In Section 7.6, we examined the following code as a candidate for the main function:

```c
float compute(float x, float y) {
    return x + y;
}
```

The implementation can be further modified to improve its performance and readability. Consider the following assembly code:

```machine_code
.globl compute
compute:
    pushl %ebp
    movl %esp, %ebp
    movl -8(%ebp), %eax
    movl -12(%ebp), %edx
    addl %eax, %edx
    popl %ebp
    ret
```

This approach was generated by compiling the C code from the following:

```c
#include <stdio.h>

float compute(float x, float y) {
    return x + y;
}
```

This approach is more efficient than the initial implementation because it avoids using the more expensive `addl` instruction. Instead, it uses the faster `add` instruction, which adds two 32-bit integers.

We obtained this implementation using a tool that automatically translates C code into machine code. The tool optimizes the resulting code by replacing the `addl` instruction with the `add` instruction, which is faster on modern processors.