Assembler

A 2-pass Assembler written in Python for a 12-bit accumulator architecture as part of my Computer Organisation course

To run

```
$ git clone https://github.com/arundhati-b/Assembler.git
$ python Assembler/assembler.py < [input-file-name].txt
```

Output code will appear in file bin.txt

Supported instructions

Opcodes

`CLA` Clear Accumulator
`LAC` Load into Accumulator from Memory
`SAC` Store Accumulator contents in Address
`ADD` Add address contents to accumulator contents
`SUB` Subtract address contents from accumulator contents
`BRZ` Branch to address if accumulator contains zero
`BRN` Branch to address if accumulator contains negative value
`BRP` Branch to address if accumulator contains positive value
`INP` Read from terminal and put in address
`DSP` Display value in address on terminal
`MUL` Multiply accumulator and address contents
`DIV` Divide accumulator contents by address content. Quotient in R1 and remainder in R2
`STP` Stop execution

Comments

Precede the comment by `//`

```
//This is a comment
CLA // CLears Accumulator
```

Labels

Follow label name by a `:`

```
lbl1: STP
```

Symbols/Variables

Variables ***must*** be defined using `DW` followed by atmost one operand

```
I DW 22 //Symbol 'I' stored
```

Literals

Format: `'=*'`

```
ADD '=1'
```