1.a.i. We are given a map of size 4000Km by 4000Km, and this has to be folded into size 1m by 1m. Folding on each side once will reduce the width of that side by half. Number of folding needed to reduce the size from 4000Km to 1m in either of the side of the map is:

\[
\frac{4 \times 10^6 m}{2^x} = 1m
\]

\[
2^x = 4 \times 10^6
\]

\[
x = \log_2(4 \times 10^6)
\]

\[
x \approx 22
\]

Each side has to be folded 22 times to reduce to the width of 1m. Since, we need to fold on both the sides in alternate direction to get a square of size 1m by 1m, we need to fold the map by 22 * 2 i.e., 44 times.

1.a.ii. Folding a map of size 4000Km by 4000Km, 12 times involves folding each side of the map by 6 times. Hence the width of each side of the map after folding it by 6 times respectively will be:

\[
x = \frac{4 \times 10^6 m}{2^6}
\]

\[
x = 62,500m
\]

After folding the map by 12 times, it reduces to the size of 62.5Km by 62.5Km.

1.a.iii. Every fold results in twice the width \((2^{\text{number of folds}})\). Thickness of the paper is 0.1mm, and it is folded 12 times.

\[
0.1 mm \times 2^{12} = 409.6 mm
\]

Hence, the thickness of the folded paper would be 0.4096m.

1.b. A byte is a single character on the keyboard and we typed at one character per second, which means we typed at the rate of 1 byte per second. Time taken at the given rate to type a petabyte length document would be
\[ 10^{15} \text{bytes} \times \frac{1 \text{ second}}{1 \text{ byte}} \times \frac{1 \text{ hour}}{3600 \text{ seconds}} \times \frac{1 \text{ day}}{24 \text{ hours}} \times \frac{1 \text{ year}}{365 \text{ days}} = 31,709,792 \text{ years} \]

So the 30 million estimate is about right.

1.c. A link transmitting at the rate of 40 terabits per second for a year, would have transmitted:

\[ \frac{40 \times 10^{12} \text{ bits}}{\text{second}} \times \frac{1 \text{ byte}}{8 \text{ bits}} \times \frac{3600 \text{ seconds}}{1 \text{ hour}} \times \frac{24 \text{ hours}}{1 \text{ day}} \times \frac{365 \text{ days}}{1 \text{ year}} = 1.578 \times 10^{20} \text{ bytes per year} \]

We need 1.578 \times 10^{20} \text{ bytes} of disk space to store the data.

1.d. I assume that on an average each pizza is of size 14 inches (diameter), hence the area of each pizza is \( \pi \times 7^2 \), which is around 154 square inches.

I also assume that each pizza has around 8 slices. I assume that each Princeton student on an average consumes 2 slices of pizza in a week. There are 36 weeks in an academic year.

I also assume that there are around 5000 undergraduate students in Princeton.

Hence, total amount of pizza eaten by all Princeton students in an academic year would be:

\[ \frac{154 \text{ inches}^2}{8 \text{ slices}} \times \frac{2 \text{ slices}}{1 \text{ week,1 student}} \times \frac{36 \text{ weeks}}{1 \text{ academic year}} \times 5000 \text{ students} = 6,930,000 \text{ inches}^2 \]

All Princeton students in an academic year consume 6,930,000 square inches of Pizza.

2.a. 

Get a number from Keyboard → Print the number → Is it zero? → Yes → Stop → No → Subtract 1 from the number → Is it zero? → Yes → Stop
GET
Top PRINT
    IFZERO Finish
    SUB 1
    GOTO Top
Finish STOP

2.b.

GET
STORE Sum
Top IFZERO Finish
    SUB 1
    STORE Num
    ADD Sum
    STORE Sum
    LOAD Sum
    GOTO Top
Finish LOAD Sum
    PRINT
    STOP
Num init 0
Sum init 0
2.c.

[Diagram showing a flowchart with steps:
- Retrieve Numbers from keyboard
- Store in Count
- Store 1 in Num1 and Num2
- Step: Yes, Is Count zero? No, Subtract 1 from Count, Point Num1
- Is Count zero? Yes, Step, No, Subtract Count by 1, Add Num1, Num2, Store Sum in Temp, Store Num2 in Num1, Store Temp in Num2, Point Num1]
2d.

GET
STORE Count
IFZERO Finish
LOAD Num1
PRINT
LOAD Count
SUB 1
STORE Count
Loop IFZERO Finish
SUB 1
STORE Count
LOAD Num2
ADD Num1
STORE Temp
LOAD Num2
STORE Num1
LOAD Temp
STORE Num2
LOAD Num1
PRINT
LOAD Count
GOTO Loop
Finish STOP

Count INIT 0
Num1 INIT 1
Num2 INIT 1
Temp INIT 0