8085 Assembler

Systems Programming Lab: Group 1

Who are we ?

1. Soumya Kanti Naskar
2. Sahil Pandey
3. Chinmoy S. Mahapatra
4. Abhirup Mondal

What is it ?

This is a X-window based GUI assembler and simulator for 8085 Assembly Language.

Object code is generated using a two pass assembly process.

Assembled code can be executed for quick testing, with full 65536 bytes of available memory.

Provision for step-wise execution for debugging is present, with register contents updated in real time and filtered display of modified memory contents.

What are the features ?

Errors are instantly reported to quickly locate and fix bugs.

The software includes a fully functional text editor with syntax highlighting, auto-indentation, copy/paste, unlimited undo/redo support to ease coding.

How to Compile

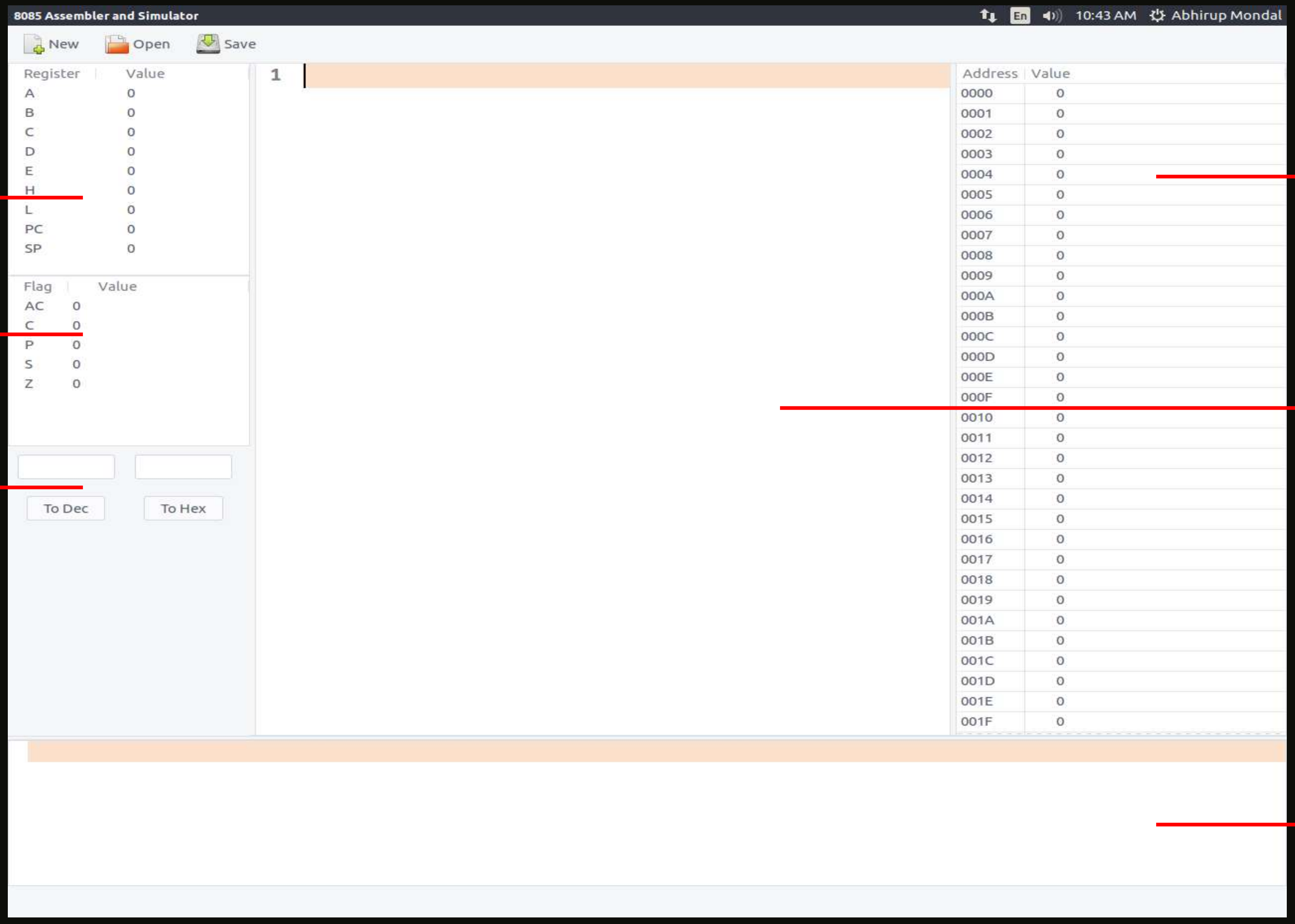
- Required to compile:

```
gtkmm-3.0-dev
pango
gtksourceviewmm-3.0-dev
g++
```

- To compile:

```
g++ -o simulator src/*.cpp src/*/*.cpp `pkg-config gtkmm-3.0 pango gtksourceview-3.0 --cflags --libs` -std=c++11

./simulator
```

The Interface



Register	Value
A	0
B	0
C	0
D	0
E	0
H	0
L	0
PC	0
SP	0

Flag	Value
AC	0
C	0
P	0
S	0
Z	0

Address	Value
0000	0
0001	0
0002	0
0003	0
0004	0
0005	0
0006	0
0007	0
0008	0
0009	0
000A	0
000B	0
000C	0
000D	0
000E	0
000F	0
0010	0
0011	0
0012	0
0013	0
0014	0
0015	0
0016	0
0017	0
0018	0
0019	0
001A	0
001B	0
001C	0
001D	0
001E	0
001F	0

Register Status

Flag Status

Conversion Buttons

Simulator’s Main Memory

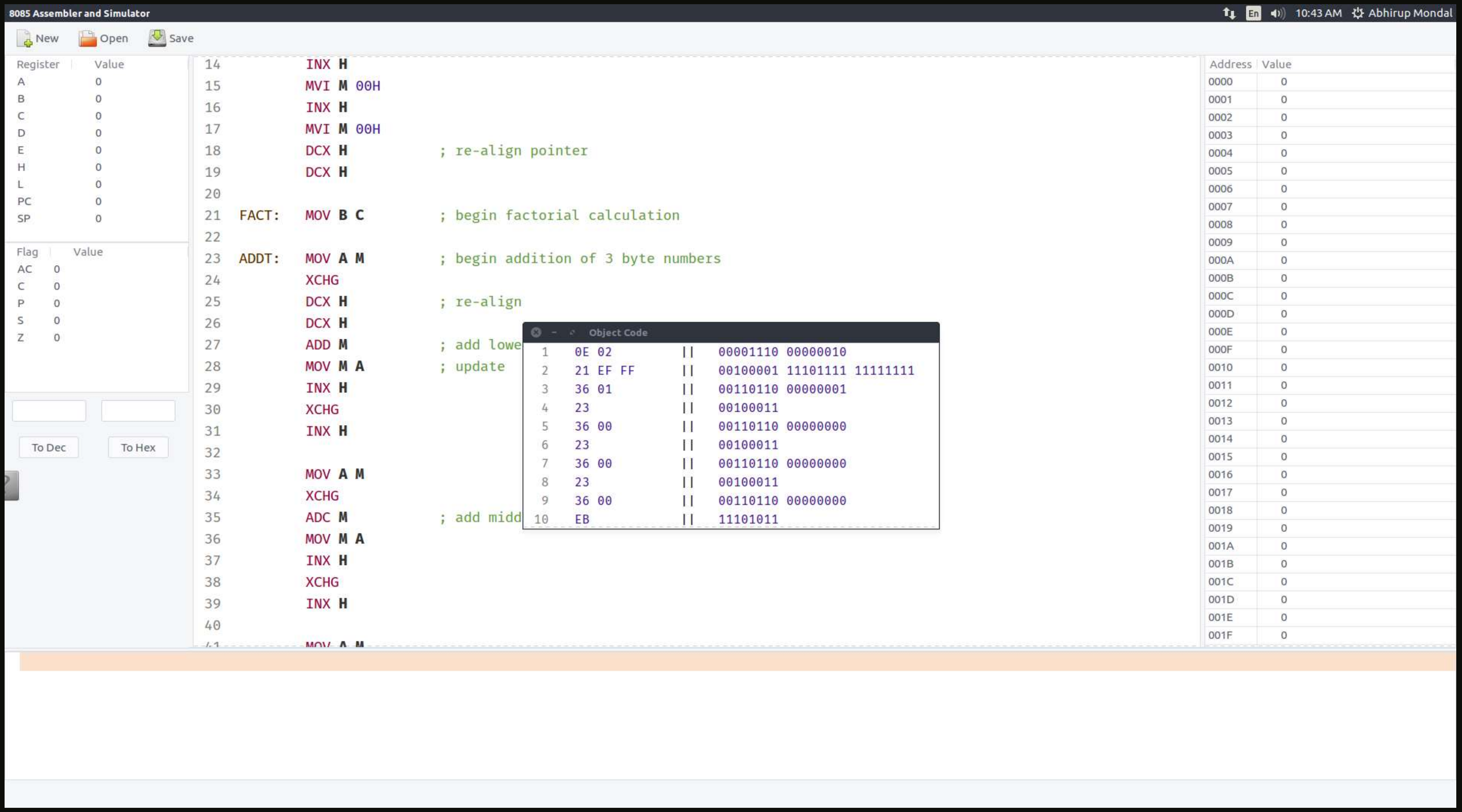
- FULL 65536 bytes
- Scrollable, Searchable
- Modifiable

Fully Functional Text Editor

- Syntax Highlighting
- Automatic Indents
- Custom Color Scheme

Assembler Messages

Object Code



8085 Assembler and Simulator

Register	Value
A	0
B	0
C	0
D	0
E	0
H	0
L	0
PC	0
SP	0

Flag	Value
AC	0
C	0
P	0
S	0
Z	0

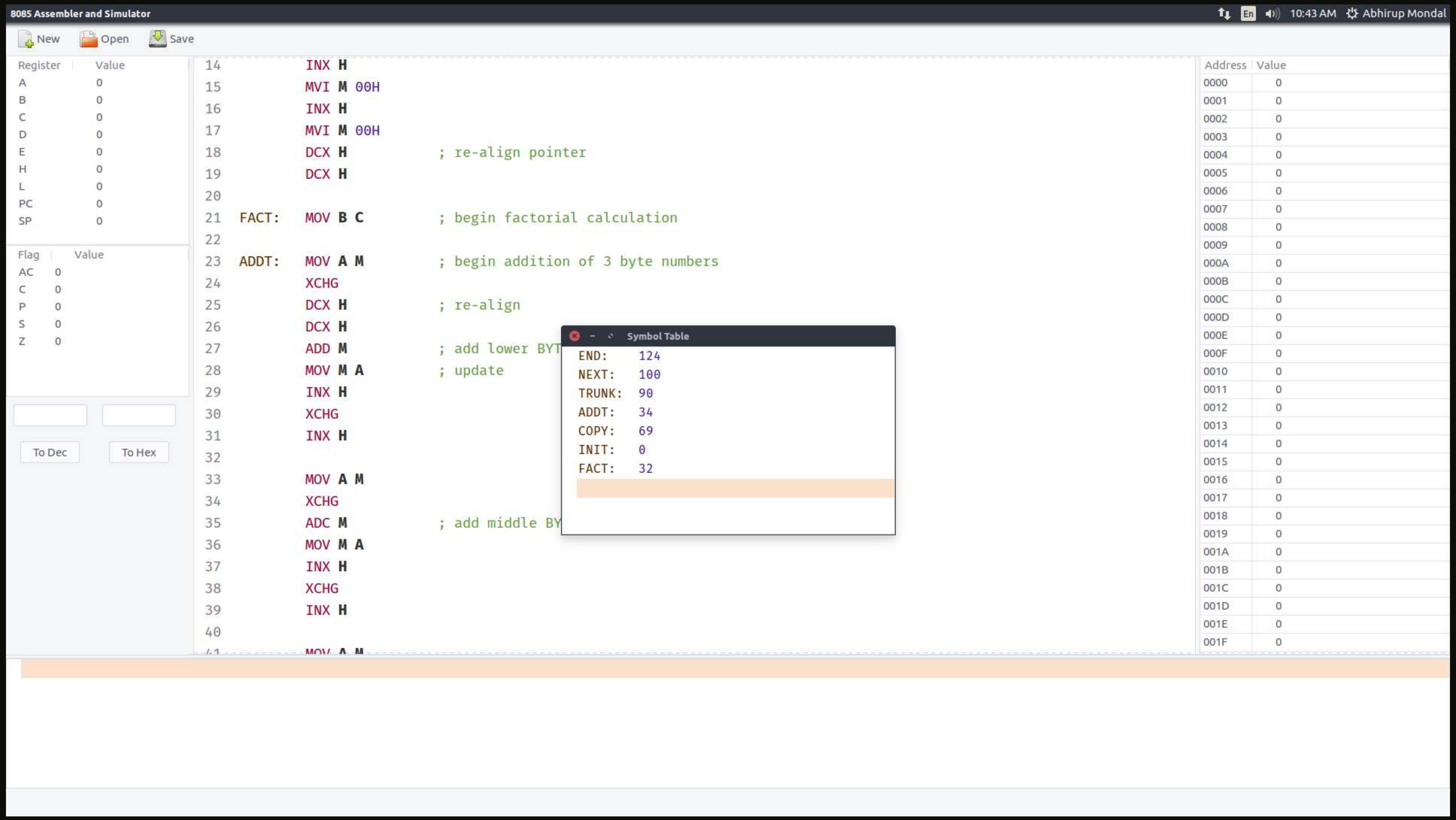
```
        INX H
        MVI M 00H
        INX H
        MVI M 00H
        DCX H           ; re-align pointer
        DCX H

FACT:   MOV B C         ; begin factorial calculation

ADDT:   MOV A M         ; begin addition of 3 byte numbers
        XCHG
        DCX H           ; re-align
        DCX H
        ADD M           ; add lowe
        MOV M A         ; update
        INX H
        XCHG
        INX H

        MOV A M
        XCHG
        ADC M           ; add midd
        MOV M A
        INX H
        XCHG
        INX H

        MOV A M
```

Object Code

```
0E 02        ||   00001110 00000010
21 EF FF     ||   00100001 11101111 11111111
36 01        ||   00110110 00000001
23           ||   00100011
36 00        ||   00110110 00000000
23           ||   00100011
36 00        ||   00110110 00000000
23           ||   00100011
36 00        ||   00110110 00000000
EB           ||   11101011
```

Address	Value
0000	0
0001	0
0002	0
0003	0
0004	0
0005	0
0006	0
0007	0
0008	0
0009	0
000A	0
000B	0
000C	0
000D	0
000E	0
000F	0
0010	0
0011	0
0012	0
0013	0
0014	0
0015	0
0016	0
0017	0
0018	0
0019	0
001A	0
001B	0
001C	0
001D	0
001E	0
001F	0

Symbol Table



Register	Value
A	0
B	0
C	0
D	0
E	0
H	0
L	0
PC	0
SP	0

Flag	Value
AC	0
C	0
P	0
S	0
Z	0

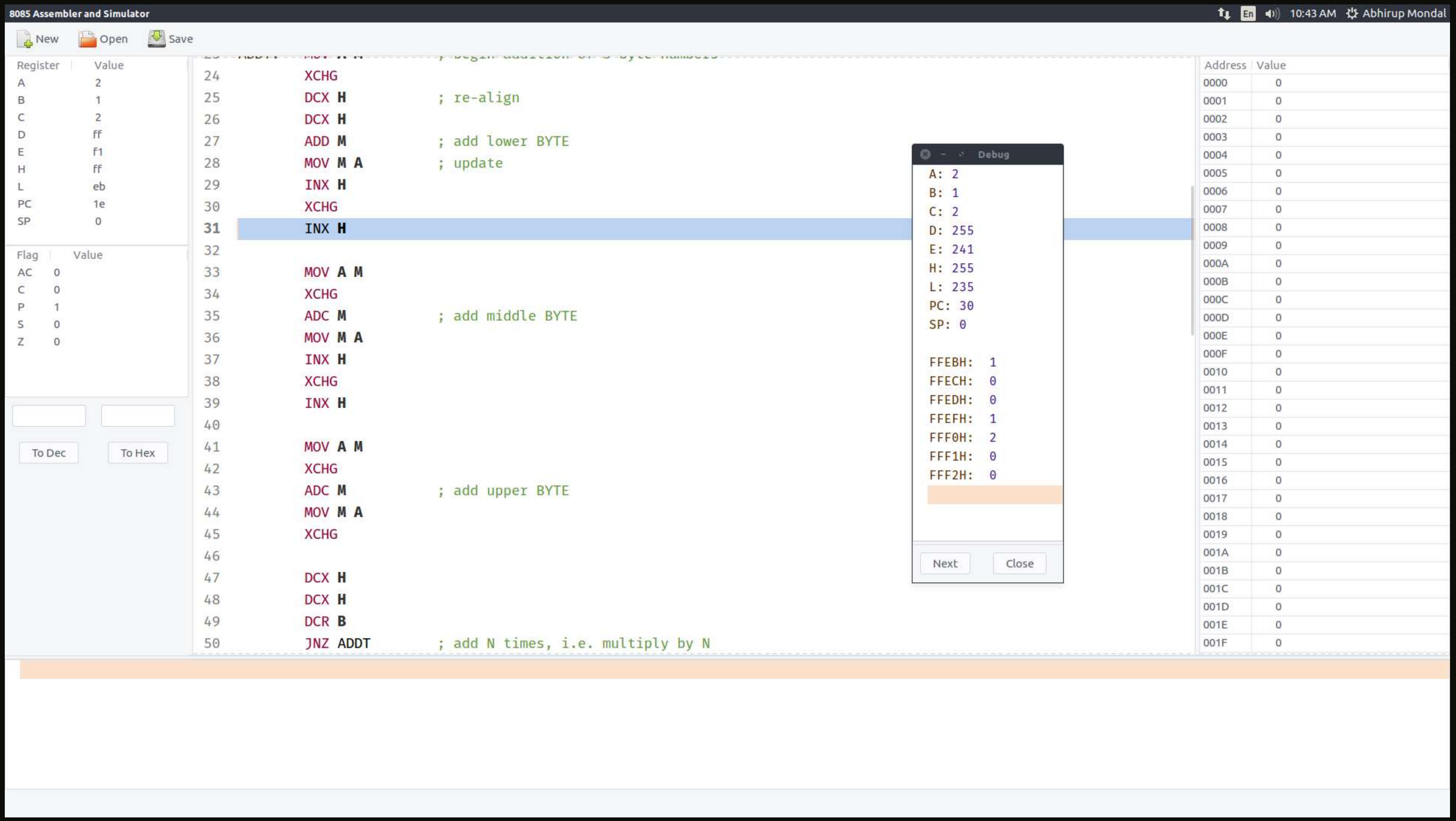
```
        INX H
        MVI M 00H
        INX H
        MVI M 00H
        DCX H           ; re-align pointer
        DCX H

FACT:   MOV B C         ; begin factorial calculation

ADDT:   MOV A M         ; begin addition of 3 byte numbers
        XCHG
        DCX H           ; re-align
        DCX H
        ADD M           ; add lower BYT
        MOV M A         ; update
        INX H
        XCHG
        INX H

        MOV A M
        XCHG
        ADC M           ; add middle BY
        MOV M A
        INX H
        XCHG
        INX H

        MOV A M
```

Symbol Table

```
END:    124
NEXT:   100
TRUNK:  90
ADDT:   34
COPY:   69
INIT:   0
FACT:   32
```

Address	Value
0000	0
0001	0
0002	0
0003	0
0004	0
0005	0
0006	0
0007	0
0008	0
0009	0
000A	0
000B	0
000C	0
000D	0
000E	0
000F	0
0010	0
0011	0
0012	0
0013	0
0014	0
0015	0
0016	0
0017	0
0018	0
0019	0
001A	0
001B	0
001C	0
001D	0
001E	0
001F	0

Debug



Register	Value
A	2
B	1
C	2
D	ff
E	f1
H	ff
L	eb
PC	1e
SP	0

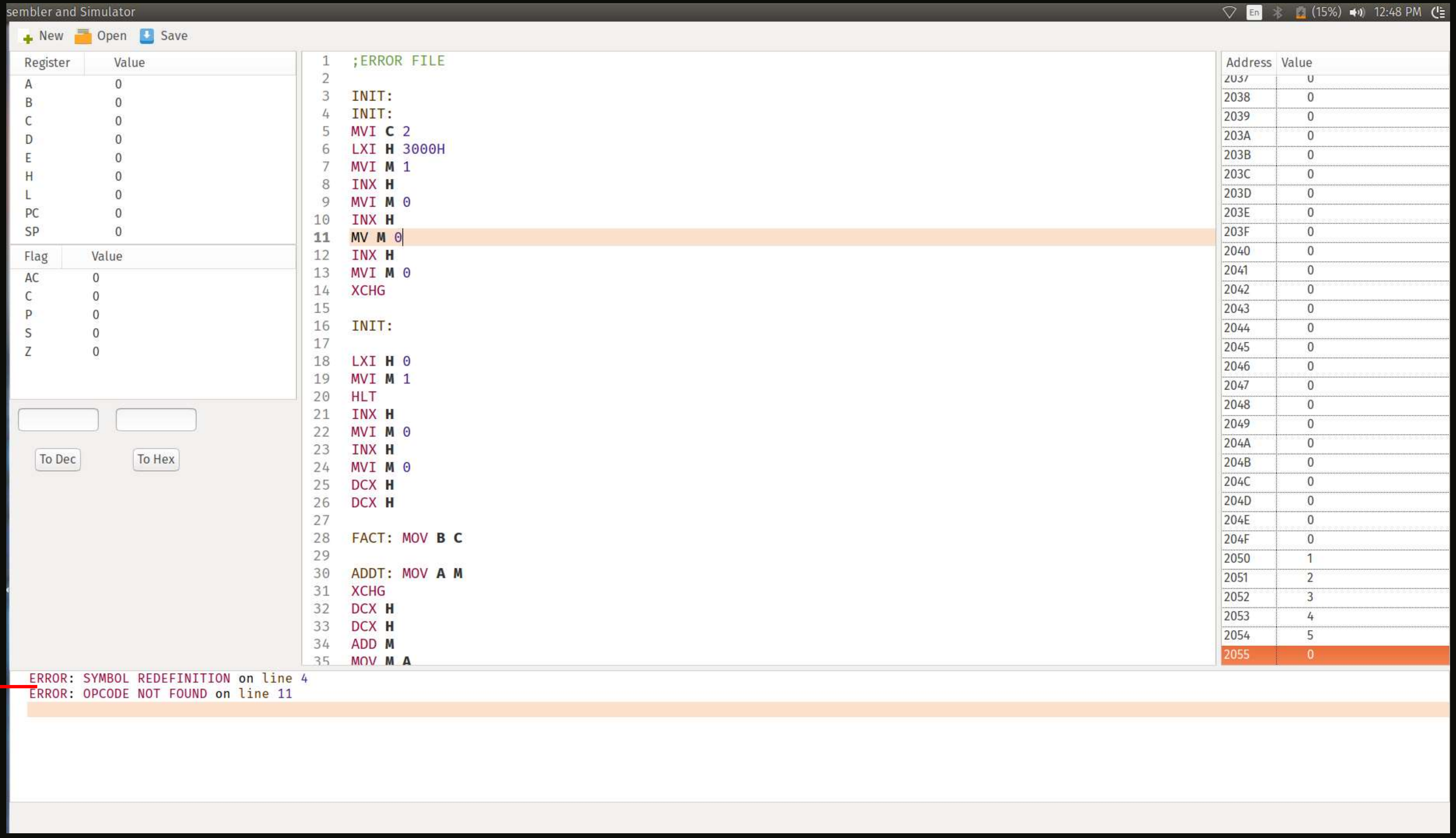
Flag	Value
AC	0
C	0
P	1
S	0
Z	0

```
        XCHG
        DCX H           ; re-align
        DCX H
        ADD M           ; add lower BYTE
        MOV M A         ; update
        INX H
        XCHG
        INX H

        MOV A M
        XCHG
        ADC M           ; add middle BYTE
        MOV M A
        INX H
        XCHG
        INX H

        MOV A M
        XCHG
        ADC M           ; add upper BYTE
        MOV M A
        XCHG

        DCX H
        DCX H
        DCR B
        JNZ ADDT        ; add N times, i.e. multiply by N
```

```
A: 2
B: 1
C: 2
D: 255
E: 241
H: 255
L: 235
PC: 30
SP: 0

FFEBH:  1
FFECH:  0
FFEDH:  0
FFEFH:  1
FFF0H:  2
FFF1H:  0
FFF2H:  0
```

Address	Value
0000	0
0001	0
0002	0
0003	0
0004	0
0005	0
0006	0
0007	0
0008	0
0009	0
000A	0
000B	0
000C	0
000D	0
000E	0
000F	0
0010	0
0011	0
0012	0
0013	0
0014	0
0015	0
0016	0
0017	0
0018	0
0019	0
001A	0
001B	0
001C	0
001D	0
001E	0
001F	0

Error Localization



Register	Value
A	0
B	0
C	0
D	0
E	0
H	0
L	0
PC	0
SP	0

Flag	Value
AC	0
C	0
P	0
S	0
Z	0

```
1   ;ERROR FILE
2
3   INIT:
4   INIT:
5   MVI C 2
6   LXI H 3000H
7   MVI M 1
8   INX H
9   MVI M 0
10  INX H
11  MV M 0
12  INX H
13  MVI M 0
14  XCHG
15
16  INIT:
17
18  LXI H 0
19  MVI M 1
20  HLT
21  INX H
22  MVI M 0
23  INX H
24  MVI M 0
25  DCX H
26  DCX H
27
28  FACT: MOV B C
29
30  ADDT: MOV A M
31  XCHG
32  DCX H
33  DCX H
34  ADD M
35  MOV M A
```

Address	Value
2037	0
2038	0
2039	0
203A	0
203B	0
203C	0
203D	0
203E	0
203F	0
2040	0
2041	0
2042	0
2043	0
2044	0
2045	0
2046	0
2047	0
2048	0
2049	0
204A	0
204B	0
204C	0
204D	0
204E	0
204F	0
2050	1
2051	2
2052	3
2053	4
2054	5
2055	0

```
ERROR: SYMBOL REDEFINITION on line 4
ERROR: OPCODE NOT FOUND on line 11
```

Error Messages