Convert a string representing the lines/spaces in a 12 digit barcode/UPC-A into a string of numbers 0-9.

Introduction

A black line represents a 1 and a space represents 0. The string you are returning is the decimal representation of those binary digits. NOTE: The digits 0-9 are not represented as bits the usual way, UPC has its own tables which are preloaded as LEFT_HAND and RIGHT_HAND dictionaries.

The LEFT_HAND and RIGHT_HAND dictionaries contain 7 digit bit strings as keys and an integer for values. For example:

'0011001' = 1 in LEFT_HAND

'1100110' = 1 in RIGHT_HAND, left and right side bit strings are complements of each other.

The structure of a UPC-A is as follows:

Starts with a guard pattern, always representing 101. Known as left-hand guard pattern.

6 groups of 7 bits each, eg: ' ▌▌ ▌' = '0011001' = 1(left-hand). The first group is the number system digit.

A center guard pattern, always representing 01010. Divides "left-hand" groups and "right-hand" groups.

6 groups of 7 bits each, eg: '▌▌ ▌▌ ' = '1100110' = 1(right-hand). The last group is the modulo check digit.

Ends with a guard pattern, always representing 101. Known as right-hand guard pattern.

The format for your returned string is:

'(number_system) (left_hand) (right_hand) (modulo_check)'

Example barcode

Graphical representation:

```
'▌ ▌   ▌▌ ▌ ▌▌   ▌  ▌▌  ▌   ▌▌ ▌   ▌▌ ▌   ▌▌ ▌ ▌ ▌ ▌▌▌  ▌ ▌▌  ▌▌ ▌▌ ▌▌  ▌  ▌▌▌ ▌▌  ▌▌ ▌   ▌  ▌ ▌'
```

Binary representation:
'(101)0001101,0110001,0011001,0001101,0001101,0001101(01010)1110010,1100110,1101100,1001110,1100110,1000100(101)'

Decimal representation:

(left_guard)0,5,1,0,0,0(center_guard)0,1,2,5,1,7(right_guard)

Expected output:

'0 51000 01251 7'

The above barcode is present in the file ‘testdata1.txt’.

Task

Create an application in C# using Visual Studio (any version) that reads UPC-A codes from a file (UTF-8 encoding no BOM, example file is included), one bar code per line (Windows line feeds) and prints the corresponding string to the console (or use an output file). Use GitHub, Git or Mercurial for source control management (commit messages will be reviewed). Unit tests shall be included, preferably using NUnit3.

Inputs will always be correct, left to right, and a single bar/space will always represent 1/0. The validity of modulo check digit is irrelevant here, we're just doing conversion and formatting. Guards are not part of the returned string but will be part of the input.

Encoding characters are bar, ▌, and SPACE.

Delivery

GitHub link or archive containing a clone of the repository.

Reference

https://en.wikipedia.org/wiki/Universal_Product_Code

Attachments

testdata1.txt, UTF-8 no BOM text file containing one UPC-A example