# IDbyDNA Candidate Challenge – Software Engineer

Please complete and answer the following questions. Keep in mind that we don’t want to see unnecessary complexity in the answers – please answer as clearly and concisely as possible.

## Part 1

Download a sample dataset using this link:

<https://drive.google.com/file/d/1zYt0XHzPOI37klCEpy0pjQIBpaf5wrz4/view?usp=sharing>.

This is a FASTA file (for explanation of the format see <https://en.wikipedia.org/wiki/FASTA_format>). The first entry is:

>SRR1748776.1 1 length=251

CGGTTCAGCAGGAATGCCGAGATCGGAAGAGCGGTTCAGCAGGAATGCCGAGACCGGATAGCGATCTCGT

ATGCCGTCTTCTGCTTGAAAAAAAAAGACAAGGCTCCTGAATTCGCGTCTGCATATCGGGTGACCATCCC

CCAAGGCCTAATCCGCCAACCTGACCGACAGCGATCCATTACCGCGAGGGAAAGGCGCTACTACCCCCTG

TGAGGTCAGCGAACCAGATCCTTACACCGGATCGGTATAGC

The line that starts with a ‘>’ is called the header. The lines that follow are the sequence. Even though there can be line breaks in the sequence, it is actually to be interpreted as one long string. In this sample we call each of the header/sequence blocks a *read*. There are 130016 reads in the file you have downloaded. All the reads in this sample are the same length of 251 base pairs (letters).

A k-mer is a substring of length *k*. For example, the first 6-mer of CGGTTCAGCAGGAATGCCGAGATCGGAAGA would be CGGTTC. The second 6-mer would be GGTTCA, the third 6-mer would be GTTCAG, etc.

Build a hash table to count all of the k-mers of length 25 (25-mers) in the downloaded reads. Exclude any k-mer that contains a letter that is not A, T, C, or G. Do not use any third-party hash tables, but you can use a third-party hashing function.

1. How many distinct k-mers are there?
2. What is the total number of k-mers?
3. Which k-mer has the highest count?

Provide your hash table code as an attachment.

## Part 2

As part of a larger application that needs access to this data, create something that will accept input from multiple data sources. The format of the input data will be different; however, the data you need to work with always be somewhere inside the data you receive.

The data you are concerned with is the pairing of a string, and its associated number. For example, a pairing might be "A" and 1.

The input you have available to you will be in one of the following formats. Keep in mind that in the future, there may be 10+ different input formats you would need to deal with.

FormatA {

Part1: "A",

Part2: "B",

Part3: "C",

NumberPart1: 1,

NumberPart2: 2,

NumberPart3: 3

}

FormatB {

Count: 2,

Parts: ["A", "B", "X", "X"],

Numbers: [1, 2]

}

FormatB {

Count: 3,

Parts: ["A", "B", "X", "X"],

Numbers: [1, 2, 88]

}

FormatC {

NumberStrings {

1: "A",

2: "B",

3: "C",

4: "D"

}

}

The format of the data the application you're concerned with is as so:

CommonFormat {

"A": 1,

"B": 2,

"C": 3

}

How would you structure an solution that would handle these formats, as well as have the ability to handle even more input formats in the future?

Write such an application such that it would convert the above inputs into something to the effect of:

A:1 B:2 C:3

A:1 B:2

A:1 B:2 X:88, error found an X

A:1 B:2 C:3 D:4

If there is a pair where the key "X" is mapped to a number, then this is considered an error condition, and there should be some indication that this error condition was detected.

Assumptions that are safe to make:

1. Strings will be at least length 1
2. You only need to consider cases where there are valid pairs. e.g., A valid part string will always have a valid number, and vice versa.
3. You do not need to read from any external file, or source. It's perfectly acceptable to define and populate structures inside the application for the purposes of this challenge.