

# User Guide

## 1. Run SystemSimulator.JAR.

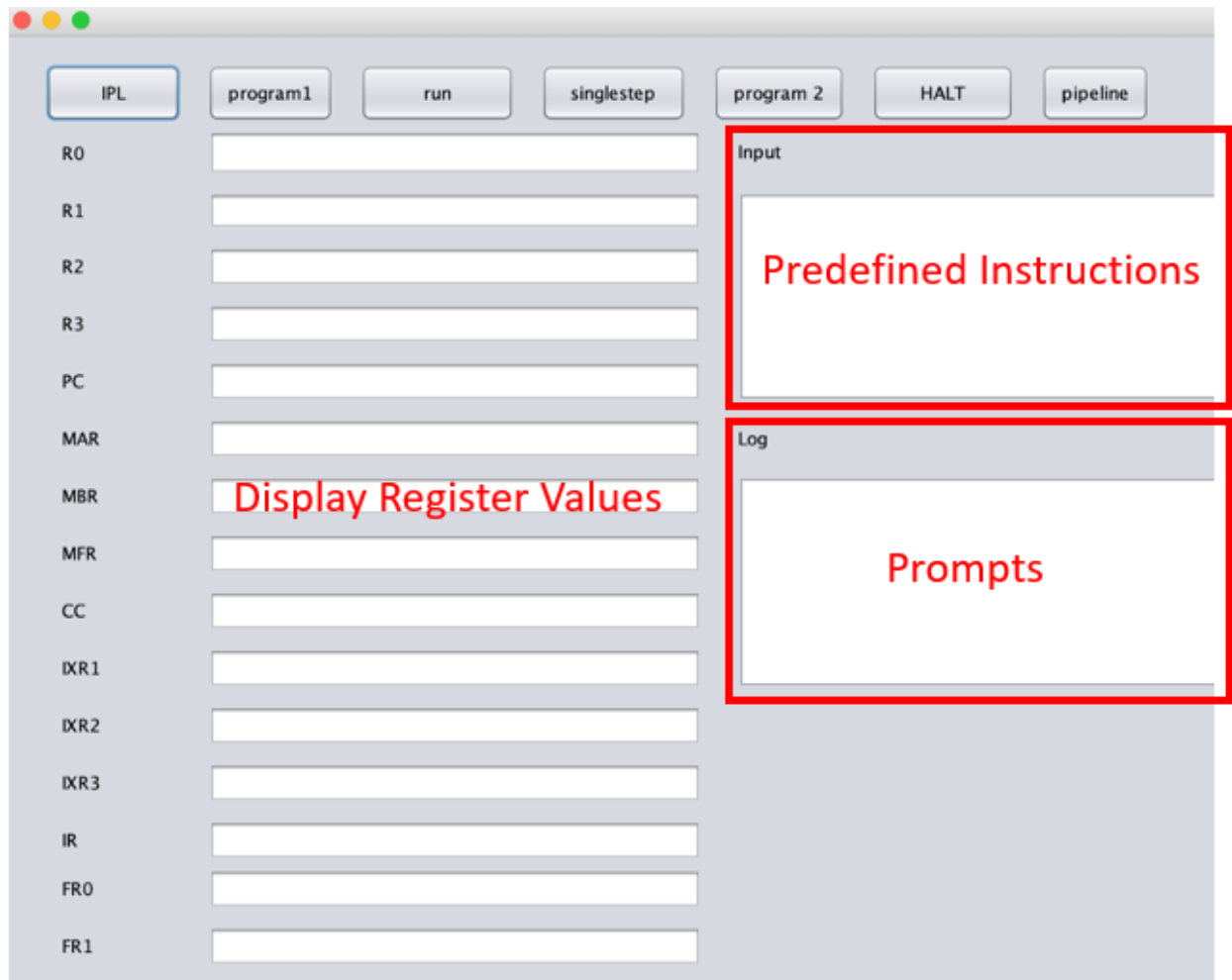


Figure 1. User Interface Popped up after Running the Program

2. **Click on IPL button to initialize the program.** After clicking on this button, the simulator will load the PC value (initialized to 6) and the program 1 which are already written. Note all the instructions and values are displayed in the format of decimal numbers. A prompt message “ipl complete” is printed in the log box. The format of the instruction follows the definition in the handout (eg. LDA 3,1,0,11, where 3 is register number, 1 is index register number, 0 is indirect addressing and 11 is address).

The user interface consists of a top control bar with buttons: **IPL** (active), **program1**, **run**, **singlestep**, **program 2**, **HALT**, and **pipeline**.

On the left, a list of registers and memory locations is shown, each with an input field:

- R0
- R1
- R2
- R3
- PC: 06
- MAR
- MBR
- MFR
- CC
- IXR1
- IXR2
- IXR3
- IR
- FR0
- FR1

On the right, the **Input** box displays the following instructions:

```
LDA 3,1,0,11
LDR 2,2,0,12
LDX 0,2,0,13
STR 1,1,0,14
STX 1,1,0,15
AMR 1,2,0,16
SMR 1,1,0,17
AIR 1,18
SIR 2,19
```

Below the input box, the **Log** box displays the message:

```
ipl complete
```

Figure 2. User Interface after Clicking on IPL Button

3. Demonstrate that all instructions (including floating point and vector operations) by clicking one of:

**Run Button:** Execute all instructions preloaded. After clicking on this button, the value of related registers including R, IX, MAR, MBR, IR, PC will be displayed in their own textbox.

The screenshot shows a software interface for a computer simulation. At the top, there are several control buttons: 'IPL', 'program1', 'run' (highlighted with a blue border), 'singlestep', 'program 2', 'HALT', and 'pipeline'. Below these buttons is a list of registers and their current values, each with a corresponding input field:

Register	Value
R0	0
R1	4
R2	0
R3	16381
PC	36
MAR	36
MBR	4753
MFR	
CC	1
IXR1	1
IXR2	50
IXR3	
IR	4753
FR0	3.1416
FR1	4

On the right side of the interface, there are two scrollable text areas:

- Input:** Contains a list of assembly instructions:

```
OUT 1,0
JCC 2,0,0,31
FADD 0,0,0,11
FSUB 0,0,0,12
LDFR 1,0,0,13
STFR 0,0,0,14
VADD 0,0,0,15
VSUB 0,0,0,22
CNVRT 0,0,0,17
```
- Log:** Contains a log of the current state:

```
IR value:4630
PC value:36
Register value(R3):2.5
MAR value:36
MBR value:4753
IR value:4753
PC value:37
```

Figure 3. User Interface after Clicking on Run Button

**SingleStep Button:** Execute one preloaded instruction at one single time. After clicking this button, you will see the PC value increase by one, and the values in MAR, MBR and other associated registers have been changed. When all the instructions are executed, nothing will happen when clicking on this button.

**Note:** If you click on the SingleStep button, you can not click on the Run button to execute the remaining instructions at once. You have to keep clicking on SingleStep button until all the instructions are executed, or HALT the program.

The screenshot shows a simulator window with a top toolbar containing buttons: IPL, program1, run, singlestep (highlighted with a blue border), program 2, HALT, and pipeline. The main area is divided into two columns. The left column lists various registers and their current values: R0 (empty), R1 (1), R2 (0), R3 (12), PC (10), MAR (9), MBR (1423), MFR (empty), CC (empty), IXR1 (empty), IXR2 (0), IXR3 (empty), IR (1423), FR0 (empty), and FR1 (empty). The right column contains two scrollable text areas. The top area, labeled 'Input', lists instructions: OUT 1,0; JCC 2,0,0,31; FADD 0,0,0,11; FSUB 0,0,0,12; LDFR 1,0,0,13; STFR 0,0,0,14; VADD 0,0,0,15; VSUB 0,0,0,22; and CNVRT 0,0,0,17. The bottom area, labeled 'Log', shows a history of values: MBR value:42303, IR value:42303, PC value:9, Register value(R1):1, MAR value:9, MBR value:1423, IR value:1423, and PC value:10.

Register	Value
R0	
R1	1
R2	0
R3	12
PC	10
MAR	9
MBR	1423
MFR	
CC	
IXR1	
IXR2	0
IXR3	
IR	1423
FR0	
FR1	

**Input:**

```
OUT 1,0
JCC 2,0,0,31
FADD 0,0,0,11
FSUB 0,0,0,12
LDFR 1,0,0,13
STFR 0,0,0,14
VADD 0,0,0,15
VSUB 0,0,0,22
CNVRT 0,0,0,17
```

**Log:**

```
MBR value:42303
IR value:42303
PC value:9
Register value(R1):1
MAR value:9
MBR value:1423
IR value:1423
PC value:10
```

Figure 4. User Interface after Clicking on SingleStep Button

#### 4. Demonstrate program 1 by clicking on:

**Program1 Button:** program 1 will be read from program1.txt and be executed. When user input is required, a window will show up. User can type an integer into the textbox each single time, which is shown in the figure below. Type in 21 integers. The outputs will be printed onto the window shows up. For instance, if user input 5, 10, 6, 4, 3, 2, 1, 12, 13, 15, 21, 23, 24, 11, 15, 5, 22, 23, 24, 20, and the final input number is 25, 25 and 24 will be printed out in order because 24 is the closest number to 25.

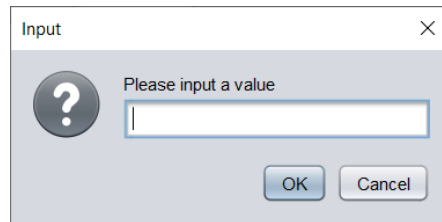


Figure 5. Pop-up Window for Input

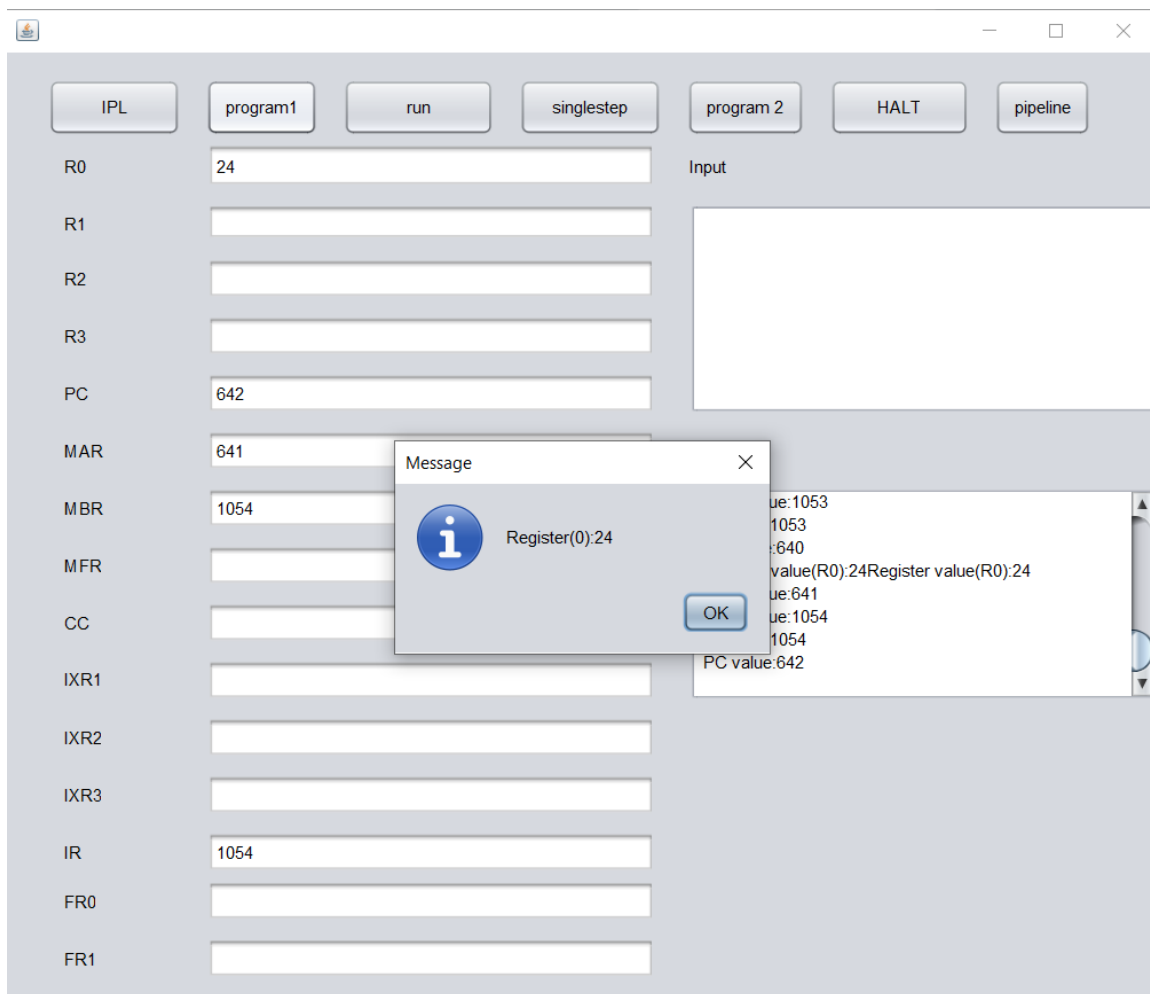


Figure 6. Pop-up Window for Output

5. Stop the simulator by clicking on:

**Halt Button:** a window will pop up and the content of all textboxes will be erased.

**Note:** Please Halt the simulator every time before you click on another button! (i.e., After running program1, click on Halt, then run program 2 or whatever you want)

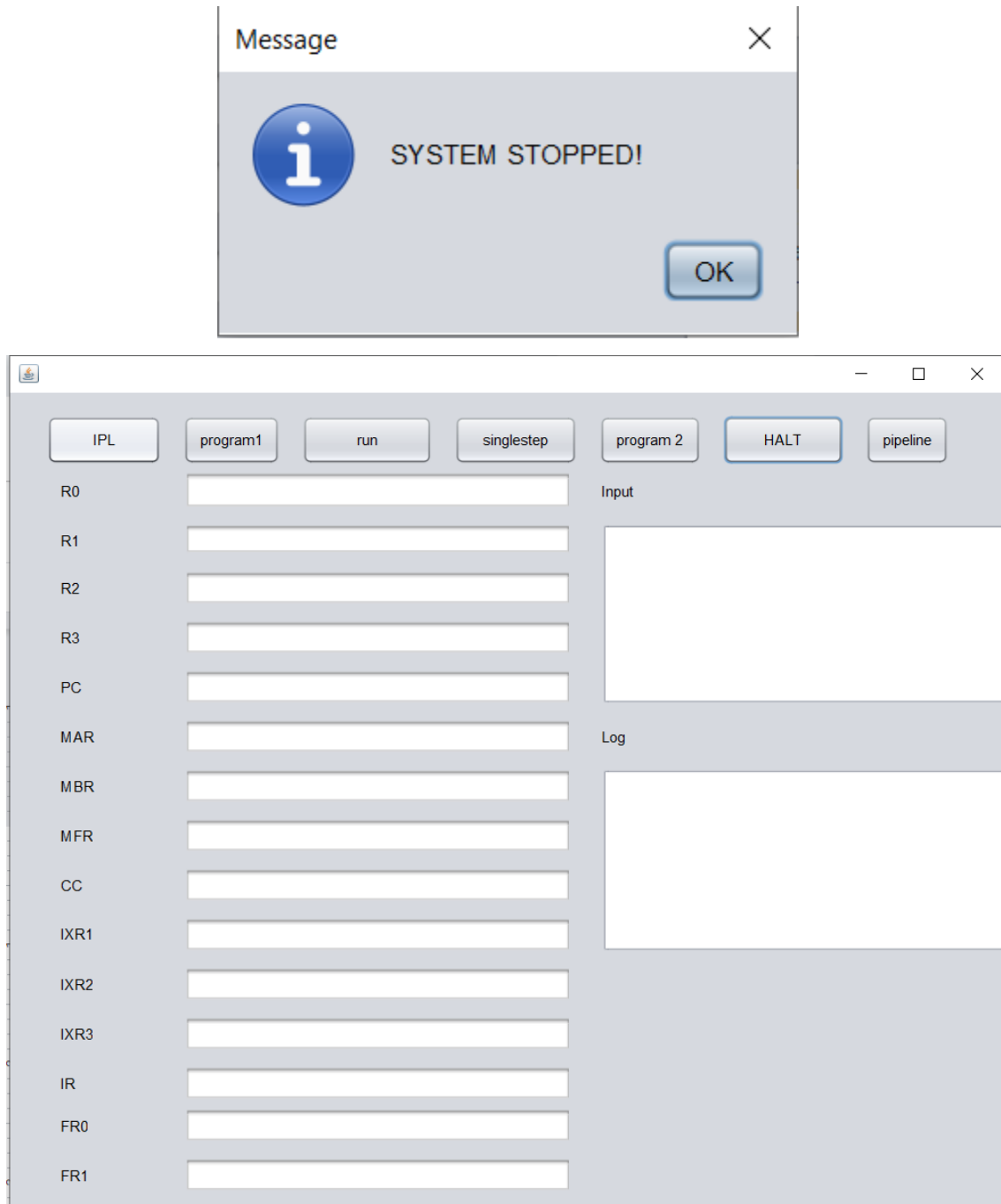


Figure 7. Halt Execution

**6. Demonstrate program 2 by clicking on:**

**Program2 Button:** program 2 will be read from program2.txt and be executed. The whole paragraph will be printed out as shown in Figure 8. When user input is required, a window will show up. User can type a word into the textbox, which is shown in Figure 9 below. For demonstration purpose we entered the word "at". After searching, the word found, the sentence number (2) and the word count (3) will be printed out in order as shown in Figure 10.

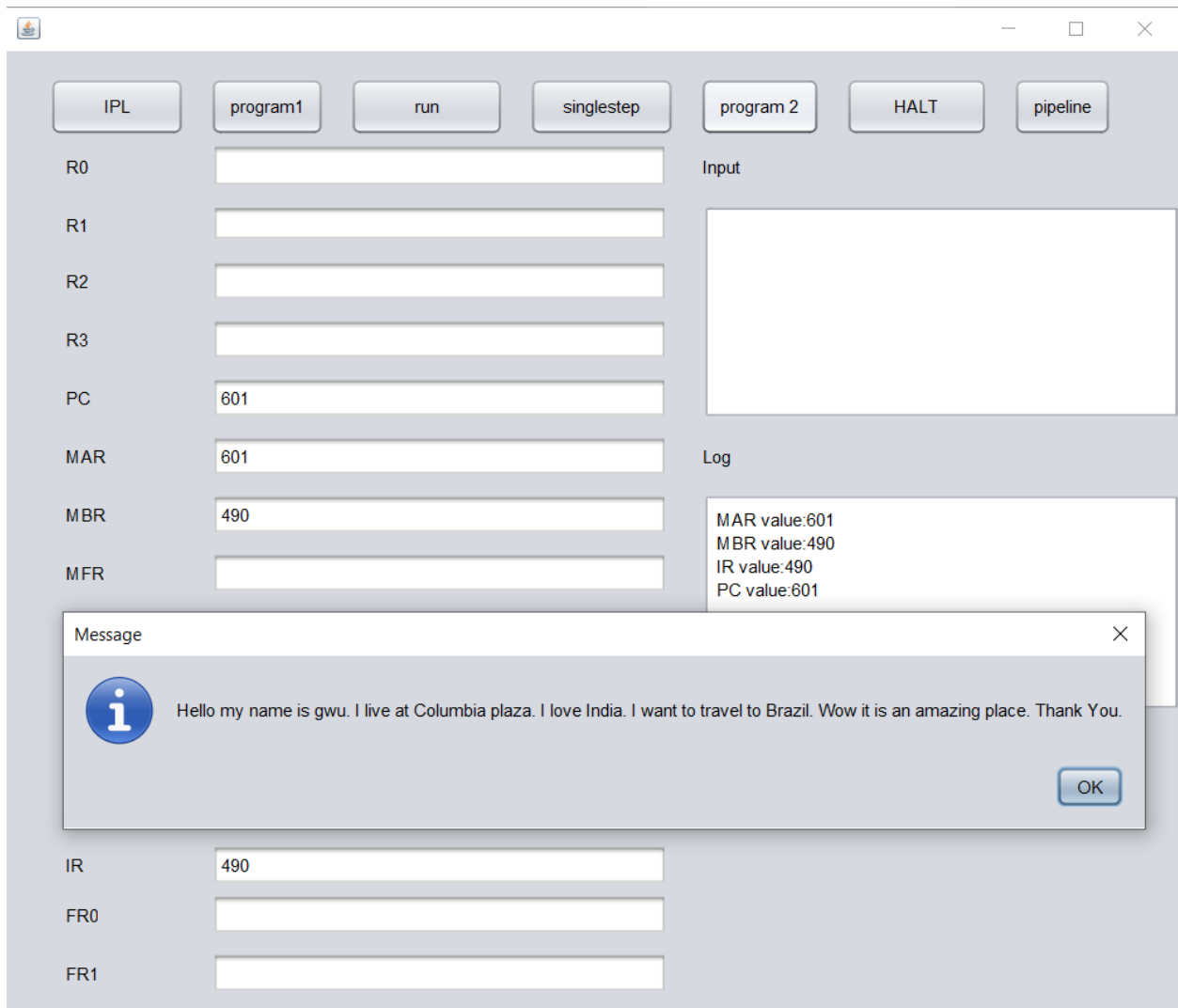


Figure 8. Paragraph Printed

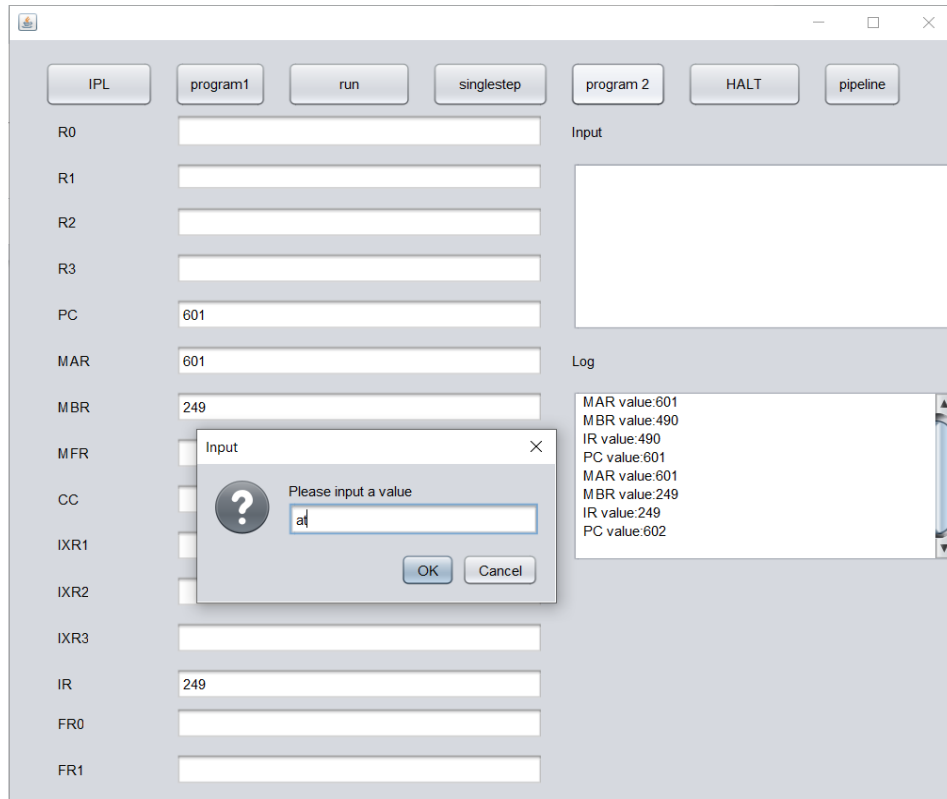
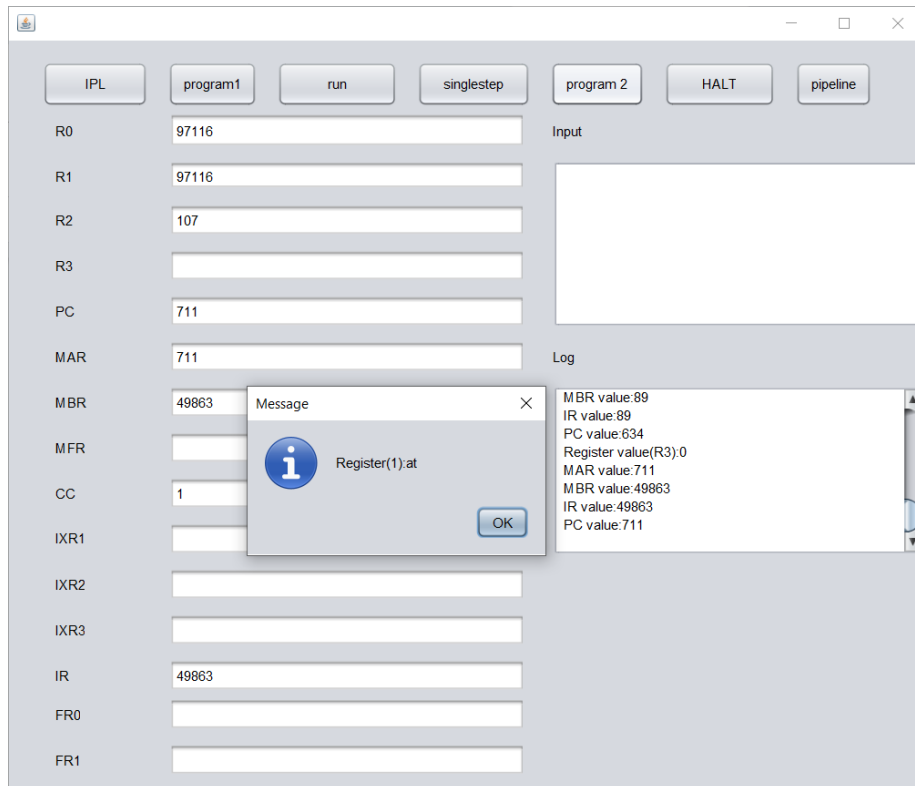
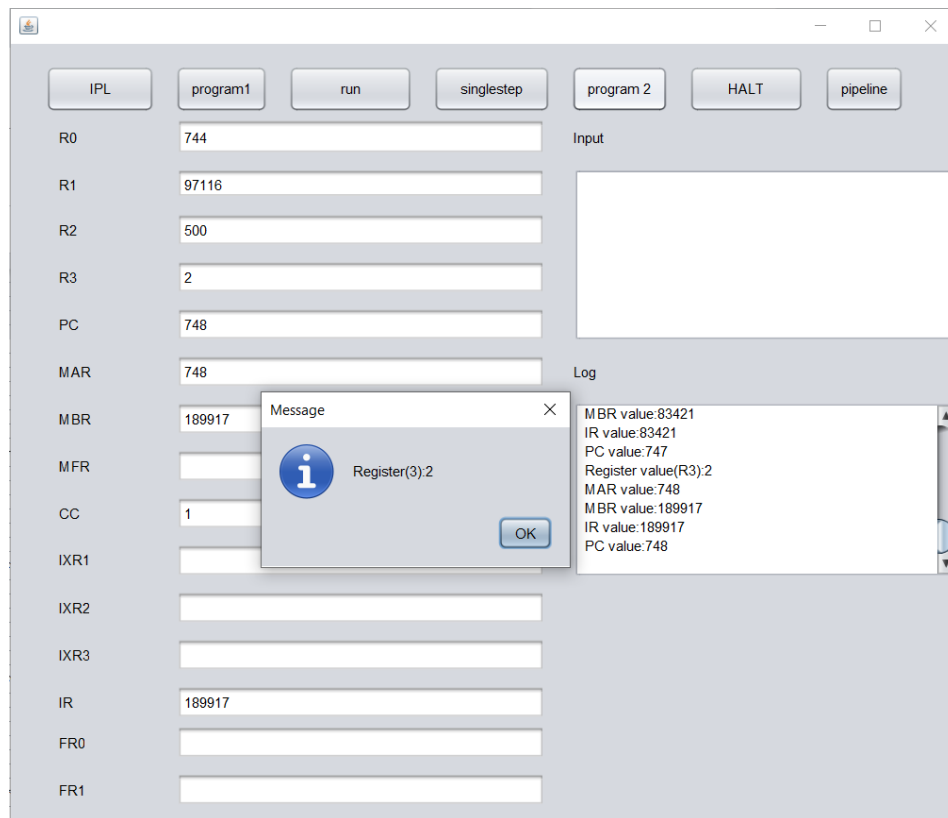


Figure 9. Ask for User Input (We entered “at” for demonstration purpose)

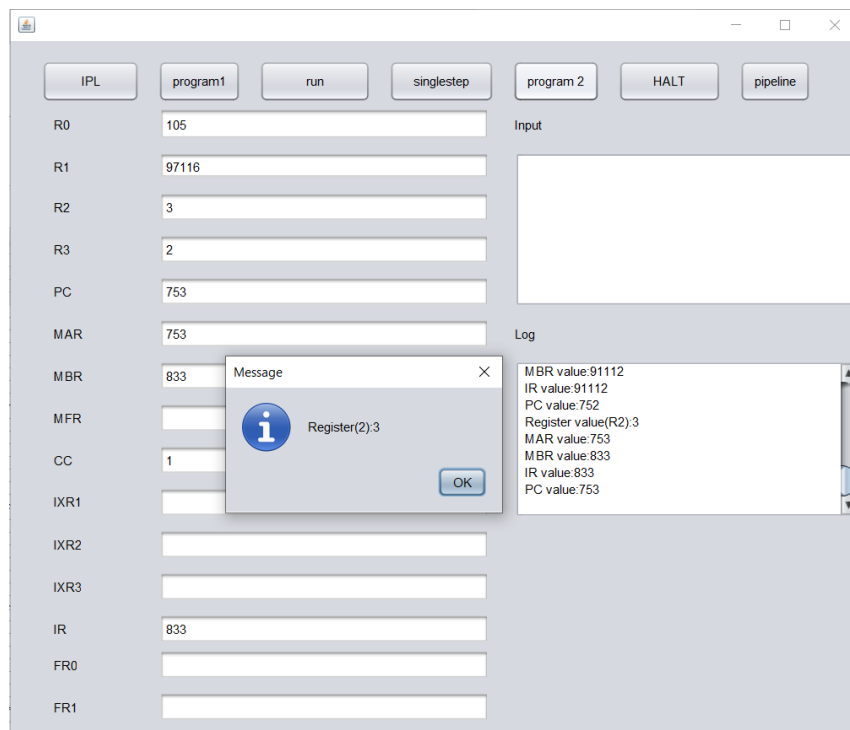


(a)





(b)



(c)

Figure 10. Outputs (a) the word found; (b) the sentence number; (3) the word count

## 7. Demonstrate pipelining by clicking on:

**Pipeline Button:** multiple instructions will be loaded into the textbox, and will be executed with pipelining. We have implemented pipeline for all instructions including floating point and vector operations. We are demonstrating five stages while executing, which will be shown in the log textbox in the form of “STR is decoding-----”, “STFR is writing back-----”, etc. After execution, “Pipelining ends” shows up in the log textbox. (It takes time for these things to show up in the log textbox! Please wait for one minute)

The screenshot shows a simulation window with a top toolbar containing buttons: IPL, program1, run, singlestep, program 2, HALT, and pipeline. The pipeline button is highlighted with a blue border. Below the toolbar, on the left, is a list of registers and their values: R0 (24), R1 (716), R2 (-19), R3 (12), PC (25), MAR (25), MBR (0), MFR (empty), CC (empty), IXR1 (1), IXR2 (empty), IXR3 (empty), IR (0), FR0 (empty), and FR1 (empty). On the right, there are two textboxes. The top one, labeled 'Input', contains a list of instructions: AMR 1,2,0,16, SMR 1,1,0,17, AIR 1,18, SIR 2,19, JZ 1,1,0,20, JNE 1,1,0,21, SOB 1,1,1,22, JGE 2,1,1,23, and JMA 1,0,24. The bottom one, labeled 'Log', shows the following text: IR value:9080, PC value:24, Register value(R0):3.1416, MAR value:25, MBR value:1753, IR value:1753, PC value:25, and Pipelining Ends-----.

Figure 11. Demonstration of pipelining

## 8. End the program.