Project 2 Design Document

**Program Requirements:**

This program is meant to convert dates from one format to another. The program accepts an integer value representing a date in the format of ISO 8601, and displays the date as ‘MM/DD/YYYY’. We know that ISO 8601 is represented as an 8-digit integer. We know that these values represent YYYYMMDD in relation to the date.

**Program Inputs:**

* The ISO 8601 formatted date
  + int ISO
  + Must be a positive integer value.
  + The first digit should be between ‘1’ and ‘9’.
  + The value should be 8 digits in lengths.

**Program Outputs:**

* The year
  + int year
  + Must be a four-digit number.
  + Must be more than 999 and less than 1,0000.
* The month
  + int month
  + Must be between a one- and two-digit number.
  + Must not display leading zeros.
  + Must be between ‘1’ and ‘12’
* The day
  + int day
  + Must be between a one- and two-digit number.
  + Must not display leading zeros.
  + Must be between ‘1’ and ‘31’
* Sorting digits
  + Int d1, d2, …, d8
  + Must be one digit each.
  + These will be used to separate individual digits from ISO.
  + These will not be displayed to console.
* Length of integer
  + Int length
  + Will be declared with a value of ‘1’
  + Will not be displayed to the console.

**Test Plan:**

The tests will verify that the program will display the data correctly. The test will also ensure that the program will only allow valid dates. Months should only allow values between ‘1’ and ‘12’, days should only allow values between ‘1’ and ‘31’, and years should only allow values between ‘1000’ and ‘9999’. The output should only display the date as ‘MM/DD/YYYY’ with no leading zeros.

* Case 1: 1st required test case.
  + Example Input: 20191216
  + Expected Output: 12/16/2019
* Case 2: 2nd required test case.
  + Example Input: 17010301
  + Example Output: 3/1/1701
* Case 3: valid input 1
  + Example Input: 19970325
  + Expected Output: 3/25/1997
* Case 4: valid input 2
  + Example Input: 19961118
  + Expected Output: 11/18/1996
* Case 5: invalid length 1
  + Example Input: 123456789
  + Expected Output: “invalid length”
* Case 6: invalid length 2
  + Example Input: 1234567
  + Expected Output: “invalid length”
* Case 7: invalid month 1
  + Example Input: 20201313
  + Expected Output: “invalid month”
* Case 8: invalid month 2
  + Example Input: 20200012
  + Expected Output: ”invalid month”
* Case 9: invalid day 1
  + Example Input: 19970332
  + Expected Output: “invalid day”
* Case 10: invalid day 2
  + Example Input: 19970300
  + Expected Output: “invalid day”

**Solution Overview:**

We will be using the ‘iostream’ library to define our I/O stream objects. We start with creating our variables. An integer will be used to hold the value for the ISO formatted date. Three integers other will be used to hold the month, day, and year values. Lastly, the program uses 8 additional integer variables to represent each digit from ISO.

The user will receive a prompt asking them to input the ISO 8601 formatted date. This value will be stored in the variable labeled ISO. We will then verify that the integer is 8 digits in length. We will do this by converting the integer into a string and using the length function to store the value into the variable length. If the length of the string is 8, then the length of the integer is 8 digits. If this is true, the program will continue as follows. Otherwise, an error message will be sent to the console.

The value of each digit of the integer will then be stored into variables labeled d1 through d8. This is done by setting the value of the final digit equal to the value of the remainder of the quotient of the ISO variable and 10, and dropping the end digit from ISO. This will be repeated until all 8 digit values are stored in variables. d1, d2, d3, and d4 represent the year and should be stored in the year variable. To store the value, use code similar to “year=(d1\*1000)+(d2\*100)+(d3\*10) + d4”. d5 and d6 represent the month and should be stored in the month variable. d7 and d8 represent the day and should be stored in the day variable.

If month is at least 1 and at most 12, the program will check the day values. If day is at least 1 and at most 31, the program will check the year value. If the year value is at least 1000 and at most 9999, the program will display with formatted date. If one of these conditions are not true, then the program will display the appropriate error message.

The output will be displayed in the format ‘MM/DD/YYYY’. All leading 0’s will be removed The program should return a value of ‘0’ to close the main function.

**Algorithm Flowchart**

A close up of a map

Description automatically generated