- CPSC 275: Introduction to Computer Systems

CPSC 275: Introduction to Computer Systems

Fall 2025

- Syllabus
- Schedule
- Resources
- Upload
- Solution

Homework 16

NOTE: You are not required to hand in the following exercises, but you are strongly encouraged to complete them to strengthen your understanding of the concepts covered in class.

1. When given the C code

```
void cond(int a, int *p)
{
   if (p && a > 0)
      *p += a;
}
```

The `gcc` C compiler generates the following assembly code for the body of the function:

```
# a in %edx, p in %eax
   testl %eax, %eax
   je    .L3
   testl %edx, %edx
   jle   .L3
   addl  %edx, (%eax)
.L3:
```

   A. Write a `goto` version in C that performs the same computation.
   B. Explain why the assembly code contains two conditional branches, even though the C code has only one `if` statement.

2. An alternate rule for translating `if` statements into `goto` code is as follows:

```
   t = test-expr;
   if (t)
      goto true;
   else-statement
   goto done;
true:
   then-statement
done:
```

Rewrite the `goto` version of `diff` (see Lecture 16) based on this alternate rule.

3. Assuming that `n`, `low`, and `high` are in `%ebx`, `%ecx`, and `%edx`, respectively, write assembly code for the following function:

```
int between(int n, int low, int high) {
    if (n >= low && n <= high)
        return 1;
    else
        return 0;
}
```

4. Assuming that `a`, `b`, and `c` are in `%ebx`, `%ecx`, and `%edx`, respectively, write assembly code for the following function:

   ```
   int min3(int a, int b, int c) {
       if (a <= b && a <= c)
           return a;
       else if (b <= a && b <= c)
           return b;
       else
           return c;
   }
   ```

- **Welcome: Sean**
    - LogOut

Trinity College
HARTFORD CONNECTICUT