

Submit answers by committing a file into the hw5 sub-directory of your CNETID-cs154-aut-19 svn repository. Do not create this directory yourself: an “svn update” at the top level of your checkout will create it. Your answer file should be either hw5.txt or hw5.pdf for answers in plain ASCII text, or PDF, respectively. PDFs of scanned hand-written pages must not exceed 6 megabytes. No other file formats, or filenames are acceptable, and no files besides hw5.txt or hw5.pdf will be graded. Not following directions will result in losing points.

**(1)** (20 points)

<pre>1 void addto(long *arr, long *pb, long n) { 2     long i; 3     for (i = 0; i &lt; n; i++) { 4         arr[i] += *pb; 5     } 6 }</pre>	<pre>1 # n in %rdx; pb in %rsi; arr in %rdi 2 movq \$0, %rax # i = 0 3 .L3: 4 movq (%rsi), %rcx 5 addq %rcx, (%rdi,%rax,8) 6 addq \$1, %rax 7 cmpq %rdx, %rax 8 jne .L3</pre>
--	---

The intended purpose of the C function on the left is to add some (single, unchanging) value `*pb` to all elements of `arr`. Assume for now that there is a reason to be passing the value indirectly, via a pointer. The assembly generated with high optimization (“-O3”) is on the right. Only the assembly for the loop is shown, with cosmetic changes and some explanatory comments.

- A. Line 4 of the assembly corresponds to what C expression on what line of the C code above?
  
- B. Suppose that, with the intent of optimizing the code, lines 3 and 4 of the assembly are switched, so “`movq (%rsi), %rcx`” is done once, before the loop. Describe a scenario in which that “optimization” actually **changes** the effect of calling `addto` (so the transformation is in fact not a valid optimization).
  
- C. What is the general name for this scenario? Chapter 5 describes this and other “optimization blockers”.
  
- D. How would you re-write `addto` (in C) to achieve the same effect of the re-ordering discussed here?