

CS 3750 [Fall 2017]

Assignment 2

Due date: Wednesday, 20 September 2017, by 11:00 p.m.

Checksum digit is an important error detection measure for integrity check for many codes, including UPC (Universal Product Code) barcodes. Typically, an UPC is a 12-digit number where the last digit on the right serves as the checksum digit. The checksum digit can be calculated based on other 11 digits of the UPC and if it does not match with the calculated value, it is certain that there is some error in the code transmission.

The checksum digit in a 12-digit UPC is calculated as follows:

Step 1: Sum of all digits at odd positions (counting from 1, left to right, 1, 3, 5, 7, 9, 11)

Step 2: Sum of all digits at even positions except the last position (counting from left to right, 2, 4, 6, 8, 10 – the 12th position is not included in the sum)

Step 3: Calculate $(3 * \text{oddsun}) + \text{evensun}$

Step 4: If the result obtained at step 3 leaves remainder 0 when divided by 10, then checksum is 0. Otherwise, the checksum is $(10 - \text{the remainder when divided by } 10)$.

Programming:

A sample file has been provided with the assignment description. The file contains 10 different UPCs. Write a Java program to check whether the UPC codes in the sample files are valid. If found invalid, indicate what would be the correct checksum digit (assuming there is no error in other 11 digits).

Program execution requirements:

1. The program must accept the input file as a command line argument. The test input file will have similar format as the sample file provided.
2. The output format should be indication of “valid” or “invalid” for each UPC on file. If “invalid”, the output should also indicate the correct checksum digit value. You do not need to write the output onto a file. Just display it on console.

Submission instructions:

write your name, course (CS3750), semester-year (Fall 2017), and assignment number (Assignment 2 – Program 1) at the top of the program file as Java comments. Submit the program file (not the .class file) through BlazeVIEW dropbox.