

Project II Delay

As you all know, Project 2 has been delayed. We will be introducing a new project II in its place. The new project is described below. It is to be coded in Java.

New Project IIa – same scoring as for project II

Your team is to write an assembler for the simulator that encodes all instructions . It will take a text file input as shown below and translate the code into a file identical in format to the input file that was required in Project I. We have had some team dropouts, and this will serve to sharpen your Java skill and to make the programs for Project II and Project III easier to develop.

An example text format for the file is given below.

```
LOC    6
Data   10
Data   3
Data   End
Data   0
Data   12
Data   9
Data   18
Data   12
LDX    2,7
LDR    3,0,10
LDR    2,2,10
LDR    1,2,10,1
LDA    0,0,0
LDX    1,9
JZ     0,1,0
LOC    1024
End:   HLT
```

Your assembler is to translate the entire file and then close it. An annotated file is given below

Note the addresses for your understanding. The source file uses decimal but the translated file uses hex locations as the Project I file did.

	LOC	6
0006 000A	Data	10
0007 0003	Data	3
0008 0400	Data	End
0009 0000	Data	0
000A 000C	Data	12
000B 0009	Data	9
000C 0012	Data	18
000D 000C	Data	12
000E 8480	LDX	2,7
000F 070A	LDR	3,0,10
0010 060A	LDR	2,2,10
0011 05AA	LDR	1,2,10,1
0012 0C00	LDA	0,0,0
0013 8449	LDX	1,9
0014 8040	JZ	0,1,0
	LOC	1024
0400 0000	End:	HLT

- You may turn in a file like the above if you like but only the first two columns are required for the text version of the input file.
- Labels can be upper or lower case but must end in ":"
- Assume the code starts at 0 if no LOC is provided at beginning
- Note that the loader will allow loads below location 6
- If you like you can comment your input file, you will have to ignore comments.

Use a Split function on source file

1. Split by blanks or no- printable characters then trim each
2. Then split the part with the instructions by ","

Use two passes

1. Open input file and compute location of all labels, store the label and value
2. Close input file
3. Open input file and read through to produce output file and close.
4. Use a table or dictionary like object with numbers for the op codes
5. Write a function that will take the text op code part and return the hexadecimal for the instruction