**CSE 174 Spring 2019**

**PROGRAM #11: 25 points – Due Sunday, April 21, by 11:59 p.m.**

**Outcomes:**

* Write programs that use arrays
* Format and comment source code that adheres to a given set of formatting guidelines

**Scoring:**

At a bare minimum, the program you submit must have the assigned source code, and your source code must compile and run without crashing.

* If you submit source code that does not compile, your score will be zero.
* If you submit source code that roughly resembles the requirements and it compiles, but it crashes under normal operating conditions (nice input from the user), your score will be reduced by 75%.
* Deductions will be made for not meeting the usual formatting requirements (commenting, indentation, appropriate variable names, and so on.

|  |  |  |
| --- | --- | --- |
|  | **Full credit** | **No credit or Partial credit** |
| **Solve the specified problem (20 points)** | Program correctly solves the specified problem for any UPC | There are errors in the solution |
| **Use appropriate programming techniques**  **(5 points)** | Program correctly uses appropriate data types (such as a integer array to store the UPC), and uses multiple short methods to break the larger problem into smaller parts. | Inappropriate data types are used, and/or the program should have been broken into smaller methods. |

**Background:**

A barcode scanner for Universal Product Codes (UPCs) verifies the 12-digit code scanned by comparing the code’s last digit (called a check digit ) to its own computation of the check digit from the first 11 digits as follows:

1. Calculate the sum of the digits in the odd-numbered positions (the first, third, …, eleventh digits) and multiply this sum by 3.
2. Calculate the sum of the digits in the even-numbered positions (the second, fourth, …, tenth digits) and add this to the previous result.
3. If the last digit of the result from step 2 is 0, then 0 is the check digit. Otherwise, subtract the last digit from 10 to calculate the check digit.
4. If the check digit matches the final digit of the 12-digit UPC, the UPC is assumed correct.

**Requirements:**

Write a program that prompts the user to enter the 12 digits of a barcode separated by spaces. The program should store the digits in an integer array, calculate the check digit, and compare it to the final barcode digit. If the digits match, output the barcode with the message “validated.” If not, output the barcode with the message “error in barcode.” Also, output with labels the results from steps 1 and 2 of the check-digit calculations. Note that the “first” digit of the barcode will be stored in element 0 of the array.

Try your program on the following barcodes, three of which are valid. For the first barcode, the result from step 2 is 79 (0 + 9 + 0 + 8 + 4 + 0) \* 3 + (7 + 4 + 0 + 0 + 5).

0 7 9 4 0 0 8 0 4 5 0 1

0 2 4 0 0 0 1 6 2 8 6 0

0 1 1 1 1 0 8 5 6 8 0 7

0 5 1 0 0 0 1 3 8 1 0 1

A few specifics:

* Name your class Program11.
* Break this problem into small single-purpose methods. Any method should not be longer than 12 lines long (plus comments). **Your code should have at least two methods that either have array as input parameter or return array.**
* To submit, create a folder named program11. It should contain only Program11.java. Then, zip and submit that folder.

**Sample run:**

Enter the UPC you want to check: 0 7 9 4 0 0 8 0 4 5 0 1

The sum of the digits in the odd-numbered positions = 21

The sum of the digits in the even-numbered positions = 16

The weighted sum = 79

The last digit of the weighted sum = 9

The check digit = 1

The bar code you entered is 0 7 9 4 0 0 8 0 4 5 0 1

The bar code is validated

Enter the UPC you want to check: 0 2 4 0 0 0 1 6 2 8 6 0

The sum of the digits in the odd-numbered positions = 13

The sum of the digits in the even-numbered positions = 16

The weighted sum = 55

The last digit of the weighted sum = 5

The check digit = 5

The bar code you entered is 0 2 4 0 0 0 1 6 2 8 6 0

Error in barcode