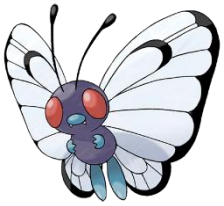


CSGE601020 Dasar-Dasar Pemrograman 1

(Foundations of Programming 1)

Tugas Pemrograman 4 (Programming Assignment 4)

EAN-13 Barcode Generator

LAST DAY for uploading the result of your work to SCeLE: Friday 10 Dec 2021 (11:55 PM).
Don't forget to write enough comments in your Python source code.
Please contact the TA (teaching assistant) for giving a demo of your work as soon as possible.
The TA will give you a mark after the demo.

Please start working on this assignment immediately. Avoid plagiarism.

Marking scheme:
60 % correctness
30 % explanation in demo session
10 % program documentation (comments, neatness)

Task Description

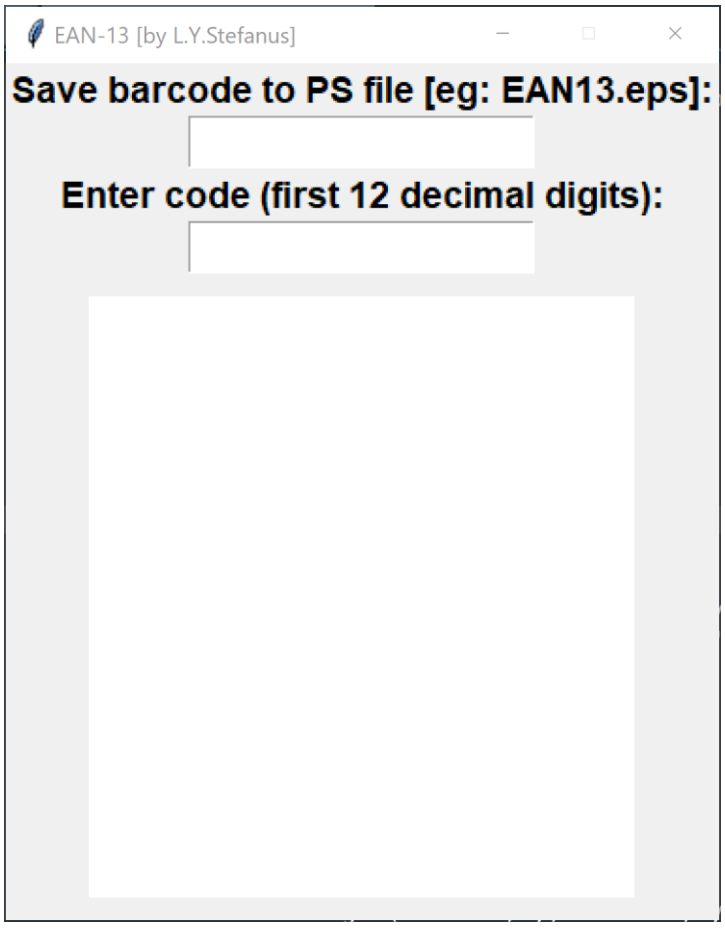
- In this assignment, you are asked to write a Tkinter GUI-based object-oriented Python program for printing EAN-13 barcodes on computer screen and saving the image as a PostScript file. A PostScript file can be printed directly on a laser printer or converted to a PDF file. A PostScript file can also be viewed by a software tool like GSview.
- Your program should consist of at least one class definition, for example, a subclass of Canvas. See the lecture slides for an example. The last line of your Python program has to be:
```
if __name__ == "__main__":
    main()
```
- EAN-13 barcodes are commonly used on various products sold in supermarkets. For example, 8997029809979 is the EAN-13 barcode for a medical mask.

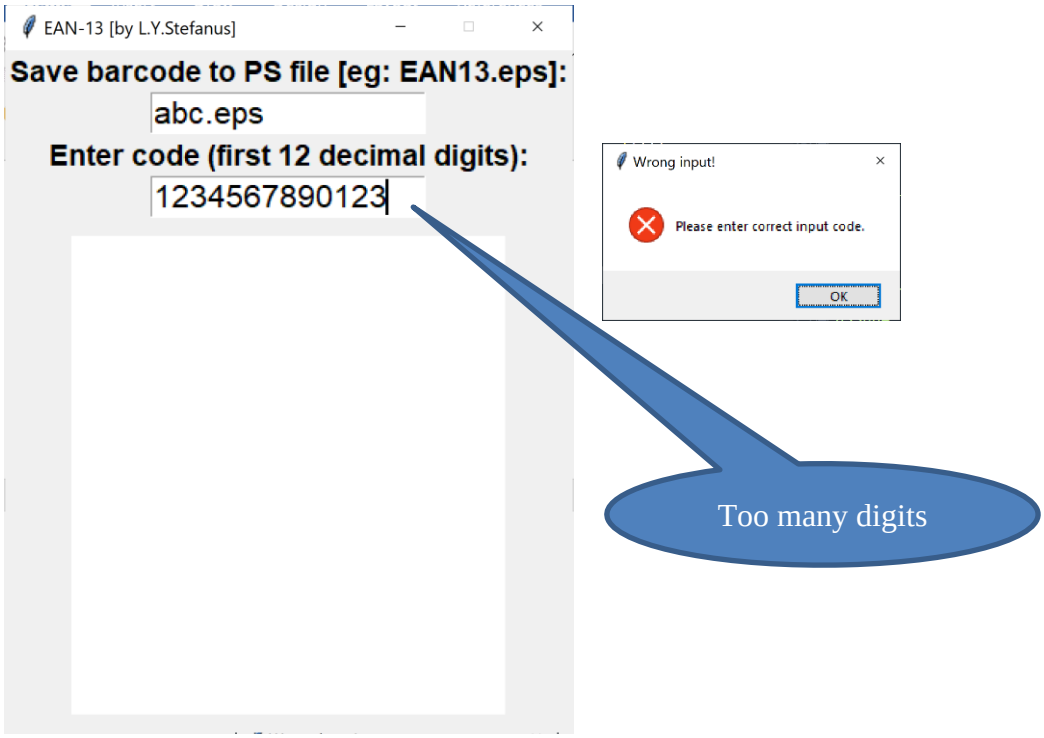
Explanation about EAN-13 barcodes taken from wikipedia is given at the end of this document.

- Your program has to do the following steps:
    1. Ask the user to enter a file name for writing the PostScript output. Validate the input accordingly. If the input is wrong, prompt the user again. The Canvas widget has a method for creating PostScript data.
    2. Ask the user to enter a code consisting of 12 decimal digits. Validate the input accordingly. If the input is wrong, prompt the user again.
    3. Calculate the **checkdigit** according to the EAN system.
    4. Print on screen (canvas) the correct EAN barcode for the 12 input digits and the checkdigit. The image is also written to the file from Step (1) in the PostScript format. Your barcode should consist of at least four different colors, like the examples given below. You can add more colors.
- You should choose a suitable data structure for storing the encoding table of EAN-13 barcodes.
- You can experiment to use the barcode reader apps from your smartphone to read the barcode resulted from your program. If this barcode is a valid product barcode, you can find the description of the product in the Internet using the barcode.

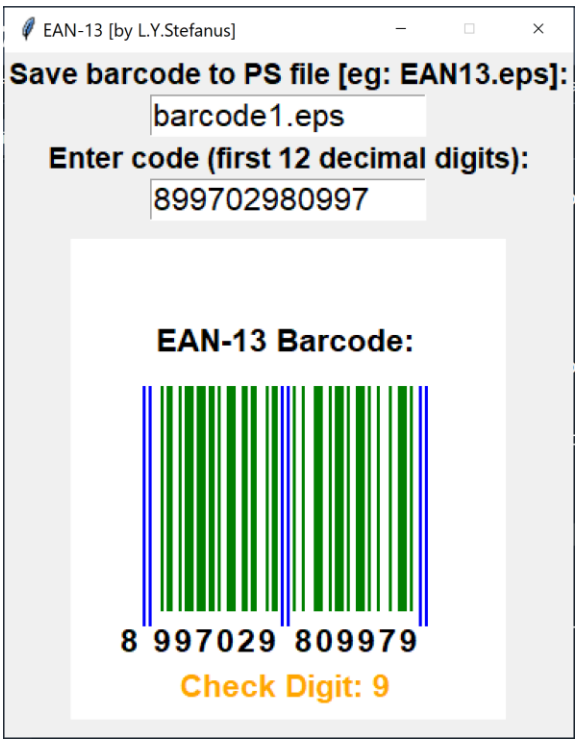
Examples of user interactions:

(1)



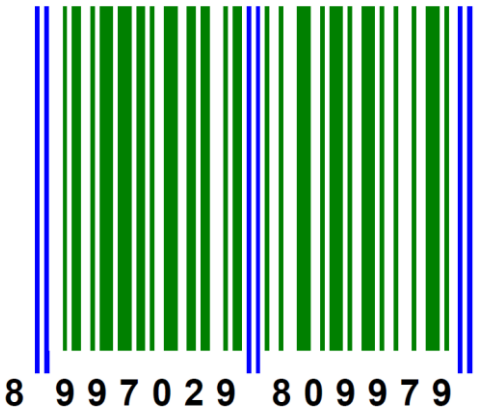


(2)



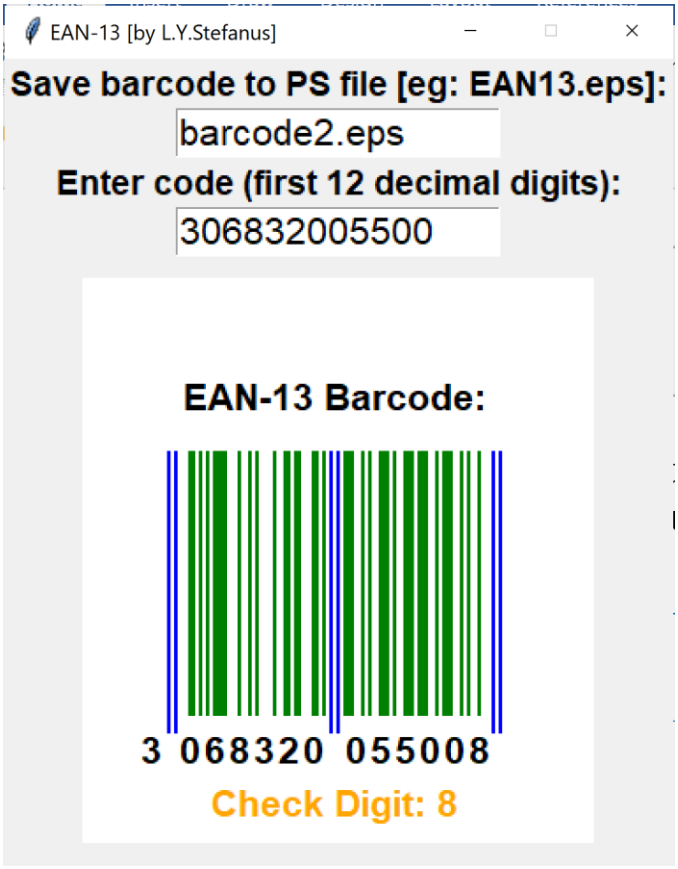
The display of the PostScript file **barcode1.eps** is as follows:

EAN-13 Barcode:



Check Digit: 9

(3)



The calculation of the checkdigit for this example is given at the end of this document. Study the EAN-13 barcode system explained in the next pages.

Happy programming! 'Met ngoding!

L. Y. Stefanus & the Asdos Team

==========

European Article Number

Adapted from Wikipedia, the free encyclopedia

An **EAN-13** barcode (originally European Article Number) is a barcoding standard which is a superset of the original 12-digit Universal Product Code (UPC) system developed in North America. The EAN-13 barcode is defined by the standards organisation GS1. All the numbers encoded in UPC and EAN barcodes are known as Global Trade Item Numbers (GTIN).

The **EAN-13** barcodes are used worldwide for marking products often sold at retail point of sale. The **GTIN-13** encoded in the bar code has four components:

- GS1 Prefix, the first two or three digits, usually identifying the national GS1 Member Organisation to which the manufacturer is registered (not necessarily where the product is actually made). When the EAN-13 symbol encodes a conversion of a 10-digit ISBN number, the GS1 Prefix will be 978 or 979 respectively, or 977 for ISSNs.
- Company number, consisting of four, five or six digits depending on number of GTIN-13s required by the manufacturer to identify different product lines.
- Item reference, consisting of two to six digits.
- Check digit, a single checksum digit. The check digit is computed modulo 10, where the weights in the checksum calculation alternate 1 and 3. In particular, since the weights are relatively prime to 10 the EAN system will detect all single digit errors. But since the difference of consecutive weights is even, the EAN system does not detect all adjacent transposition errors.

The complete number is used as a reference key to look up information about the product line held on a database; the number is never normally broken down into its components within users' systems.

The List of GS1 country codes can be found on the Internet. GS1 country code for Indonesia is 899.

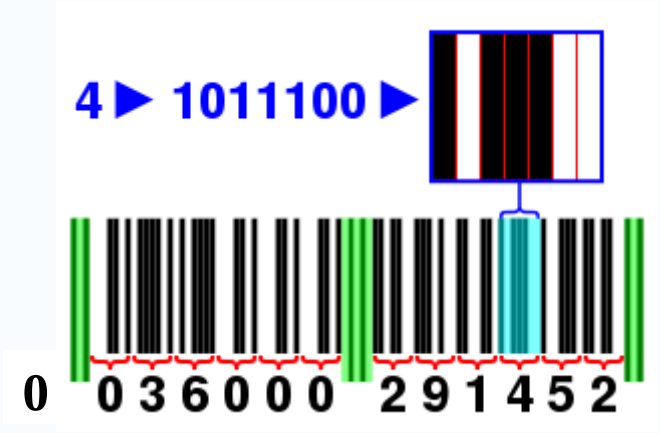
The first two or three digits of the GTIN of any product identify the GS1 Member Organisation which the manufacturer has joined. Note that EAN-13 codes beginning with 0 are rarely used, as this is just an addition to a 12-digit UPC. Since most scanners and registers worldwide can read both equally, most manufacturers in North America still only use UPC.

Encoding EAN13

To encode an EAN-13 barcode, the digits are first split into 3 groups, the first digit, the first group of 6 and the last group of 6. The first group of six is encoded using a scheme whereby each digit has two possible encodings, one of which has even parity and the other has odd parity. The first digit is encoded by selecting a pattern of choices between these two encodings for the next six digits, according to the table below. (Unlike the other digits, the first digit is not represented directly by a pattern of bars.) All digits in the last group of six digits are encoded using a single set of patterns which are the same patterns used for UPC.

If the first digit is zero, all digits in the first group of six are encoded using the patterns used for UPC, hence a UPC barcode is also an EAN-13 barcode with the first digit set to zero.

Each digit (except the first) is represented by a seven-bit sequence, encoded by a series of alternating bars and spaces. Guard bars separate the two groups of six digits.
For example:



The EAN13 encodes the two groups of 6 decimal digits as **SXXXXXXMRRRRRRE**, where S (start) and E (end) are the bit pattern 101, M (middle) is the bit pattern 01010 (called guard bars), and each X (which is L or G) and R are digits, each one represented by a seven-bit code, explained below. This is a total of 95 bits. The bit pattern for each numeral is designed to be as little like the others as possible, and to have no more than four consecutive 1s or 0s in order. Both are for reliability in scanning.

Since S, M, and E all include two bars, and each of the 12 digits consists of two bars and two spaces, all EAN13 barcodes consist of exactly (3 × 2) + (12 × 2) = 30 bars.

Structure of EAN-13

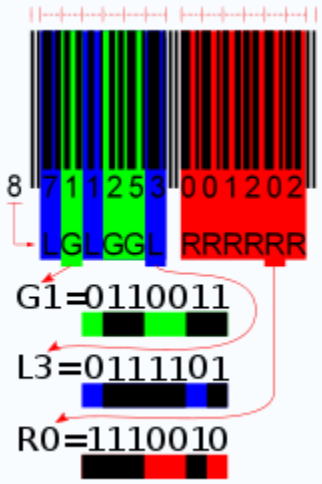
First digit	First group of 6 digits	Last group of 6 digits
0	LLLLLL	RRRRRR
1	LLGLGG	RRRRRR
2	LLGGLG	RRRRRR
3	LLGGGL	RRRRRR
4	LGLLGG	RRRRRR
5	LGGLLG	RRRRRR
6	LGGGLL	RRRRRR
7	LGLGLG	RRRRRR
8	LGLGGL	RRRRRR
9	LGGLGL	RRRRRR

Encoding of the digits

Digit	L-code	G-code	R-code
0	0001101	0100111	1110010
1	0011001	0110011	1100110
2	0010011	0011011	1101100
3	0111101	0100001	1000010
4	0100011	0011101	1011100
5	0110001	0111001	1001110
6	0101111	0000101	1010000
7	0111011	0010001	1000100
8	0110111	0001001	1001000
9	0001011	0010111	1110100

Note: Entries in the R-column are bitwise complements of the respective entries in the L-column. Entries in the G-column are the entries in the R-column reversed.

Example:



Checksum calculation

The checksum is calculated taking a varying weight value times the value of each number in the barcode to make a sum. The resulting sum modulo 10 (i.e. the last digit) is subtracted from 10, and the result is used as checksum digit (If the new result is 10, then zero is used instead).

Weights

The weight for a specific position in the EAN-code is either 3 or 1. An EAN-13 code starts with a weight of 1.

Getting the weights for a barcode

Position	1	2	3	4	5	6	7	8	9	10	11	12
Weight	1	3	1	3	1	3	1	3	1	3	1	3

Example: Let's calculate the *checkdigit* for the barcode in the execution of the Python program in example (3) above. The input code is `306832005500`. What is the correct check digit?

Answer:

checksum = (0 + 8 + 2 + 0 + 5 + 0) * 3 + (3 + 6 + 3 + 0 + 5 + 0) = 62;
x = checksum % 10 = 2;
if (x != 0) checkdigit = 10 - x else checkdigit = x;
so the checkdigit = 8.