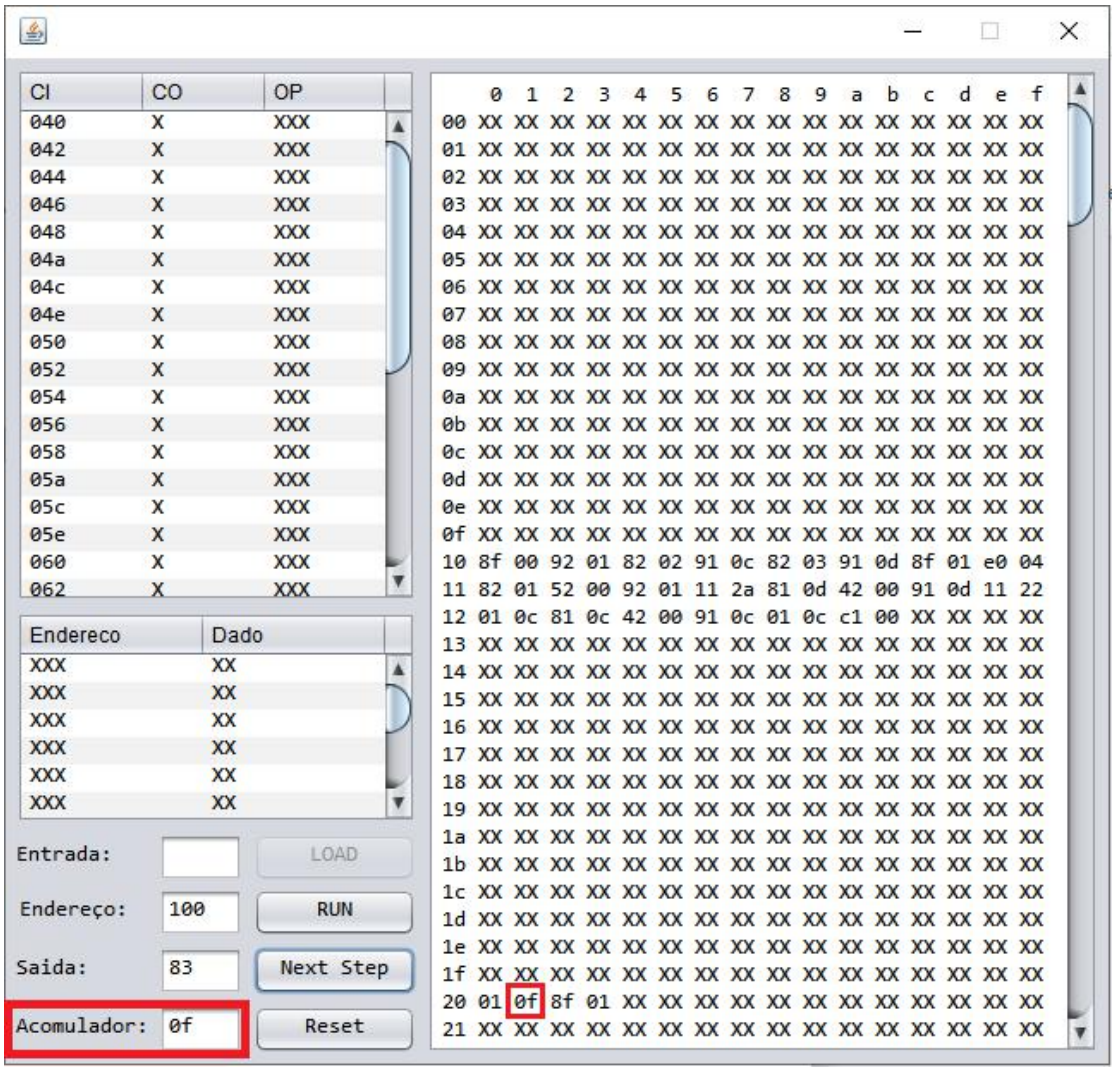
```
LD   f01
PD   004
```



Mostra a primera posição do vector de letras, neste caso “S”, o qual é 53 em ASCII (1010011) e a saída do programa sempre se mostra em decimal por isso aparece 83.

Executando as próximas linhas:

```
LD   201
-    200
MM   201
JZ   12a
```

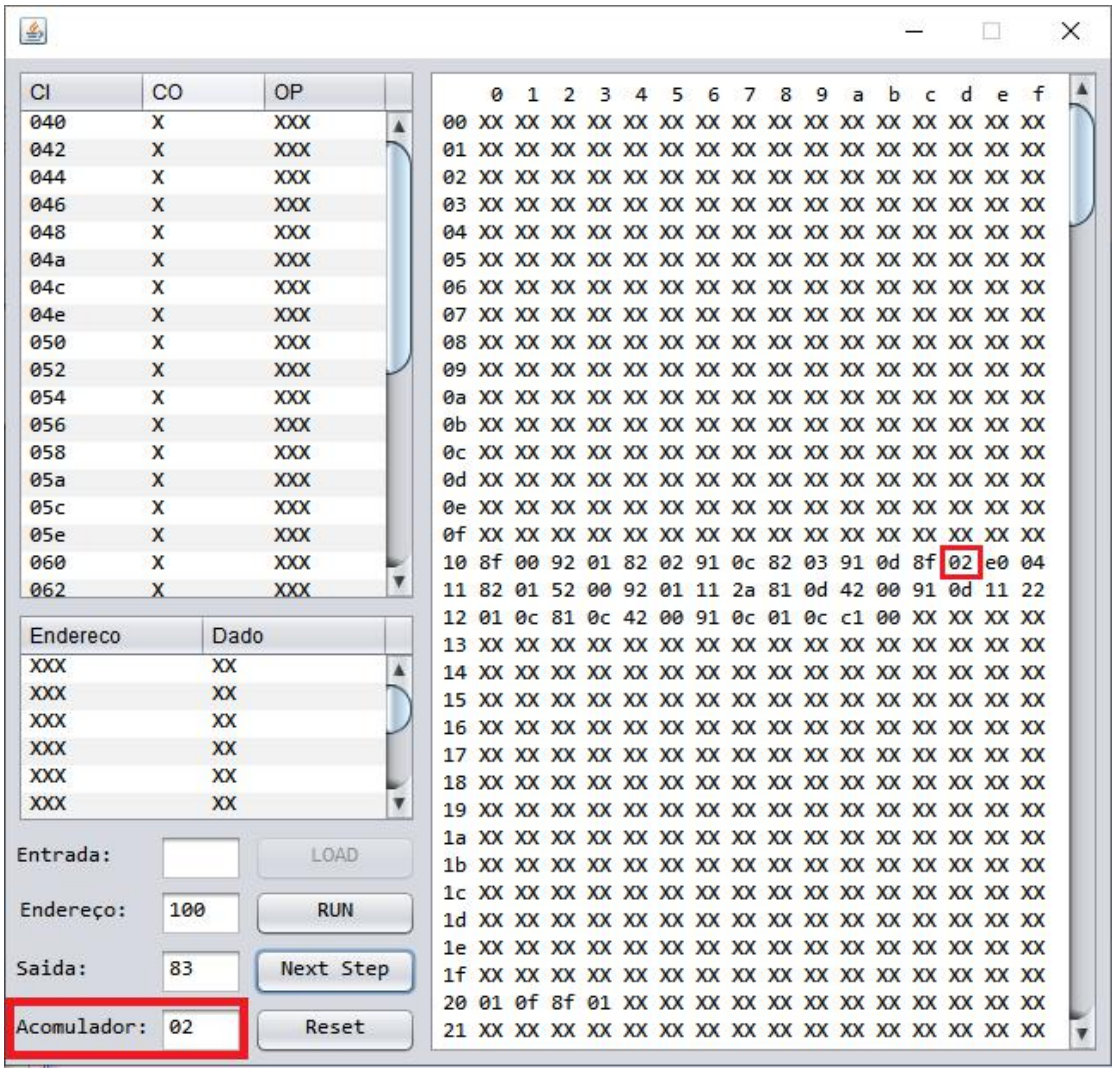


Diminui o contador em um e guarda em memoria no endereço 201 e se é 0 termina o loop.

E assim por diante executa tudo o código, mostrando o vector de letras e disminuido o contador até 0.

Executando as próximas linhas:

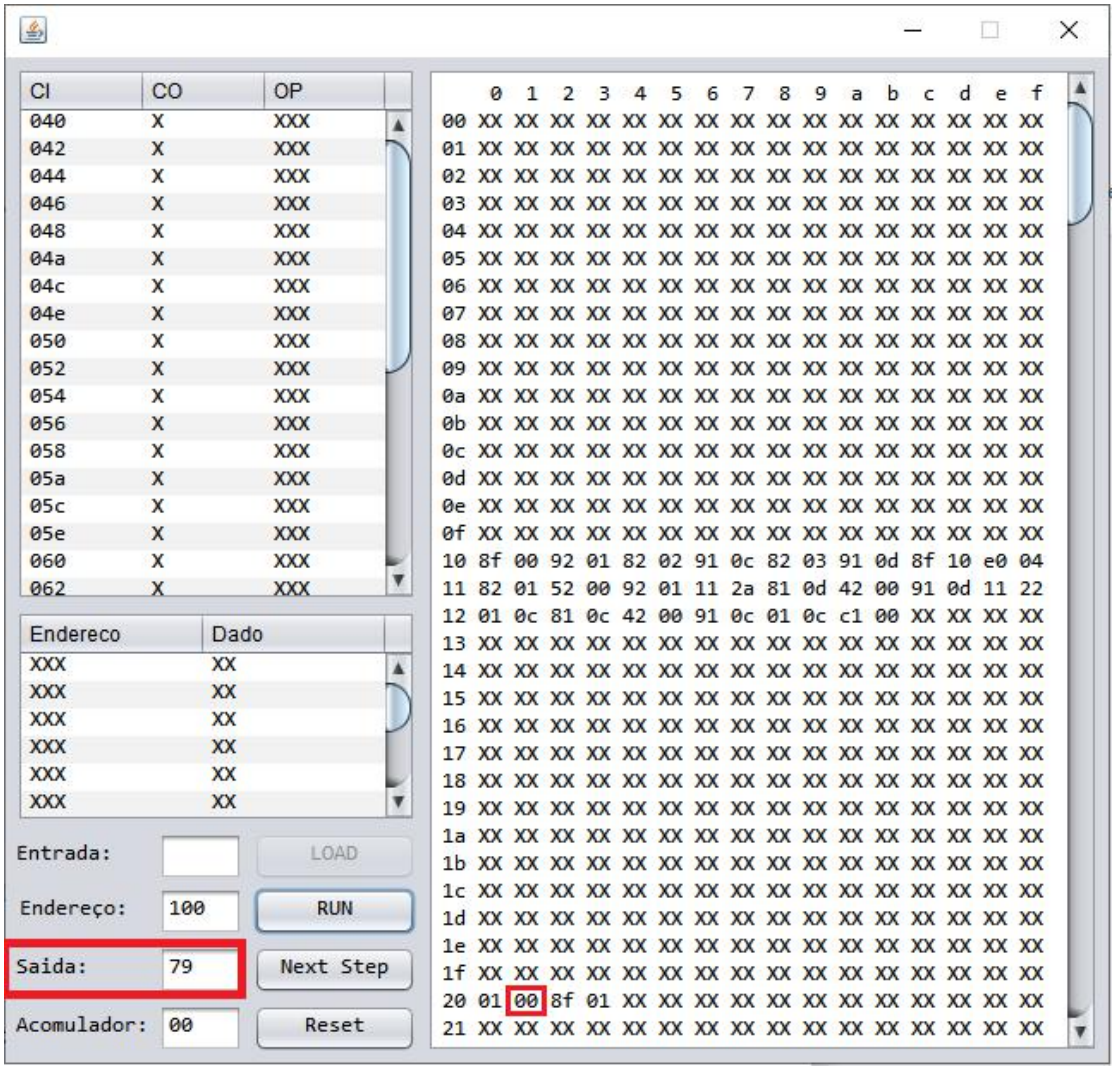
```
LD   10d
+    200
MM   10d
```



Atualiza a instrução LDA INIC no endereço 10d.

E assim por diante, executa tudo o código, mostrando o vector de letras, disminuido o contador até 0 e atualizando a instrução LDA INIC.

Finalmente executando tudo o código (click no botão RUN), na imagem baixo pode-se ver o resultado na saída (79), o qual é o numero em decimal da ultima letra (“O”) de minha sequencia em ASCII 4f (1001111) e pode-se ver também o contador na memoria em 0.



Se quiser executar o programa dar click aquim e descarregar o executável (MVNQ1.jar).
Só tem que abrir o executável, dar click em LOAD e executar com RUN ou Next Step no caso Passo a Passo.

Segunda questão

Funcionalidade do program: Começa pelo dígito menos significativo (mais à direita), e vá preenchendo a célula correspondente do Código com o resto da divisão por 16 a soma dos valores dos dígitos já considerados até aquele ponto.
Ao final, o Código terá sido preenchido com oito valores, entre 0 e 15. Denote cada m deles na forma de um dígito hexadecimal.

NUSP	1	1	7	2	6	1	1	1
CODIGO	4	3	2	b	9	3	2	1

O programa recebe o vector NUSP em ASCII ou seja:

NUSP	1	1	7	2	6	1	1	1
ASCII	31	31	37	32	36	31	31	31

E com este vector ASCII gera o vector CODIGO.

Código objeto carregável

```
CI   CO   OP       ROTULO   COMENTARIO

411 LD   COMP     START    //comprimento do vector NUSP
413 MM   CONT              //inicia o contador do loop

415 LD   LD       IN       //constrói inicialmente a instrução LD
417 MM   LD0               //a ser diminuída a cada execução do loop
419 LD   LD+1              //a qual faz referencia ao vetor que contem
41b MM   LD0+1             //o numero USP.
41d LD   MM                //depois constrói-se a instrução MM
41f MM   MM0               //a ser diminuída a cada execução do loop
421 LD   MM+1              //a qual faz referencia ao vetor que contem
423 MM   MM0+1             //o CODIGO.

425 LD   NUSP+7   LD0      //obter o numero USP
427 -    0a9               //transformação de ASCII a decimal
429 MM   NUSP+n            //salvar o resultado
42b +    SUM               //realizar a suma com o acumulado
42d MM   SUM               //salvar a suma
42f -    BASE              //identificar se é maior que a base
431 JN   FUNC              //se é maior
433 LD   SUM               //carrega o resultado da suma
435 -    BASE              //resta a base
437 MM   SUM               //e salva o resultado
439 LD   SUM      FUNC     //se nao, carrega o resultado da suma
43b MM   CODIGO+7 MM0      //e salva o resultado no vetor CODIGO

43d LD   MM+1              //pega a posição atual no vetor CODIGO
43f -    UM                //diminui a posição
441 MM   MM+1              //salva em memoria a siguiente posição
443 LD   LD+1              //pega a posição atual no vetor NUSP
445 -    UM                //diminui a posição
447 MM   LD+1              //salva em memoria a siguiente posição
449 LD   CONT              //pega o contador
44b -    UM                //diminui um
44d MM   CONT              //salva o contador
44f JZ   FORA              //se o contador é zero, termina
451 JP   IN                //se não, começa com o siguiente número USP
```

```
453 HM  START   FORA    //pára a máquina e volta ao início se for reacionada

----------------------------

200     145     MM      //91 em hexa
201     16      MM+1    //10 em hexa
202     129     LD      //81 em hexa
203     8       LD+1    //08 em hexa
204     48      0a9     //30 em hexa
205     16      BASE    //10 em hexa
206     1       UM
207     0       CONT
208     0       NUSP+n
209     0       SUM     //acumulado

----------------------------

100     8       COMP
101     49      NUSP    //31 em hexa //1 em ASCII
102     49              //31 em hexa //1 em ASCII
103     55              //37 em hexa //7 em ASCII
104     50              //32 em hexa //2 em ASCII
105     54              //36 em hexa //6 em ASCII
106     49              //31 em hexa //1 em ASCII
107     49              //31 em hexa //1 em ASCII
108     49              //31 em hexa //1 em ASCII

109     0       CODIGO  //Debe resultar 4
10a     0               //Debe resultar 3
10b     0               //Debe resultar 2
10c     0               //Debe resultar b
10d     0               //Debe resultar 9
10e     0               //Debe resultar 3
10f     0               //Debe resultar 2
110     0               //Debe resultar 1
```

Instrução MM XXX

Instrução LD XXX

Uma vez tendo este código se compatibiliza com meu montador e fica assim:

Tabela do Programa:

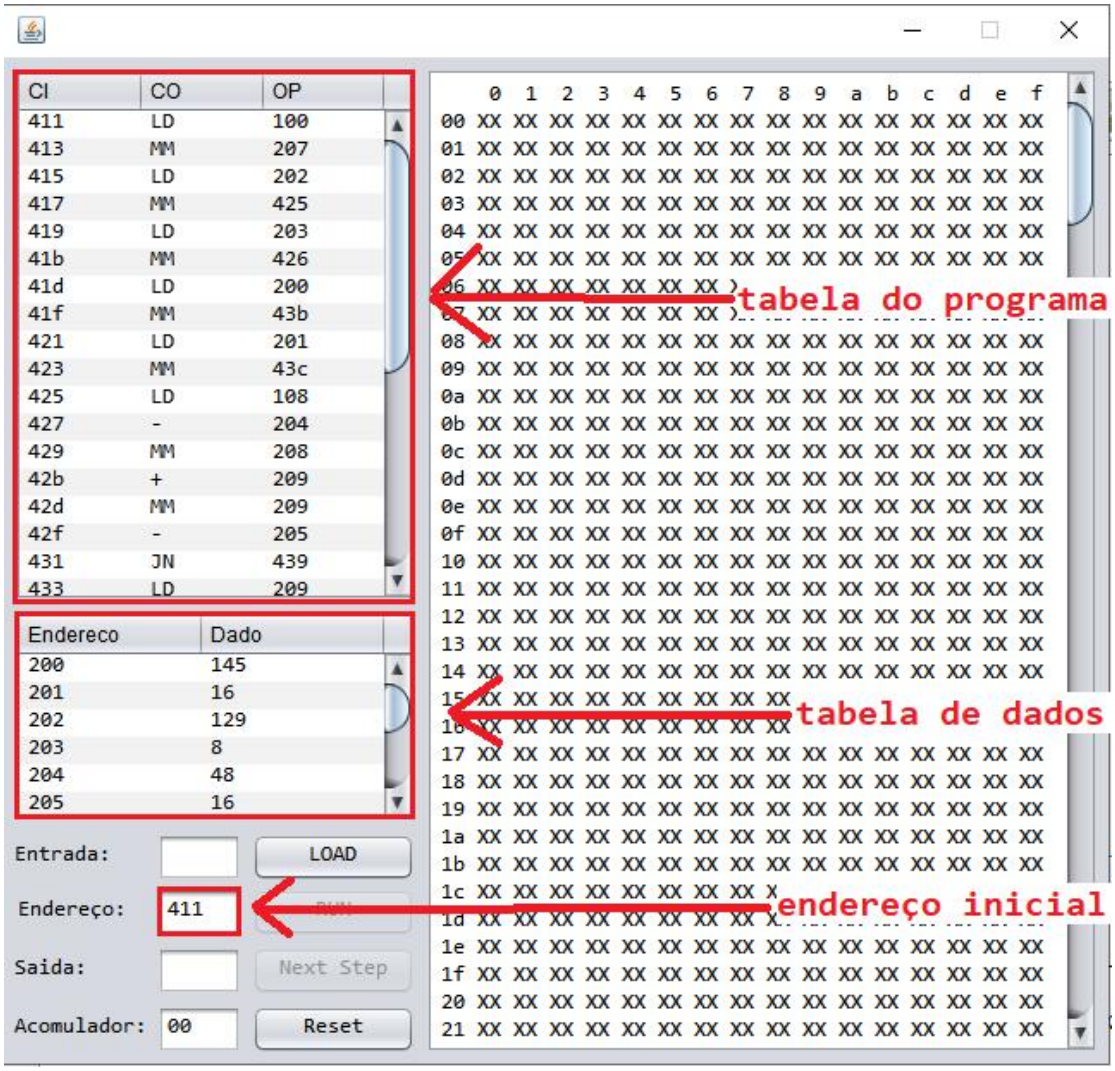
```
IC   CO  OP
411  LD  100
413  MM  207

415  LD  202
417  MM  425
419  LD  203
41b  MM  426
41d  LD  200
41f  MM  43b
421  LD  201
423  MM  43c

425  LD  108
427  -   204
429  MM  208
42b  +   209
42d  MM  209
42f  -   205
431  JN  439
433  LD  209
435  -   205
437  MM  209
439  LD  209
43b  MM  110

43d  LD  201
43f  -   206
441  MM  201
443  LD  203
445  -   206
447  MM  203
449  LD  207
44b  -   206
44d  MM  207
44f  JZ  453
451  JP  415

453  HM  411
```

Tabela de dados:

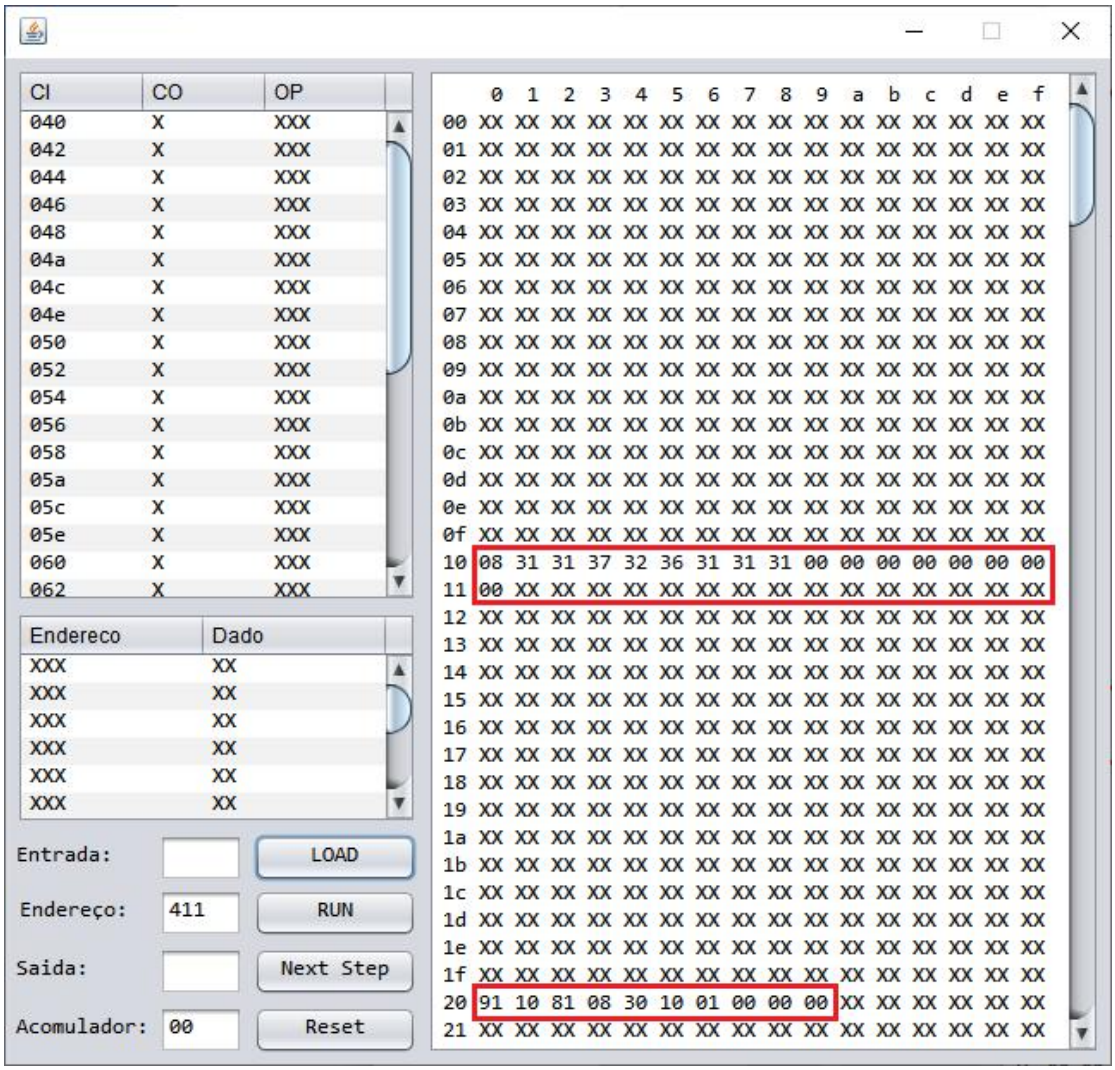
```
200     145  MM       //91 em hexa
201     16   MM+1     //10 em hexa
202     129  LD       //81 em hexa
203     8    LD+1     //08 em hexa
204     48   0a9      //30 em hexa
205     16   BASE     //10 em hexa
206     1    UM
207     0    CONT
208     0    NUSP+n
209     0    SUM

100     8    COMP
101     49   NUSP//31 em hexa, 1 em ASCII
102     49        //31 em hexa, 1 em ASCII
103     55        //37 em hexa, 7 em ASCII
104     50        //32 em hexa, 2 em ASCII
105     54        //36 em hexa, 6 em ASCII
106     49        //31 em hexa, 1 em ASCII
107     49        //31 em hexa, 1 em ASCII
108     49        //31 em hexa, 1 em ASCII

109     0    CODIGO   //Debe resultar 4
10a     0             //Debe resultar 3
10b     0             //Debe resultar 2
10c     0             //Debe resultar b
10d     0             //Debe resultar 9
10e     0             //Debe resultar 3
10f     0             //Debe resultar 2
110     0             //Debe resultar 1
```

Implementação do código objeto na MVN

Primeramente escrevemos os dados e as instruções no programa:



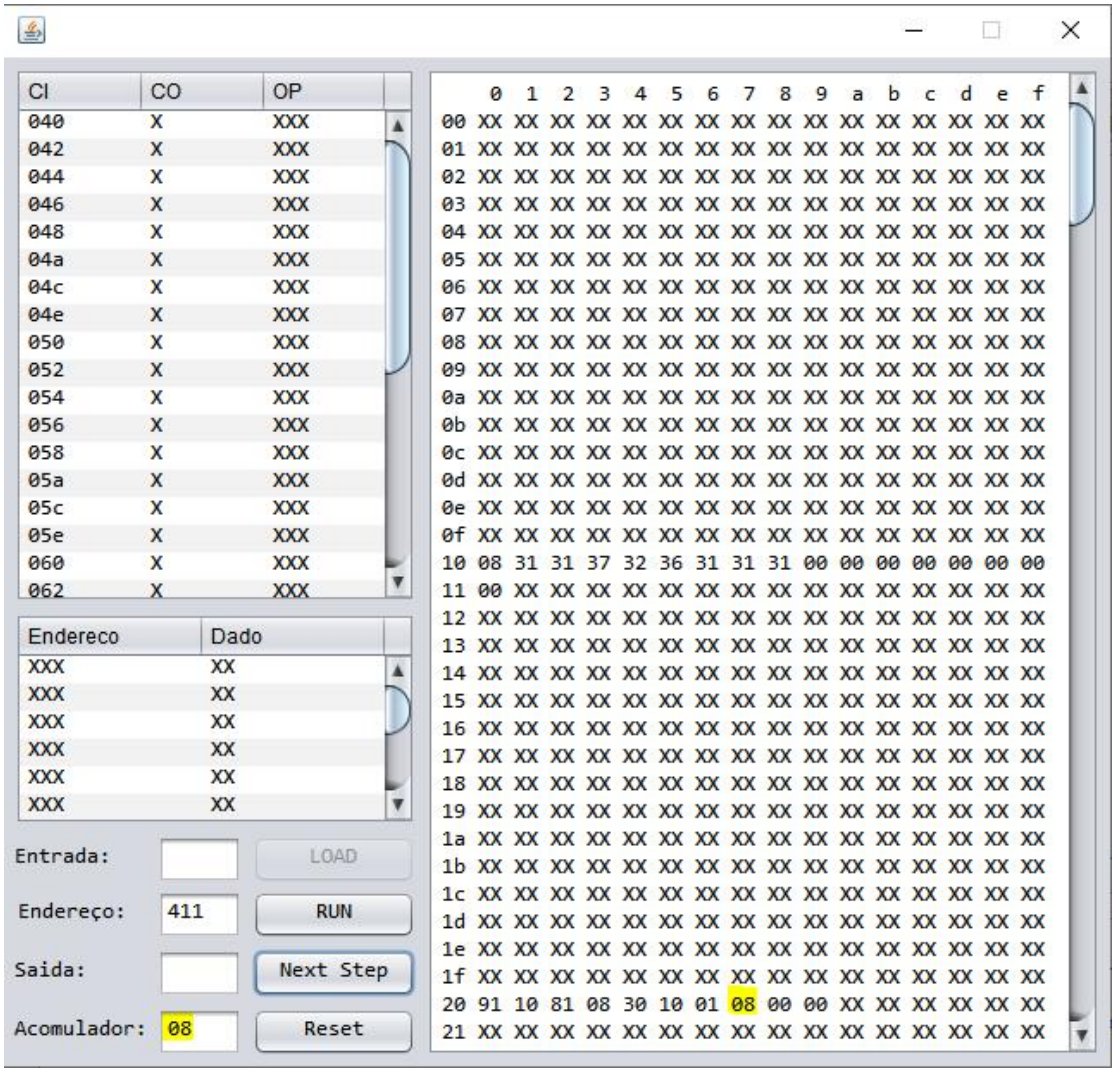
Logo damos click em *LOAD* para montar tudo na memoria:



Pode-se ver parte do código de maquina na memoria.

Executando as próximas linhas:

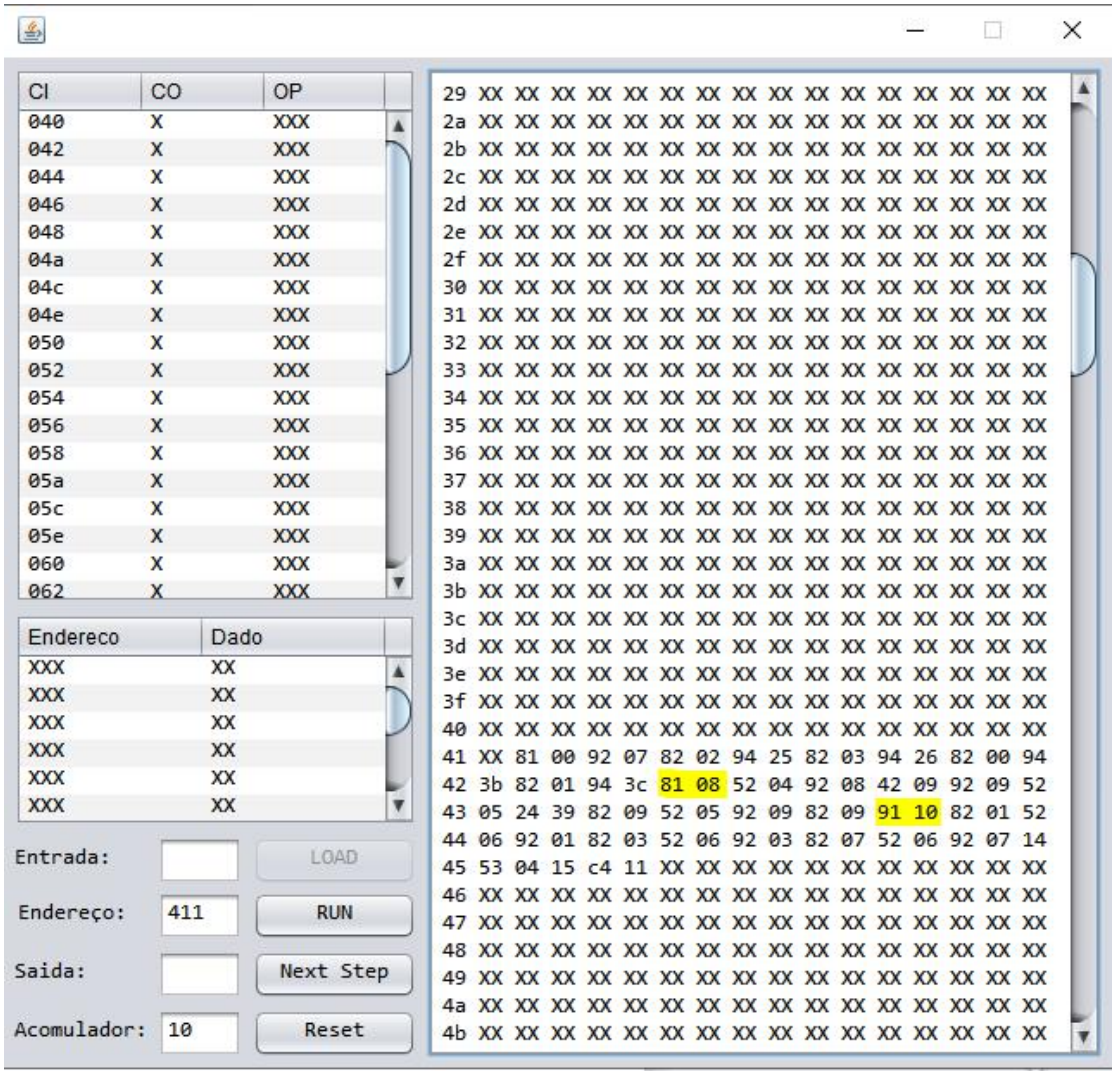
```
LD   100
MM   207
```



Escreve em memoria o comprimento do vetor NUSP (8).

Executando as próximas linhas:

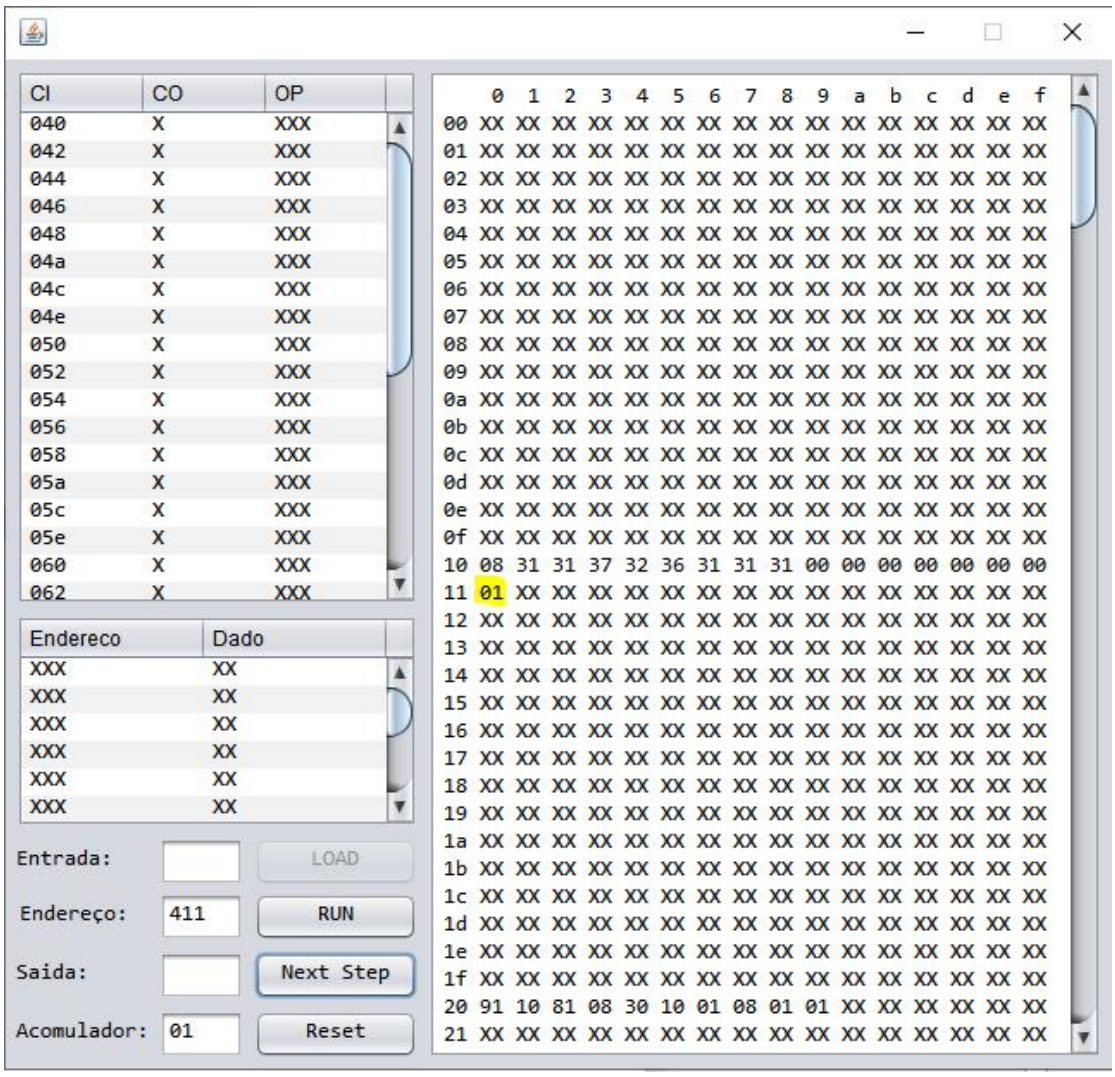
```
LD   202
MM   425
LD   203
MM   426
LD   200
MM   43b
LD   201
MM   43c
```



Inicia as instruções LD XXX e MM XXX.

Executando as próximas linhas:

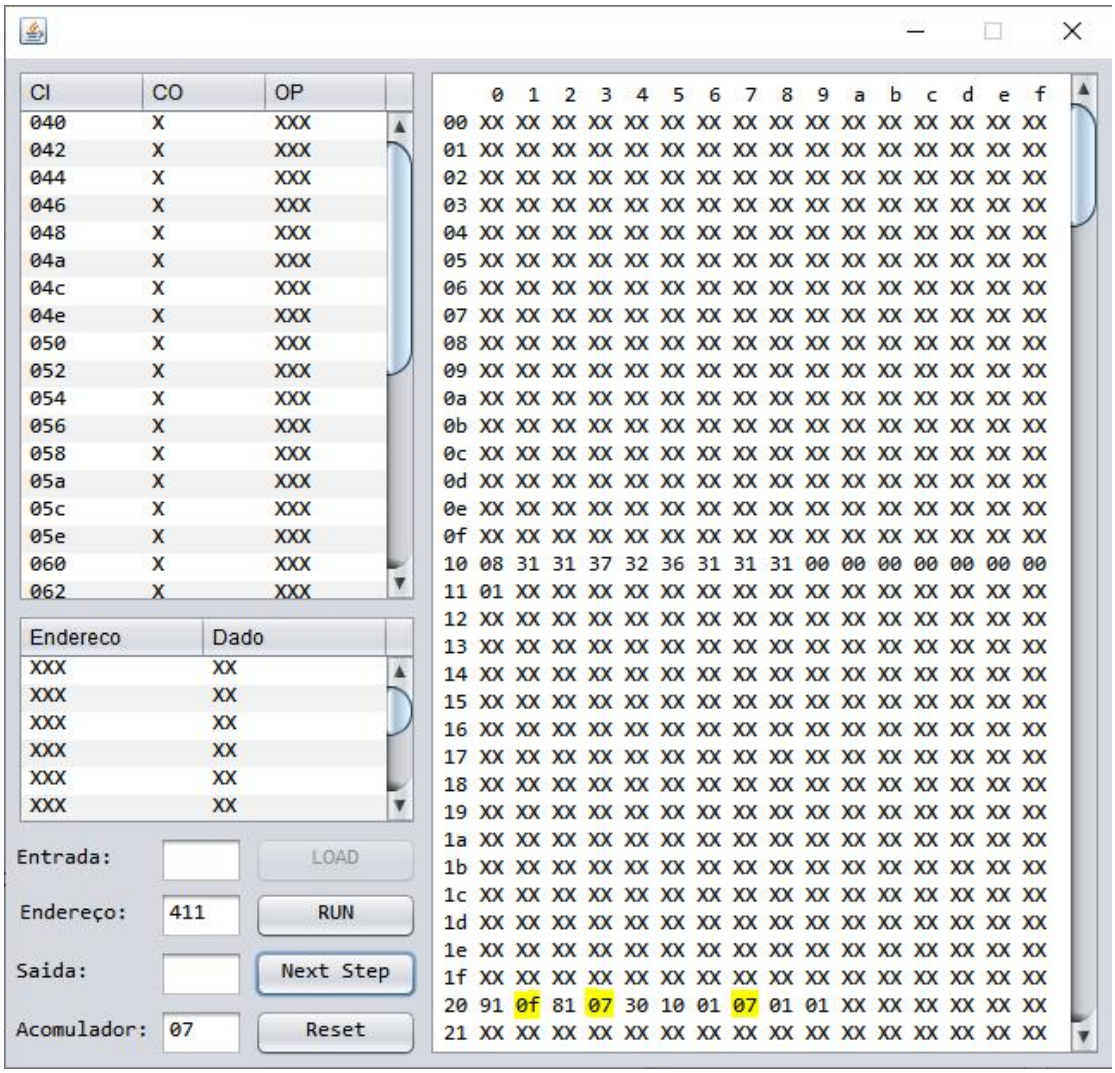
```
425 LD   108
427 -    204
429 MM   208
42b +    209
42d MM   209
42f -    205
431 JN   439
433 LD   209
435 -    205
437 MM   209
439 LD   209
43b MM   110
```



Pega o digito menos significativo do vetor NUSP e começa a criar o vetor CODIGO.

Executando as próximas linhas:

```
43d LD   201
43f -    206
441 MM   201
443 LD   203
445 -    206
447 MM   203
449 LD   207
44b -    206
44d MM   207
44f JZ   453
451 JP   415
```

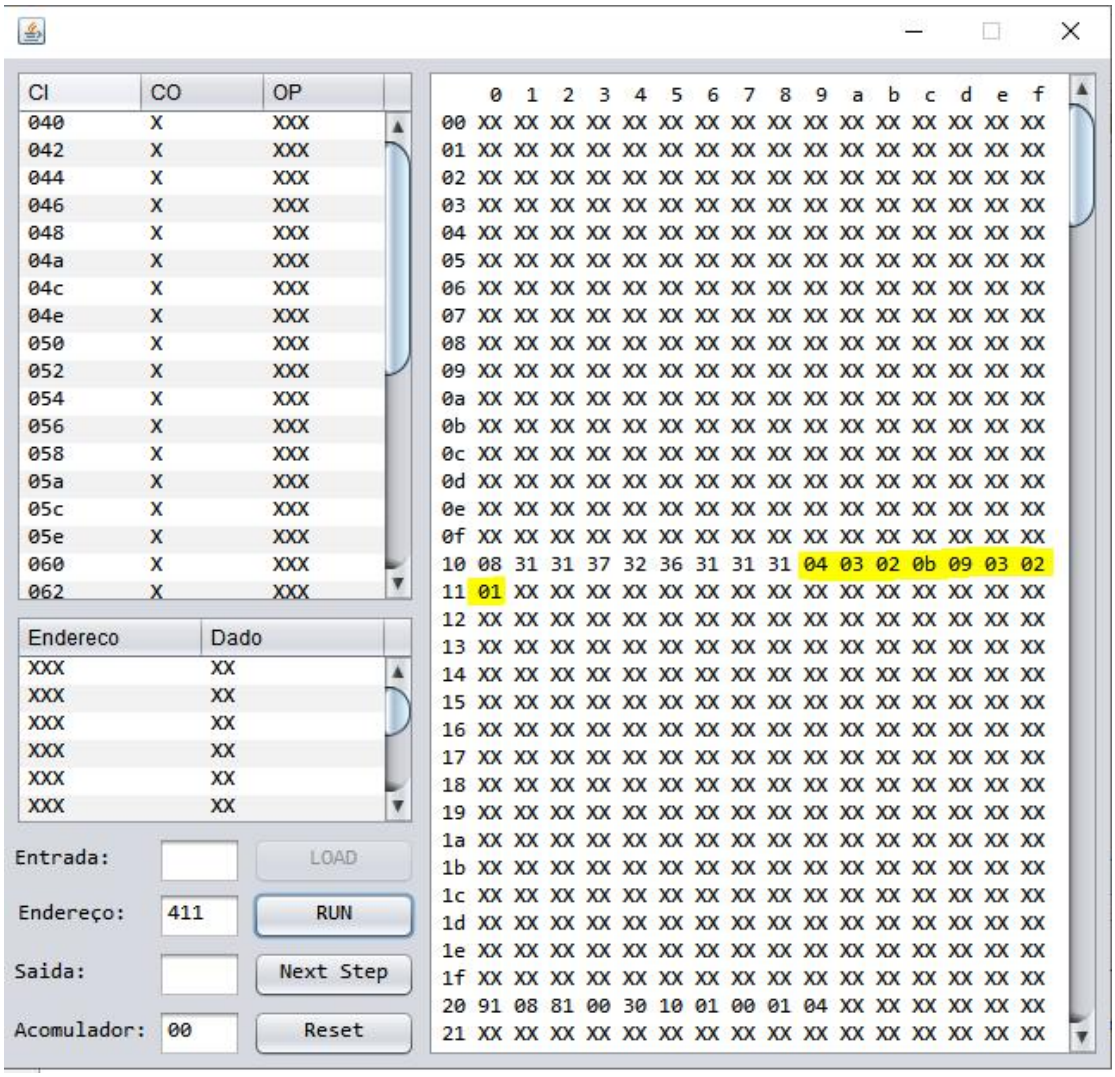


Diminui o contador, a posição a ser trabalhada do vetor NUSP e do vetor CODIGO, ou seja, as instruções LD XXX e MM XXX.

E assim por diante, executa tudo o código, disminuido o contador, atualizando as instruções LD XXX e MM XXX e criando o vetor CODIGO até que o contador seja 0.

Finalmente executando o código tudo (com o botão RUN), na memoria pode-se ver o vetor CODIGO esperado:

CODIGO	4	3	2	b	9	3	2	1



Se quiser executar o programa dar click aquim e descarregar o executável (MVNQ2.jar).
Só tem que abrir o executável, dar click em LOAD e executar com RUN ou Next Step no caso Passo a Passo.