USI CHECK CHARACTER ALGORITHM

# The USI check character is calculated using a Luhn Mod N algorithm.

**The character weighting and algorithm pseudo code for calculating it is shown below:**

char[] validChars = {'2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'J', 'K','L', 'M', 'N', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z'}

bool VerifyKey(string key)

{

    if (key.Length != 10)

        return false;

    char checkDigit = GenerateCheckCharacter(key.ToUpper().Substring(0, 9));

    return key[9] == checkDigit;

}

// Implementation of Luhn Mod N algorithm for check digit.

char GenerateCheckCharacter(string input)

{

    int factor = 2;

    int sum = 0;

    int n = validChars.Length;

    // Starting from the right and working leftwards is easier since

    // the initial "factor" will always be "2"

    for (int i = input.Length - 1; i >= 0; i--)

    {

        int codePoint = Array.IndexOf(validChars, input[i]);

        int addend = factor \* codePoint;

        // Alternate the "factor" that each "codePoint" is multiplied by

        factor = (factor == 2) ? 1 : 2;

        // Sum the digits of the "addend" as expressed in base "n"

        addend = (addend / n) + (addend % n);

        sum += addend;

    }

    // Calculate the number that must be added to the "sum"

    // to make it divisible by "n"

    int remainder = sum % n;

    int checkCodePoint = (n - remainder) % n;

    return validChars[checkCodePoint];

}