**The Assembler** is used for translating programs written in the Hack symbolic machine language to executable code written in the Hack binary machine language.

Specifically, the assembler serves four purposes:

- **Writing** and editing assembly programs;
- **Translation** of programs from symbolic to binary code;
- **Visualization** of the assembly process, for learning purposes;
- **Comparison** of two binary code files, for testing the correctness of other assemblers.

## Loading, writing, saving

To load a file, click the ‘Load file’ icon in the ‘Source’ panel; The changes that you make in the editor will be saved automatically in the file, which is stored in your browser memory.

Alternatively, enter or copy-paste text into the editor panel. To save, click the ‘Download’ icon. This action saves the current text in a file named source.asm, stored in the ‘Downloads’ folder on your PC. If you want, you can rename this file and move it to another folder.

## Translating

Click the ‘Translate’ icon. The translation speed can be controlled using the speed slider. The symbol table is built on the fly (First pass: initializes the symbol table, adds the label symbols, and generates no code; Second pass: Adds variable symbols and generates code).

The translated code can be loaded into the CPU emulator. To do so, click the ‘Load to the CPU emulator’ icon in the ‘Binary Code’ panel.

## Comparing / Testing

Suppose you’ve implemented a Hack assembler, and you wish to check if it generates code correctly. To do so, have your assembler and this assembler translate the same assembly program. Next, copy-paste the code generated by your assembler into the ‘Compare code’ panel, and click the ‘Compare’ button.

## Bug / issue reports

To report a bug or propose an improvement, click the *bug* icon. You will be asked to login to your GitHub account (if you have one).

*www.nand2tetris.org*