

3.9 The C loop is

```
while (save[i] == k)
    i = i + j;
```

with i , j , and k corresponding to registers $\$s3$, $\$s4$, and $\$s5$ and the base of the array save in $\$s6$. The assembly code given in the example is

Code before

```
Loop: add $t1, $s3, $s3    # Temp reg $t1 = 2 * i
      add $t1, $t1, $t1    # Temp reg $t1 = 4 * i
      add $t1, $t1, $s6    # $t1 = address of save [i]
      lw  $t0, 0($t1)     # Temp reg $t0 = save[i]
      bne $t0, $s5, Exit  # go to Exit if save[i] ≠ k
      add $s3, $s3, $s4    # i = i + j
      j  Loop             # go to Loop
Exit:
```

Number of instructions executed if $\text{save}[i + m * j]$ does not equal k for $m = 10$ and does equal k for $0 \leq m \leq 9$ is $10 \times 7 + 5 = 75$, which corresponds to 10 complete iterations of the loop plus a final pass that goes to `Exit` at the `bne` instruction before updating i . Straightforward rewriting to use at most one branch or jump in the loop yields

Code after

```
      add $t1, $s3, $s3    # Temp reg $t1 = 2 * i
      add $t1, $t1, $t1    # Temp reg $t1 = 4 * i
      add $t1, $t1, $s6    # $t1 = address of save[i]
      lw  $t0, 0($t1)     # Temp reg $t0 = save[i]
      bne $t0, $s5, Exit  # go to Exit if save[i] ≠ k
Loop: add $s3, $s3, $s4    # i = i + j
      add $t1, $s3, $s3    # Temp reg $t1 = 2 * i
      add $t1, $t1, $t1    # Temp reg $t1 = 4 * i
      add $t1, $t1, $s6    # $t1 = address of save[i]
      lw  $t0, 0($t1)     # Temp reg $t0 = save[i]
      beq $t0, $s5, Loop  # go to Loop if save[i] = k
Exit:
```

The number of instructions executed by this new form of the loop is $5 + 10 \times 6 = 65$. If $4 \times j$ is computed before the loop, then further saving in the loop body is possible.

Code after further improvement

```
      add $t2, $s4, $s4    # Temp reg $t2 = 2 * j
      add $t2, $t2, $t2    # Temp reg $t2 = 4 * j
      add $t1, $s3, $s3    # Temp reg $t1 = 2 * i
      add $t1, $t1, $t1    # Temp reg $t1 = 4 * i
```

```

    add $t1, $t1, $s6    # $t1 = address of save[i]
    lw  $t0, 0($t1)     # Temp reg $t0 = save[i]
    bne $t0, $s5, Exit  # go to Exit if save[i] ≠ k
Loop: add $t1, $t1, $t2  # $t1 = address of save [i + m * j]
    lw  $t0, 0($t1)     # Temp reg $t0 = save[i]
    beq $t0, $s5, Loop  # go to Loop if save[i] = k
Exit:

```

The number of instructions executed is now $7 + 10 \times 3 = 37$.