Can you count the Oreo chocolates?

(Intermediate/Senior engineer v1)

Write a class library using OOP paradigm that will be able to decode SGTIN-96 EPC tags. When doing decoding operation, input would be EPC tag string represented in the hexadecimal format.

Using your class library for decoding SGTIN-96 EPCs write a simple console application that will scan list of tags found in file tags.txt and answer you following question: *What is exact count of **Milka Oreo chocolates found in tags.txt file?***

Your program should also list all the serial numbers of all Milka Oreo chocolates found encoded using **SGTIN-96 standard** and list all the tags which are not encoded properly in SGTIN-96 format.

For more information about what is SGTIN and how you can decode all those values please refer to the next chapter of this document. Also, please note that there is file called *data.csv* which have all the required information needed for successful finish of this task.

You can use any language of your choice but if possible use C# or Java. Application can be a simple console application which will read all the SGTIN EPC numbers from *tags.txt* file and output the count and lists based on the condition defined in first part of the task.

Upload the source code to GitHub and send link to the repository to renato@tagitinc.com.

Bonus points

- Write simple API application which will have endpoints for decoding and validating the SGTIN-96 EPC tags. Input should be hexadecimal representation of your tag and output should be JSON/XML payload with all SGTIN decoded informations
- Write basic unit tests for your decoding class library

SGTIN

SGTIN or Serialized Global Trade Item Number EPC scheme is used to assign a unique identity to an instance of a trade item, such as a specific instance of a product or SKU.

The easiest way how to understand is to compare it with UPC number which you can find under the barcode of any trade item bought in store. The main difference is that UPC only represent a specific product type of some manufacturer while SGTIN number gives as a specific instance of that same product type of same manufacturer.

UPC will only tell us that item is Milka Oreo chocolate, but SGTIN will tell us the same alongside with unique serial item of that chocolate. This allows us to be aware of specific instances of the items around us.

Now some more technical details about the SGTIN standard. It comes in 3 different types:

- SGTIN-64
- SGTIN-96
- SGTIN-198

Number after SGTIN represent number of bits that SGTIN number can hold. So, for SGTIN-96 we expect 96 bits of information.

SGTIN consists of following important elements:

- GS1 company prefix
- Item reference (item type)
- Serial number

In the SGTIN EPC you will also find following elements which are very important for process of decoding the number:

- Header
- Filter
- Partition

We already told that SGTIN-96 EPC consists of 96 bits and they are distributed in following order:

- Header (8 bits)
- Filter (3 bits)
- Partition (3 bits)
- GS1 company prefix (20 – 40 bits)
- Item reference (24-4 bits)
- Serial reference (38 bits)

Header value

Header value of SGTIN-96 is always fixed to the same value which is represented in binary as:

0011 0000

Header Value (binary)	Header Value (hexadecimal)	Encoding Length (bits)	Coding Scheme
0011 0000	30	96	SGTIN-96
0011 0110	36	198	SGTIN-198

Filter

Filter values are numeric from 0 to 7 (decimal). Full list of possible filter values is:

Type	Filter Value	Binary Value
All Others	0	000
Point of Sale (POS) Trade Item	1	001
Full Case for Transport	2	010
Reserved	3	011
Inner Pack Trade Item Grouping for Handling	4	100
Reserved	5	101
Unit Load	6	110
Unit inside Trade Item or component inside a product not intended for individual sale	7	111

Partition

As you probably saw on the previous page company prefix and item reference parts are not fixed in size. So how to decode something which we don't know exact length?

Say hello to Partition value. Partition values represent how much out of 44 bits are spend on company prefix and how much on item reference. Partition numeric value can only be between 0 and 6 which clearly can be seen in following table:

Partition Value (*P*)	GS1 Company Prefix		Indicator/Pad Digit and Item Reference	
	Bits (*M*)	Digits (*L*)	Bits (N)	Digits
0	40	12	4	1
1	37	11	7	2
2	34	10	10	3
3	30	9	14	4
4	27	8	17	5
5	24	7	20	6
6	20	6	24	7

So, if our partition value is 3 we then know that company will be represented with 30 bits and item reference with 14 bits. Note that company prefix and item reference are always represented with exactly 44 bits (or 13 digits) when dealing with SGTIN-96 standard of encoding.

Company prefix

This numeric value represents the company prefix. Based on this prefix (code) we can exactly know which company manufactured the item that we are trying to decode.

All the company prefixes are defined by “company” called GS1 who is responsible for managing the database of all company prefixes. To get prefix from GS1 you need to pay hefty fee (of course ☺), but that is not relevant for our task. In the data.csv file you are presented with the list of some dummy company prefixes which will tell you the company name behind the specific prefix.

Company prefix can be represented using 6 to 12 digits.

Item reference

Item reference is numeric value which can be represented using 1 to 7 digits. Item reference is defined by company or manufacturer of the specific item that we are trying to encode/decode using SGTIN-96.

For this task, in the file data.csv you will be presented with the dummy item references which you will use to exactly tell which product name is derived from specific item reference found in SGTIN EPC.

Serial reference

This is unique serial number for combination of company prefix and item reference. This number will distinct specific instance of a product.

In SGTIN-96 this is numeric value and its range is from 0 to 274 877 906 943. As you can see there are lot of serial number combinations.

So, to finalize the whole representation of the SGTIN EPC lets represent it in 2 ways:

1. First one is Hexadecimal format and this is how you will see it in file tags.txt

: 3074257BF7194E4000001A85

2. Second one is binary representation which is important when decoding the EPC:

Header (8bits) = 0011 0000
F: Filter (3bits) = 011

Company Prefix (7values) and Item Ref (6values) → Partition = 5

Partition (3bits) = 101
C: Company Prefix (24bits) = 000010010101111011111101
I: Item Ref (20bits) = 11000110010100111001
S: Serial (38bits) = 000…0001101010000101

EPC binary (96bits) =
00110000011101000010010101111011111101110001100101001110010000000000000000000000001101010000101

Note: The binary values above have been padded with leading 0's to fill their respective bit spaces. For example, with a partition value of "5" the Company Prefix bit space is 24 bits but the binary conversion is only 14 bits long thus it is padded with ten "0" bits.

Conclusion

So, based on this short course and introduction to SGTIN-96 standard and its definition try to use this new knowledge to solve task at the beginning of this document.

Feel free to use Google to find out more about SGTIN and how encoding/decoding is done. Also, if you will have any additional questions or there is some confusion with the task and its requirements feel free to contact me at renato@tagitinc.com.