1

```
        addi x7, x0, 0
loop1:  addi x29, x0, 0          # j = 0
loop2:  slli x28, x29, 5         # 5 = 2 + 3
        add  x28, x10, x28      # x28 = &D[4*j]
        add  x30, x7, x29       # x30 = i + j
        sd   x30, 0(x28)        # D[4*j] = i + j
        addi x29, x29, 1         # j++
        blt  x29, x6, loop2     # j < b
        addi x7, x7, 1           # i++
        blt  x7, x5, loop1      # i < a
```

2

```
fib:    addi x5, x0, 2       # x5 = 2
        blt  x10, x5, re     # n == 0 or n == 1, directly return
        addi sp, sp, -16     # save space
        sd   x1, 8(sp)       # save return address
        addi x5, x10, -1     # x5 = n - 1
        sd   x5, 0(sp)       # save n - 1
        jal  x1, fib         # call fib(n-1)
        addi x6, x10, 0      # x6 = fib(n-1)
        ld   x5, 0(sp)       # x5 = n - 1
        addi x5, x5, -1      # x5 = n - 2
        jal  x1, fib         # call fib(n-2)
        add  x10, x10, x6    # x10 = fib(n-1)+fib(n-2)
        ld   x1, 8(sp)
        addi sp, sp, 16      # pop stack
re:     jalr x0, 0(x1)
```