Overview
The International Standard Book Number (ISBN) is a multi-digit number that identifies published books uniquely. There are two forms of the ISBN, an older 10-digit form and a newer 13-digit form. For this lab, we will be working with the 10-digit form only.

One of the features of the ISBN is built-in error detection. The right-most digit of the ISBN is called a check digit, meaning that it can be used to check whether the other 9 digits are, in fact, correct. This can be used, for example, to help determine data entry errors when typing the ISBN into a computer system while trying to order a book.

The check digit works like this (as described in Wikipedia):

Modular arithmetic is convenient for calculating the check digit using modulus 11. Each of the first nine digits of the ten-digit ISBN – excluding the check digit, itself – is multiplied by a number in a sequence from 10 to 2, and the remainder of the sum, with respect to 11, is computed. The resulting remainder, plus the check digit, must equal 11; therefore, the check digit is \((11 \text{ minus the remainder of the sum of the products modulo } 11) \text{ modulo } 11\). Taking the remainder modulo 11 a second time accounts for the possibility that the first remainder is 0. Without the second modulo operation the calculation could end up with 11 - 0 = 11 which is invalid.

For example, the check digit \(s\) for an ISBN of 0-306-40615-s is calculated as follows:

\[
s = (11 - ((10 \times 10) + (3 \times 9) + (0 \times 8) + (6 \times 7) + (4 \times 6) + (0 \times 5) + (5 \times 4) + (1 \times 3) + (5 \times 2)) \mod 11) \mod 11
\]

\[
= (11 - (0 + 27 + 0 + 42 + 24 + 0 + 24 + 3 + 10)) \mod 11 \mod 11
\]

\[
= (11 - (130 \mod 11)) \mod 11 = (11 - (9)) \mod 11 = (2) \mod 11 = 2
\]

Thus the check digit is 2, and the complete sequence is ISBN 0-306-40615-2.

Special note: the value \(s\) required to satisfy this condition might be 10; if so, an 'X' should be used…but we’re not going to worry about that possibility for this assignment.

So, in general, for an ISBN number \(d_1d_2d_3d_4d_5d_6d_7d_8d_9d_{10}\), the following will be true:

\[
d_{10} = (11 - (((d_1 \times 10)+d_2\times9)+(d_3\times8)+(d_4\times7)+(d_5\times6)+(d_6\times5)+(d_7\times4)+(d_8\times3)+(d_9\times2)) \mod 11) \mod 11
\]

The Assignment
You are to write a program that will accept a 10-digit ISBN (without the dashes) as input from the keyboard and determine whether there is any error in the number that was entered, i.e. determine whether the ISBN is valid based on its check digit. Your program will output either “The ISBN you entered is valid” or “The ISBN you entered is not valid.”
Your program should give the user the opportunity to continue testing ISBN numbers until they type in the word “no”. This input should work regardless of capitalization.

**Input Description**
Your program must ask the user to enter a 10-digit ISBN.

**Sample Execution (Bold and Underlined indicates user input)**

Enter a 10-digit ISBN: **0439064872**

The ISBN you entered is valid.

Do you want to try another (yes/no)?: **y**

Do you want to try another (yes/no)?: **yup**

Do you want to try another (yes/no)?: **yes**

Enter a 10-digit ISBN: **0439064875**

The ISBN you entered is not valid.

Do you want to try another (yes/no)?: **no**

End of program.

**Design**
Your program should have at least one method (in addition to the main method, of course) that takes a 9-digit number (an ISBN number minus its right-most digit) as a parameter and returns a check digit for it, based on the ISBN-10 algorithm described above. You may find one or more additional methods useful in organizing your code.

**Hints**
1. Don’t forget that integers can only be processed from right to left.
2. You may assume there are always exactly 10 digits in the number entered by the user.
3. Where the question “Do you want to try another” occurs in the dialog (as shown in the Sample Execution above) is a clue to which type of loop to use.

**Testing Notes**
1. You should try your program with several ISBNs to verify that it is correctly doing the calculation. Pick any of your books and find the 10-digit ISBN (if there is one…some books have only a 13-digit ISBN. **Don’t** try to use a 13-digit ISBN, though, since it won’t work with this program). **Don’t** pick any number ending in “X”, however.
2. Try entering an ISBN where you have reversed the position of two of the digits…your program should indicate that your ISBN is not valid.
3. Try entering an ISBN where you enter the wrong check digit… your program should indicate that your ISBN is not valid.

4. To aid in testing, you can have your program output additional information other than just the final message. This might include the partial results of the calculations. **Before submitting your program**, however, all outputs other than those shown in the sample above should be eliminated (a simple way to do this is to “comment out” the output statements).

5. The text input (yes/no) should be independent of capitalization, i.e. the user should be able to enter their answer with whatever capitalization they wish and the program should be able to process it. However, **your program must detect and reject** any response that is not a “yes” or “no”. Anything other than “yes” or “no” should be ignored, and your program should just reprompt the user for the input.

**Submission**
Submit your program as **Lab5-ISBN**.