UVSim – CS2450 Project

UVSim is a small program that acts like a simple computer. It has 100 memory slots, an accumulator (for math), and runs instructions written in a language called BasicML.

Programs are plain text files with signed 4-digit numbers. The first two digits are the command, and the last two digits are the memory location. The program starts running at memory address 00 and stops when it hits a **HALT** (43xx) instruction.

What You Need

- Python 3.8 or newer
- A terminal/command line:
  - Windows: PowerShell
  - Mac/Linux: Terminal

How to Run

1. Open your terminal.
2. Go to the folder that has main.py. Example:
   ```
   cd path/to/UVSim
   ```
3. Run UVSim with a text file:
   ```
   python3 main.py "tests/Test1.txt"
   ```

File Path Examples

- Windows:
  ```
  python main.py "C:\Users\YourName\Desktop\Test1.txt"
  ```
- Mac/Linux:
  ```
  python3 main.py "/Users/yourname/Desktop/Test1.txt"
  ```

BasicML Commands

Each instruction is +/− followed by 4 digits.

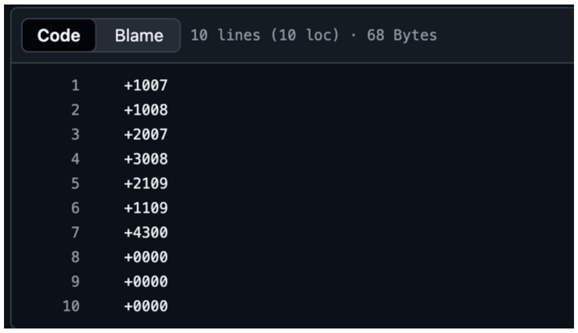
- **First 2 digits** = command
- **Last 2 digits** = memory slot (00–99)

Code	Command	What it does
10xx	READ	Ask the user for a number and save it in memory[xx]
11xx	WRITE	Print value from memory[xx]
20xx	LOAD	Load memory[xx] into the accumulator
21xx	STORE	Save the accumulator into memory[xx]

30xx	ADD	Add memory[xx] to the accumulator (truncated if too big)
31xx	SUBTRACT	Subtract memory[xx] from the accumulator (truncated if too big)
32xx	DIVIDE	Divide accumulator by memory[xx] (error if divisor = 0)
33xx	MULTIPLY	Multiply accumulator by memory[xx] (truncated if too big)
40xx	BRANCH	Jump to memory[xx]
41xx	BRANCHNEG	Jump to memory[xx] if accumulator < 0
42xx	BRANCHZERO	Jump to memory[xx] if accumulator == 0
43xx	HALT	Stop the program

Example Programs

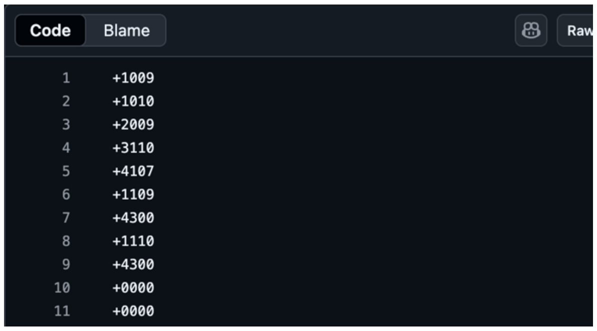
Test1.txt – Add two numbers



```
+1007
+1008
+2007
+3008
+2109
+1109
+4300
+0000
+0000
+0000
```

This program asks for two numbers, adds them, and prints the result.

Test2.txt – Print the larger number



```
+1009
+1010
+2009
+3110
+4107
+1109
+4300
+1110
+4300
+0000
+0000
```

This program asks for two numbers and prints the bigger one.

What to Expect

- If the program has a **READ** (10xx), it will stop and ask you for a number between −9999 and +9999.
- If it has a **WRITE** (11xx), it will show you the number stored at that memory spot.
- The program keeps running instructions until it reaches a **HALT** (43xx).

Errors

- **Divide by zero** → shows an error.
- **Invalid command** → shows an error.
- **Bad file format** → shows an error (must be +/− followed by 4 digits).
- **Overflow math** → numbers too big get chopped down to the last 4 digits (for example: 12345 → 2345).

Files You Can Try

- Test1.txt → Add two values
- Test2.txt → Print the larger value
- More test files are included in the repo for arithmetic, branching, input/output, and load/store.

Notes

- Always wrap file paths with quotes if there are spaces.
- Every program must start at memory[00] and end with HALT (4300).
- Empty slots can be +0000.

UVSim GUI Version – CS2450 Project

UVSim is a simple computer simulator. It lets you run programs written in BasicML using a graphical interface instead of the command line.

What You Need

- Python 3.8+
- Tkinter

How to Run

1. Open the GUI by running:
   ```
   python3 main.py
   ```
   (or whichever file starts the GUI version)
2. The main window will open. From here you can:
   - **Open File** → Pick a .txt program file (like Test1.txt).
   - **Run Program** → Executes the loaded file.
   - **Reset** → Clears memory and accumulator to start fresh.

GUI Layout

- **Loaded File** label: shows which program is open
- **Output Box**: displays results from WRITE, and commands, and shows messages like "Ready," "Running," or error messages
- **Upload box:** Button to open a file navigation page where use may select the text file they wish to upload

–

Example

1. Open Test1.txt in the GUI.
2. Click **Run Program**.
3. The GUI will ask you to enter two numbers (from the **READ** commands).
4. After you type them in, the result will appear in the **Output Box**.

BasicML Commands

The GUI runs the same commands as before.

- READ (10xx) → You'll see a popup asking for a number.
- WRITE (11xx) → The output box shows the value.
- LOAD/STORE/ADD/SUBTRACT/MULTIPLY/DIVIDE → Happen in the background.
- BRANCH/BRANCHNEG/BRANCHZERO → Control how the program jumps.
- HALT (43xx) → Stops the program.

Errors

If something goes wrong, a **popup message** or the **status bar** will explain it:

- Invalid input
- Division by zero
- Invalid or malformed instruction file

Included Test Files

- Test1.txt → Add two numbers
- Test2.txt → Print the larger number
- More tests included for branching, math, and error handling

Notes

- Programs must still start at memory[00] and end with HALT (4300).
- Instructions must be **+/− followed by 4 digits.**
- Data slots can be +0000.