

ARMAssemblyInt:BigEx:D

main first half

```
014 stmfid sp!, {fp, lr}
015 add fp, sp, #4
016 sub sp, sp, #56
017 mov r3, #320
018 str r3, [fp, #-20]
019 mov r3, #0
020 str r3, [fp, #-8]

021 mov r3, #0
022 str r3, [fp, #-16]
023 b .L2
024 .L3:

025 ldr r3, [fp, #-16]
026 mov r3, r3, asl #2

028 add r3, sp, r3
029 mov r2, #0
030 str r2, [r3]

031 ldr r3, [fp, #-16]
032 add r3, r3, #1
033 str r3, [fp, #-16]
034 .L2:
035 ldr r3, [fp, #-16]
036 cmp r3, #9
037 ble .L3

038 mov r3, #2
039 str r3, [fp, #-12]
040 b .L4
041 .L6:
042 ldr r2, [fp, #-12]
043 sub r3, fp, #60
044 mov r0, r2
045 mov r1, r3
046 bl checkNum
047 mov r3, r0
048 cmp r3, #0
049 beq .L5
050 ldr r3, [fp, #-8]
051 add r3, r3, #1
052 str r3, [fp, #-8]
053 .L5:
```

```
const int numsToTest = 320;
unsigned primeSet[10]; // Space for 320 bits
int totalPrimes = 0, toTest, idx;

for (idx = 0; idx < 10; idx++)
    primeSet[idx] = 0;

for (toTest = 2; toTest < numsToTest; toTest++)
    if (checkNum(toTest, primeSet))
        totalPrimes++;
```

