Submission 8

8.1.2

```
0000000000000000 <copy_c>:
   0:   a9401807        ldp     x7, x6, [x0]
   4:   a9411005        ldp     x5, x4, [x0, #16]
   8:   a9420803        ldp     x3, x2, [x0, #32]
   c:   f9401800        ldr     x0, [x0, #48]
  10:   a9001827        stp     x7, x6, [x1]
  14:   a9011025        stp     x5, x4, [x1, #16]
  18:   a9020823        stp     x3, x2, [x1, #32]
  1c:   f9001820        str     x0, [x1, #48]
  20:   d65f03c0        ret
```

0-c: copies first 7 elements into temp variables; uses ldp to save instructions copying 2 values at once
10-1c: writes the temp variables into the array b; uses stp to save instructions copying 2 values at once
20: returns

8.2.2

```
0000000000000000 <add_c>:
   0:   b4000120        cbz     x0, 24 <add_c+0x24>
   4:   d2800004        mov     x4, #0x0                        // #0
   8:   f8647825        ldr     x5, [x1, x4, lsl #3]
   c:   f8647846        ldr     x6, [x2, x4, lsl #3]
  10:   8b0600a5        add     x5, x5, x6
  14:   f8247865        str     x5, [x3, x4, lsl #3]
  18:   91000484        add     x4, x4, #0x1
  1c:   eb04001f        cmp     x0, x4
  20:   54ffff41        b.ne    8 <add_c+0x8>  // b.any
  24:   d65f03c0        ret
```

0: if (i_n_value == 0) returns [otherwise checks at the end of loop]
4: i = 0
8-14: load variables of a[i] / b[i] (lsl #3 because 64bit) into temp registers, to then add them together in the first (temp1 = temp1 + temp2) and writes it in c[i]
18: i = i+1
1c-24: repeats loop while i<i_n_values, otherwise returns