Most did awesome in this problem set. In problem 1b, there are some overly precise answers. In part ii of 1b, some people did not divide the answer by 17—the problem says the stack of paper is 17 times taller than Fine Hall, so as the last step, the height has to be divided.

There are some misunderstanding of running time in part 3b. The nested loops make the running time $N^2$, because the outer loop iterates $N$ times, and for each value of $i$, the inner loop also iterates $N$ times. $N^2$ is quadratic. In comparison, $2^N$ is exponential, which does not correspond with the given program. The running time is also not linear, which would be true if we only had a single non-nested loop. Additionally, for part 3d, the number of lines in your code would not have a great impact on the running time.

In part 3c, some people used the * operator in their `mul()` function. Note that the problem states that you should use repeated addition to replace this operator. Some people implemented a recursive function, calling `mul()` inside `mul()`. This approach might require a little more thought, and in certain cases it could actually fail—try `mul(999, 999)`. This will exceed the maximum recursion depth in Python because too many functions remains to be executed. For the purpose of this assignment, I did not take any point off if you successfully implemented the recursive approach.
Problem 1:  [15 pts, 3 each]

(a) Is 1TB too high, too low, about right, and why?

I think it’s too low, but it depends on the assumptions, so it’s fine if someone made their case plausibly.

My reasoning: 150 miles * 5280 feet * 12 inches = 9.5M inches, or 10^7 inches. If a 1-inch book has 500 pages and each page has 400 words, that’s 200,000 words/book or about 1 MB/book, so 10^7 * 10^6 = 10^13 = 10 TB.

(b)  
   (i) How far away is Salt Lake City in miles or km?

1,900 miles (3,000km). 11 million sheets x 11 inches / 12 / 5280. Watch out for excessive precision.

   (ii) How tall is Fine Hall in feet or meters?

200 feet (65m)? One way to reason: a ream of paper (what you see at printers) is 500 sheets and is about 2 inches thick. Or a sheet is 0.1 mm. We used this value in an earlier problem set.

(c) How many five-year periods to clean it up?

20 periods. Dividing by 2 20 times takes it from 1.6 * 10^6 sq km to 1.6 sq km, which is less than 2.4 sq km. The problem was carefully worded so as to make 20 really the right answer: “how many five-year periods ... to fit comfortably”.

(d) How does $250/month get to $1200/month in 32 years?

5%/year doubles in about 14 years to 500 and again to 1000 at 28 years. Four more years at 5% would add somewhat more than 4 * 5% of 1000 or 200, so 1200. It’s not surprising for someone who knows the Rule of 72.

Problem 2:  [8 pts, 2 each]

(a) How many bits for IPv4.5 addresses?

33. 2^33 is somewhat over 8 billion, which is enough.

(b) How many bytes for IPv4.5 addresses?

5. Round up

(c) How many Ethernet addresses per person (power of 2)?
\[
\frac{2^{48}}{2^{33}} = 2^{15}
\]

(d) How many IPv6 addresses per person (power of 2)?

\[
\frac{2^{128}}{2^{33}} = 2^{95}
\]

**Problem 3: [14 pts: 5 + 2 + 5 + 2]**

(a) What does it print if \( N \) is 3?

```
1 1, 1 2, 1 3, 2 1, 2 2, 2 3, 3 1, 3 2, 3 3
```

(b) How does the running time in proportion to \( N \)?

\( N^2 \) (quadratic). It goes through the outer loop \( N \) times; for each of these it does the inner loop \( N \) times. \( N^2 \) is NOT exponential.

(c) Paste your `mul` function here (monospace font, please)

One of several ways:

```
prod = 0
while m > 0:
    prod = prod + n
    m = m - 1
return prod
```

An alternative, using more familiar constructs: use a loop like the one in (a):

```
prod = 0
i = 1
while i <= m:
    prod = prod + n
    i = i + 1
return prod
```

(d) How would you change the function?

Use the smaller of \( m \) and \( n \) as the loop limit, so that you are adding up fewer instances of the larger number.