Name: Breona Pizzuta

Partner (if any): Ben Carpenter

Pledge: "I pledge my honor that I have abided by the Stevens Honor System."

#### CS 382 Lab 4 Task 2

Start by using the b \_start command:

```
[ Register Values Unavailable ]
 -lab4.s-
b+>
               MOV X0, 0 //Set register X0 to 0
       10
               MOV X1, 0 //Set register X1 to 0
       11
       12
       13
               ADR X2, vec1 //Loads vec1 into register X2
                              Loads vec2 into register X3
       14
               ADR X3, vec2
       15
               ADR X4, dot //Loads dot into register X4
       16
                                  Load the first element of vec1
       17
               LDR X5.
                        [X2, 0] /
       18
               LDR X6, [X3, 0] //Load the first element of vec2
       19
                                  Multiplies X5 and X6 storing in X7
       20
               MUL X7.
                       X5, X6 /
                                  Dest X0 add X0 and X7
       21
               ADD X0, X0, X7
       22
remote Thread 1.5002 In: start
(gdb) b _start
Breakpoint 1 at 0x4000b0: file lab4.s, line 10.
(gdb)
```

# Step through the first 19 lines:

Here we are loading in the first elements of our vectors.

```
x0
                  0x0
                                         0
 x1
                  0x0
                                         0
 х2
                  0x410104
                                         4260100
                  0x41011c
                                         4260124
 х3
 x4
                  0x410134
                                         4260148
 х5
                  0xa
                                         10
 хб
                  0x1
                                         1
                  0x0
                                         0
 x7
                                         0
 x8
                  0x0
 x9
                  0x0
                                         0
 x10
                  0x0
                                         0
                  0x0
                                         0
 x11
 x12
                  0x0
                                         0
  -lab4.s
         10
                                Set register X0 to 0
                  MOV X0, 0 /
         11
                  MOV X1, 0 //Set register X1 to 0
         12
                  ADR X2, vec1 //Loads vec1 into register X
ADR X3, vec2 //Loads vec2 into register X
ADR X4, dot //Loads dot into register X4
         13
                                   Loads vec1 into register X2
                                  /Loads vec2 into register X3
         14
         15
         16
         17
                  LDR X5, [X2, 0] //Load the first element of vec1
                  LDR X6, [X3, 0] //Load the first element of vec2
         18
         19
                  MUL X7, X5, X6 //Multiplies X5 and X6 storing in X7 ADD X0, X0, X7 //Dest X0 add X0 and X7
         20
         21
         22
                                                                                                PC: 0x40000
remote Thread 1.5002 In: start
                                                                                         L20
Breakpoint 1 at 0x4000b0: file lab4.s, line 10.
(gdb) step
=> 0x000000000004000b4 <_start+4>:
                                             01 00 80 d2
                                                                         x1, #0x0
                                                                mov
/ #0
(gdb) step
=> 0x000000000004000b8 <_start+8>:
                                             62 02 08 10
                                                                         x2, 0x410104
                                                                adr
(gdb) step
=> 0x00000000004000bc <_start+12>:
                                             03 03 08 10
                                                                         x3, 0x41011c
                                                                adr
(gdb) step
=> 0x000000000004000c0 <_start+16>:
                                             a4 03 08 10
                                                                         x4, 0x410134
                                                                adr
(gdb) step
=> 0x000000000004000c4 < start+20>:
                                             45 00 40 f9
                                                                ldr
                                                                         x5, [x2]
(gdb) step
=> 0x00000000004000c8 <_start+24>:
                                             66 00 40 f9
                                                                         x6, [x3]
                                                                ldr
(gdb) step
           0000004000cc <_start+28>:
                                             a7 7c 06 9b
                                                                mul
                                                                         x7, x5, x6
(gdb)
```

## Step through next 2 lines:

Here we store the multiplication and add the product so far

```
—Register group: general—
x0
               0xa
                                   10
x1
               0x0
x2
               0x410104
                                   4260100
х3
               0x41011c
                                   4260124
x4
               0x410134
                                   4260148
x5
               0xa
                                   10
хб
               0x1
                                   1
x7
               0xa
                                   10
x8
               0x0
                                   0
x9
               0x0
                                   0
x10
               0x0
                                   0
x11
               0x0
                                   0
x12
               0x0
                                   0
-lab4.s
       16
               LDR X5, [X2, 0] /
                                 Load the first element of vec1
                                 Load the first element of vec2
       18
               LDR X6, [X3, 0]
       19
               MUL X7, X5, X6 /
                                 /Multiplies X5 and X6 storing in X7
       20
       21
               ADD X0, X0, X7
                                 Dest X0 add X0 and X7
       22
       23
       24
               LDR X8, [X2,8] //Load the second element of vec1
       25
               LDR X9, [X3,8] //Load the second element of vec2
       26
       27
                                 Multiplies X5 and X6 storing in X7
       28
               ADD X0, X0, X7
                                 /Dest X0 add X0 and X7
                                                                                    PC: 0x46
emote Thread 1.5141 In: start
                                                                              L24
gdb) step
                                                                x2, 0x410104
> 0x00000000004000b8 <_start+8>: 62 02 08 10
                                                        adr
gdb) step
> 0x000000000004000bc < start+12>: 03 03 08 10
                                                        adr
                                                                x3, 0x41011c
gdb) step
                                                                x4, 0x410134
> 0x000000000004000c0 < start+16>:
                                     a4 03 08 10
                                                        adr
gdb) step
> 0x00000000004000c4 <_start+20>:
                                                                x5, [x2]
                                       45 00 40 f9
                                                        ldr
gdb) step
> 0x000000000004000c8 <_start+24>:
                                     66 00 40 f9
                                                                x6, [x3]
                                                        ldr
gdb) step
> 0x000000000004000cc <_start+28>:
                                       a7 7c 06 9b
                                                        mul
                                                                x7, x5, x6
gdb) step
> 0x00000000004000d0 <_start+32>:
                                       00 00 07 8b
                                                        add
                                                                x0, x0, x7
gdb) step
:> 0x000000000004000d4 <_start+36>:
                                       48 04 40 f9
                                                        ldr
                                                                x8, [x2, #8]
gdb)
```

### Step through the next 2 lines:

Here we are loading in the 2nd elements of our vectors.

```
-Register group: general-
                                    10
x0
x1
               0x0
                                    0
x2
                                    4260100
               0x410104
х3
               0x41011c
                                    4260124
х4
               0x410134
                                    4260148
х5
               0xa
                                    10
хб
               0x1
                                    1
x7
               0xa
                                    10
x8
               0x14
                                    20
х9
               0x2
                                    2
x10
                                    0
               0x0
x11
               0x0
                                    0
x12
               0x0
                                    0
-lab4.s-
       22
       23
       24
               LDR X8, [X2,8] //Load the second element of vec1
       25
               LDR X9,
                         X3,8]
                                /Load the second element of vec2
       26
       27
               MUL X7, X8, X9 //Multiplies X5 and X6 storing in X7
       28
               ADD X0, X0, X7 //Dest X0 add X0 and X7
       29
       30
               LDR X10, [X2,16] //Load the third element of vec1
       31
               LDR X11, [X3,16] //Load the thrid element of vec2
       32
               MUL X7, X10, X11 //Multiplies X5 and X6 stored in X7
       33
               ADD X0, X0, X7 //Dest X0 add X0 and X7
       34
emote Thread 1.5141 In: start
                                                                                L27
gdb) step
> 0x00000000004000c0 <_start+16>:
                                                                 x4, 0x410134
                                        a4 03 08 10
                                                         adr
gdb) step
:> 0x0000000000004000c4 < start+20>:
                                        45 00 40 f9
                                                         ldr
                                                                 x5, [x2]
gdb) step
:> 0x000000000004000c8 <_start+24>:
                                        66 00 40 f9
                                                         ldr
                                                                 x6, [x3]
gdb) step
> 0x000000000004000cc <_start+28>:
                                                                 x7, x5, x6
                                        a7 7c 06 9b
                                                         mul
gdb) step
> 0x000000000004000d0 < start+32>:
                                        00 00 07 8b
                                                         add
                                                                 x0, x0, x7
gdb) step
:> 0x000000000004000d4 <_start+36>:
                                        48 04 40 f9
                                                         ldr
                                                                 x8, [x2, #8]
gdb) step
> 0x000000000004000d8 < start+40>:
                                                         ldr
                                                                 x9, [x3, #8]
                                        69 04 40 f9
adb) step
Show Applications poodc <_start+44>:
                                        07 7d 09 9b
                                                         mul
                                                                 x7, x8, x9
gue,
```

## Step through next 2 lines:

Here we store the multiplication of the second vector and add the product so far

```
-Register group: general-
x0
                0x32
                                     50
x1
                0x0
х2
                0x410104
                                     4260100
х3
                0x41011c
                                     4260124
                                     4260148
х4
                0x410134
х5
                0xa
                                     10
хб
                0x1
x7
                0x28
                                     40
x8
                0x14
                                     20
х9
                0x2
x10
                0x0
                                     0
                0x0
                                     0
x11
x12
                0x0
                                     0
 -lab4.s-
       22
       23
                LDR X8, [X2,8] //Load the second element of vec1
       24
       25
                LDR X9, [X3,8] //Load the second element of vec2
       26
       27
                                  /Multiplies X5 and X6 storing in X7
                                 //Dest X0 add X0 and X7
       28
                ADD X0. X0. X7
       29
       30
                LDR X10, [X2,16] //Load the third element of vec1
                LDR X11, [X3,16] //Load the thrid element of vec2
       31
       32
                MUL X7, X10, X11 //Multiplies X5 and X6 stored in X7
       33
                ADD X0, X0, X7 //Dest X0 add X0 and X7
remote Thread 1.5141 In: _start
                                                                                L30
                                                                                       PC: 0x40
(gdb) step
> 0x000000000004000c8 < start+24>:
                                         66 00 40 f9
                                                          ldr
                                                                  x6, [x3]
(gdb) step
> 0x000000000004000cc <_start+28>:
                                         a7 7c 06 9b
                                                          mul
                                                                  x7, x5, x6
(gdb) step
> 0x000000000004000d0 <_start+32>:
                                         00 00 07 8b
                                                          add
                                                                  x0, x0, x7
(gdb) step
> 0x000000000004000d4 < start+36>:
                                         48 04 40 f9
                                                         ldr
                                                                  x8, [x2, #8]
(gdb) step
                                                                  x9, [x3, #8]
> 0x000000000004000d8 < start+40>:
                                         69 04 40 f9
                                                          ldr
(gdb) step
> 0x000000000004000dc <_start+44>:
                                                                  x7, x8, x9
                                         07 7d 09 9b
                                                         mul
(gdb) step
=> 0x000000000004000e0 <_start+48>:
                                         00 00 07 8b
                                                          add
                                                                  x0, x0, x7
(gdb) step
      0000000004000e4 < start+52>:
                                         4a 08 40 f9
                                                                  x10, [x2, #16]
                                                         ldr
(gdb)
```

### Step through the next 2 lines:

Here we are loading in the 3rd elements of our vectors.

```
-Register group: general—
x0
                0x32
                                     50
x1
                0x0
                                     0
x2
                                     4260100
                0x410104
х3
                0x41011c
                                     4260124
х4
                0x410134
                                     4260148
x5
                0xa
                                     10
хб
                0x1
                                     1
х7
                                     40
                0x28
x8
                0x14
                                     20
х9
                0x2
x10
                                     30
                0x1e
x11
                0x3
                                     3
x12
                0x0
                                     0
 -lab4.s
       28
                ADD X0, X0, X7
                                   Dest X0 add X0 and X7
       29
       30
                LDR X10, [X2,16] /
                                    Load the third element of {\sf vec1}
       31
                LDR X11, [X3,16] //Load the thrid element of vec2
       32
       33
                MUL X7, X10, X11
                                   //Multiplies X5 and X6 stored in X7
       34
                                //Dest X0 add X0 and X7
                ADD X0, X0, X7
       35
       36
                STR X0, [X4] //Store the result into X4
       37
       38
       39
       40
emote Thread 1.5141 In: start
                                                                                  L33
                                                                                        PC:
(gdb) step
> 0x000000000004000d0 < start+32>:
                                         00 00 07 8b
                                                          add
                                                                   x0, x0, x7
(gdb) step
=> 0x000000000004000d4 <_start+36>:
                                         48 04 40 f9
                                                          ldr
                                                                   x8, [x2, #8]
(gdb) step
=> 0x000000000004000d8 <_start+40>:
                                         69 04 40 f9
                                                          ldr
                                                                   x9, [x3, #8]
(gdb) step
> 0x000000000004000dc <_start+44>:
                                         07 7d 09 9b
                                                                   x7, x8, x9
                                                          mul
(gdb) step
> 0x000000000004000e0 <_start+48>:
                                                                   x0, x0, x7
                                         00 00 07 8b
                                                          add
(gdb) step
> 0x000000000004000e4 < start+52>:
                                         4a 08 40 f9
                                                          ldr
                                                                   x10, [x2, #16]
(gdb) step
> 0x00000000004000e8 <_start+56>:
                                                                   x11, [x3, #16]
                                         6b 08 40 f9
                                                          ldr
gdb) step
> 0x000000000004000ec <_start+60>:
                                         47 7d 0b 9b
                                                          mul
                                                                   x7, x10, x11
```

# Step through next 2 lines:

Here we store the multiplication of the third vector and add the product so far

```
-Register group: general-
x0
                0x8c
                                     140
x1
                0x0
                                     0
х2
                0x410104
                                     4260100
х3
                0x41011c
                                     4260124
х4
                0x410134
                                     4260148
x5
                                     10
                0xa
хб
                0x1
x7
                0x5a
                                     90
x8
                0x14
                                     20
х9
                0x2
x10
                0x1e
                                     30
x11
                0x3
                                     3
x12
                0x0
                                     0
 -lab4.s
        28
                ADD X0, X0, X7 //Dest X0 add X0 and X7
        29
        30
                LDR X10, [X2,16]
                                    Load the third element of vec1
                                   /Load the thrid element of vec2
        31
                LDR X11, [X3,16] /
        32
                                   //Multiplies X5 and X6 stored in X7
        33
                MUL X7, X10, X11
                ADD X0, X0, X7 //Dest X0 add X0 and X7
        34
        35
        36
                STR X0, [X4] //Store the result into X4
        37
        38
        39
        40
remote Thread 1.5141 In: _start
                                                                                 L36
                                                                                       PC: 0x4000
(gdb) step
                                         69 04 40 f9
                                                                  x9, [x3, #8]
                                                          ldr
=> 0x000000000004000d8 <_start+40>:
(gdb) step
> 0x00000000004000dc <_start+44>:
                                         07 7d 09 9b
                                                          mul
                                                                  x7, x8, x9
(gdb) step
> 0x00000000004000e0 <_start+48>:
                                         00 00 07 8b
                                                          add
                                                                  x0, x0, x7
(gdb) step
> 0x000000000004000e4 < start+52>:
                                                                  x10, [x2, #16]
                                         4a 08 40 f9
                                                          ldr
gdb) step
> 0x00000000004000e8 <_start+56>:
                                         6b 08 40 f9
                                                                  x11, [x3, #16]
                                                          ldr
gdb) step
=> 0x00000000004000ec <_start+60>:
                                         47 7d 0b 9b
                                                          mul
                                                                  x7, x10, x11
gdb) step
> 0x00000000004000f0 <_start+64>:
                                         00 00 07 8b
                                                          add
                                                                  x0, x0, x7
(gdb) step
=> 0x000000000004000f4 <_start+68>:
                                         80 00 00 f9
                                                          str
                                                                  x0, [x4]
(gdb)
```

### Step until line 40:

Use x/1dg &dot: The result of the dot product should be stored here.

We see it prints 140 which is the dot product

```
x0
                0x8c
                                     140
x1
                0x0
                                     0
                                     4260100
                0x410104
x2
                                     4260124
х3
                0x41011c
                0x410134
                                     4260148
x4
                0xa
                                     10
 Files
                0x1
x7
                0x5a
                                     90
                0x14
                                     20
x8
x9
                0x2
x10
                0x1e
                                     30
x11
                0x3
                                     3
x12
                0x0
                                     0
 -lab4.s-
                LDR X11, [X3,16] //Load the thrid element of vec2
        31
        32
        33
                MUL X7, X10, X11
                                   //Multiplies X5 and X6 stored in X7
        34
                ADD X0, X0, X7 //Dest X0 add X0 and X7
        35
        36
                STR X0, [X4] //Store the result into X4
        37
        38
        39
        40
                MOV X0, 0
                                     // Set register X0 to 0 (return code)
        41
                 MOV X8. 93
                                        Set register X8 to 93 (syscall number for exit)
        42
        43
                SVC 0
                                        Invoke syscall to exit
remote Thread 1.5141 In: _start
                                                                                 L41
                                                                                       PC: 0>
=> 0x00000000004000e0 <_start+48>:
                                         00 00 07 8b
                                                          add
                                                                   x0, x0, x7
(gdb) step
=> 0x000000000004000e4 <_start+52>:
                                         4a 08 40 f9
                                                          ldr
                                                                  x10, [x2, #16]
(gdb) step
=> 0x000000000004000e8 <_start+56>:
                                         6b 08 40 f9
                                                                  x11, [x3, #16]
                                                          ldr
(gdb) step
=> 0x00000000004000ec <_start+60>:
                                         47 7d 0b 9b
                                                          mul
                                                                  x7, x10, x11
(gdb) step
=> 0x000000000004000f0 <_start+64>:
                                         00 00 07 8b
                                                          add
                                                                  x0, x0, x7
(gdb) step
=> 0x000000000004000f4 <_start+68>:
                                         80 00 00 f9
                                                          str
                                                                  x0, [x4]
(gdb) step
=> 0x000000000004000f8 <_start+72>:
                                         00 00 80 d2
                                                          mov
                                                                  x0, #0x0
/ #0
(gdb) x/1dg &dot
                140
(gdb)
```

# Step to end of program:

## Ends program

```
-Register group: general
x0
                0x0
                                     0
x1
                0x0
                                     0
x2
                0x410104
                                      4260100
х3
                0x41011c
                                     4260124
                                     4260148
                0x410134
x4
x5
                0xa
                                      10
хб
                0x1
                                      1
                0x5a
                                     90
x7
x8
                0x5d
                                      93
x9
                0x2
                                      2
x10
                0x1e
                                      30
x11
                0x3
                                      3
x12
                0x0
                                     0
 -lab4.s-
        43
                SVC 0
                                         Invoke syscall to exit
        44
        45
        46
            vec1: .quad 10, 20, 30
            vec2: .quad 1, 2, 3
        47
           dot: .quad 0
        48
        49
        50
        51
        52
        53
        54
        55
exec No process In:
                                                                                        L??
(gdb) step
=> 0x000000000004000f0 <_start+64>:
                                          00 00 07 8b
                                                           add
                                                                   x0, x0, x7
(gdb) step
=> 0x000000000004000f4 <_start+68>:
                                          80 00 00 f9
                                                           str
                                                                   x0, [x4]
(gdb) step
=> 0x00000000004000f8 <_start+72>:
                                          00 00 80 d2
                                                                   x0, #0x0
                                                           MOV
/ #0
(gdb) x/1dg &dot
                140
(gdb) step
=> 0x00000000004000fc < start+76>:
                                          a8 0b 80 d2
                                                           mov
                                                                   x8, #0x5d
/ #93
(gdb) step
=> 0x00000000000400100 < start+80>:
                                          01 00 00 d4
                                                           svc
                                                                   #0x0
(gdb) step
[Inferior 1 (process 1) exited normally]
(gdb)
```