Virtual Machines: Concepts and Applications
Homework Assignment 3
(Programming Assignment)

Now that you have implemented a Fetch & Dispatch emulator for the simplified x86 virtual machine (in the previous assignment), the next step is to gather *profiling* data. You need to do two things:

- Determine static basic blocks (BB) (i.e. single entry and ends in call/jump/branch/ret and there are no branches to the middle of the block). *"prn"* instruction will NOT be counted as a call.
- Remember we are talking about static BB not dynamic BB (check lecture 2, slides 45-50).
- For each BB above find the number of times this BB has been executed.
- The program starts execution from the instruction with "start" label. If there are no such label, then execution will start from the beginning of the file.

It is highly suggested that you do the BB detection and stats by hand first on the *.vm files provided to check with the output of your program.

What is the format of the output?

The command line is the same as the previous assignment, with the same output on the screen. Beside the output on the screen, your VM must also generate a text file called progname.txt (where progname is the name of the program. So, if the input program is test.vm the output file will be test.vm.txt).
The format of that file is very simple. For each BB, it prints the frequency of execution of that BB (in a separate line), followed by the instructions of that BB (each instruction in separate line), followed by a blank line.
For example, if the input program is:

```
loop: inc eax
      cmp eax, 1000|h
      prn eax
      jl loop
```

The content of profile.txt will be (you can neglect labels in the output file):

```
4096
inc eax
cmp eax, 1000|h
prn eax
jl loop
```

What to submit?

Through NYU classes, submit your source code, as well as the output of your program for each one of the provided *.vm files.

Grading:

You are provided with 3 *.vm files.
- We will test your submission with those files, each one 5 points → 15 points
- We will also test with 2 non-distributed files, each one 10 points → 20 points
- We will check your source code and compile it → 5 points

Total = 40 points.

Have Fun!