Homework 1A

Problem Statement

Extend the emulator that you have written to support an encoding functionality. When you encounter a *.encode* label in the program, you need to print the hexadecimal representation of the instruction.

For example,

```
.encode: add r1, r2, r3
```

must print

```
0x0048c000
```

on a separate line.

Note that the *.print* macro also needs to be supported. Secondly, we might have multiple instances of the .encode label. This is the only label that is allowed to have multiple instances.

Every label is associated with the next statement. There is no necessity for having the statement and the label on the same line. Your emulator needs to be able to operate on a file that has a "free format". This means that it should be oblivious of the number of spaces, tabs, new line characters, and carriage returns. The only assumption that you can make is that a statement will not be split across two lines, and we will never have two statements in one line.

Submission Instructions

- Create a .tar.gz archive for all your files (no directories in the archive).
    1. Name the file <entry number>.tar.gz.
    2. The entry number starts with the year of entry (not user id).
- The command make should compile all the files.
- There should be a file called run.sh .
- Here, are the three commands that we will issue
    1. tar -xvfz <tar file name>
    2. make
    3. ./run.sh <assembly file name>

Deadline

12^{th} October, 2013, Saturday, 23:55 hrs.
Submission instructions to be mailed later.