HW3 CS 4115

Andrew Grant, amg2215@columbia.edu

1

(a) Let any integer $i = a_3a_2a_1a_0$ where a_i is some byte

a_3	a_2	a_1	a_0	a[0][0]
a_3	a_2	a_1	a_0	a[0][1]
a_3	a_2	a_1	a_0	a[0][2]
a_3	a_2	a_1	a_0	a[1][0]
a_3	a_2	a_1	a_0	a[1][1]
a_3	a_2	a_1	a_0	a[1][2]

(b) address $= (i \times 12) + (j \times 4)$

(c) Assembly code:

```
        .file   "main.c"
        .text
        .globl  main
        .type   main, @function
main:
.LFB0:
        .cfi_startproc
        movl    $0, a(%rip)
        movl    $4, a+4(%rip)
        movl    $8, a+8(%rip)
        movl    $12, a+12(%rip)
        movl    $16, a+16(%rip)
        movl    $20, a+20(%rip)
        movl    $0, %eax
        ret
        .cfi_endproc
.LFE0:
        .size   main, .-main
        .comm   a,24,16
        .ident  "GCC: (Ubuntu 4.8.4-2ubuntu1~14.04.1) 4.8.4"
        .section        .note.GNU-stack,"",@progbits
```

Corresponding C code:

```
int a[2][3];
int main()
{
```

```
        int i;
        int j;
        for (i = 0; i < 2; i++) {
                for (j = 0; j < 3; j++) {
                        a[i][j] = (12 * i) + (4 * j);
                }
        }

        return 0;
}
```

Basically, I put the value $(12 \times i) + (4 \times j)$ into $a[i][j]$ in a for-loop. As can be seen, at each step in the for-loop, the value of $(12 \times i) + (4 \times j)$ is precisely the offset from the base address of the label a (the base address of the 2D array) that $(12 \times i) + (4 \times j)$ is stored into; thus $(12 \times i) + (4 \times j)$ does indeed correspond to the address of $a[i][j]$