Full Name: Zachary Wilcox
A-number: A02080016

ECE 5720, Fall 2020

Take Home 2

Due: October 1, 2020 (3:00 PM)

Instructions:

- Write your A-number on top of every sheet.
- Make sure that your exam is not missing any sheets, then write your full name on the front.
- The exam has a maximum score of 20 points. You must show your steps clearly to get any credit. Good luck!

1 (10):	
2 (10):	
TOTAL (20):	

Problem 1. (10 points):

Consider the source code below, where `M` and `N` are constants declared with `#define`.

```
int mat1[M][N];
int mat2[N][M];

void copy_element(int i, int j)
{
  mat2[i][j] =  mat1[j][i];
}
```

i j Rdx rax

A. Suppose the above code generates the following assembly code:

```
copy_element:
        movslq  %esi, %rsi
        movslq  %edi, %rdi
1       leaq    (%rsi,%rsi,2), %rdx
2       movq    %rdi, %rax
3       salq    $4, %rax
4       leaq    (%rsi,%rdx,4), %rdx
5       addq    %rdi, %rax
6       addq    %rsi, %rax
7       leaq    (%rdx,%rdi), %rdi
8       movl    mat1(,%rdi,4), %edx
9       movl    %edx, mat2(,%rax,4)
        ret
```

3i

j * 16

12 j + i = 13 i

j + 16 j = 17 j + i

j = j + 13 i

rax = 17 j + i

What are the values of `M` and `N`?

`M` = 13

`N` = 17

Problem 2. Structs and Arrays (10 points)

You are given the following C program run on a 64-bit x86-64 (little endian) processor:

```
struct diddle {
   int x;
   struct diddle *y;
   int z;
   char c[3];
 };

 int main(void) {
   struct diddle d;
   d.x = 0xdeadbeef;
   d.y = &d;
   d.z = d.x >> 16;
   d.c[0] = 0x12;
   d.c[1] = 0x34;
   d.c[2] = 0x56;
   return 0;
 }
```

a. Below is a view of the stack. Suppose we have just reached the return statement and assume `d` is placed at address **`0x7fffffffac0`**. Please fill in the bytes on the stack in hex (you may omit the 0x prefix).

Address	+0	+1	+2	+3	+4	+5	+6	+7
0x7fffffffac0	ef	be	ad	de				
0x7fffffffac8	c0	fa	ff	ff	ff	07	00	00
0x7fffffffad0	ad	de	FF	FF				
0x7fffffffad8	56	34	12					

c. What is the total size of this struct in bytes?

19 bytes used, 28 total

d. Is there a reordering of the fields in diddle that would reduce its total size? If so, what is it?

```
int x;
int z;
struct diddle *y;
char c[3];
```