**Developer Sample Code Exercise: Validating a Passport MZR**

Reference Document: - <http://www.highprogrammer.com/alan/numbers/mrp.html>

Please use the attached sample application to continue the implementation which the application can **validate a standard international passport Machine Readable Zone (MRZ) line 2.**

On a MR Passport, there are two lines. Each line is 44 characters long. There are no blank spaces; where one is necessary is it filled with the filler character~ a less than sign (<).

Here is the example MRZ.   
  
Line 1: P<UTOERIKSSON<<ANNA<MARIA<<<<<<<<<<<<<<<<<<<  
Line 2: L898902C<3UTO6908061F9406236ZE184226B<<<<<14

The Machine Readable Zone (MRZ) line 2 is last line of the passport and is split into sections:

MRZ Line 2 consists of #########CnnnYYMMDDCgyymmddCppppppppppppppCX

|  |  |
| --- | --- |
| **Section** | **Description** |
| ######### | **Passport Number (Input)**  This is the passport number, as assigned by the issuing country. Each country is free to assign numbers using any system it likes. If the number has non-letter or number characters they are replaced with the filler character <. |
| C | **[Check digit](#Checkdigit) 1 (Calculation)**  (Refer to check digit below) |
| nnn | **Nationality (Input)**  e.g. MYS for Malaysia  The issuing country or organization, encoded in three characters: -  [Country](#Country) (Refer to country list below.) |
| YYMMDD | **Date of Birth (Input)**  The date of the passport holder's birth in YYMMDD form. Year is truncated to the least significant two digits. Single digit months or days are prepended with 0.  In the example, Anna was born in August 6th, 1969. That encodes as "690806". |
| C | **[Check digit](#Checkdigit) 2 (Calculation)** (Refer to check digit below) |
| g | **Gender (Input)**  Gender of the passport holder, M for males, F for females, and < for non-specified. |
| yymmdd | **Passport Expiration Date (Input)**  The date the passport expires in YYMMDD form. Year is truncated to the least significant two digits. Single digit months or days are prepended with 0.  Anna's passport expired on June 23rd, 1994 and is encoded "940623". |
| C | **[Check digit](#Checkdigit) 3 (Calculation)** (Refer to check digit below) |
| pppppppppppppp | **Personal Number (Input)**  This field can be used for any purpose that the issuing country desires.  The United States uses this field in some unknown way. My personal passport has a single digit number in the field.  Anna's personal number is ZE184226B, this field is encoded "ZE184226B<<<<<". |
| C | **[Check digit](#Checkdigit) 4 (Calculation)**  (Refer to check digit below) |
| X | [**Check digit**](#Checkdigit) **5 (Calculation) - Final check digit**  This is a check digit for positions 1 to 10, 14 to 20, and 22 to 43 on the second line. Thus, the nationality and sex are not included in the check. The check digit is calculated using algorithm - [Check digit](#Checkdigit) (Refer to check digit below) |

**Validation result return**

|  |  |
| --- | --- |
| Validation type | Result |
| Passport number check digit | Pass or Fail |
| Date of birth check digit | Pass or Fail |
| Passport expire date check digit | Pass or Fail |
| Personal number check digit | Pass or Fail |
| Final check digit | Pass or Fail |
| Gender cross check | Pass or Fail |
| Date of birth cross check | Pass or Fail |
| Passport expire date cross check | Pass or Fail |
| Nationality code cross check | Pass or Fail |
| Passport number cross check | Pass or Fail |

**Check digit**

First, break the input into individual characters and numbers.

Next, convert non-digits into numbers. A through Z are encoded to 10 through 25. The filler character < is encoded as zero.

< A B C D E F G H I J K L

0 10 11 12 13 14 15 16 17 18 19 20 21

M N O P Q R S T U V W X Y  
  
22 23 24 25 26 27 28 29 30 31 32 33 34

Z

35

Now, multiply each number by the corresponding weighting. The first digit is multiples by 7, the next by 3, and the next by 1. The pattern then repeats (7, 3, 1, 7, 3, 1, 7, 3, 1, etc.).

Add up the results, and then divide by 10. The remainder is the check digit.

As a special case, if the personal number on the second line is not used (and thus entirely filled with the filler character <), the check digit for that section can be replaced with the filler character <.

An example for the input AB2134:

**Input :** A B 2 1 3 4 < < <

**Value :** 10 11 2 1 3 4 0 0 0

**Weight:** 7 3 1 7 3 1 7 3 1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Products:** 70 33 2 7 9 4 0 0 0

**Sum:** 70 + 33 + 2 + 7 + 9 + 4 + 0 + 0 + 0

= 125

Division: 125 ÷ 10 = 12, remainder 5

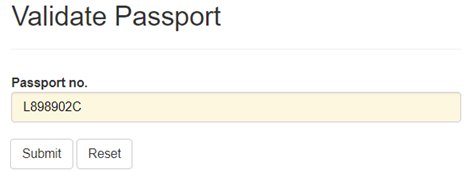
**Country**

|  |  |
| --- | --- |
| **Country** | **Code** |
| Australia | AUS |
| China | CHN |
| Indonesia | IDN |
| Malaysia | MYS |
| United Kingdom of Great Britain | GBR |
| United States of America | USA |

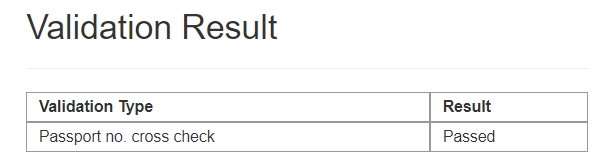
**Requirements:**

1. Please follow secure coding practices e.g. input validation and output encoding.
2. Include proper exception handling.
3. Initial implemented:

Only one input – passport no without validation.



No validation logic in place and any number will return pass result.



1. Final implementation should be similar to: -

