EAN-13 Design Task

# The Task

When a barcode is scanned it generates at full code that may contain details of the product. The code is normally 12 digits long. In order to check that the code has been scanned correctly, the numbers within it are used to calculate a single digit Check digit. This is added to the end of the 12 digit code to produce a full 13 digit code. When this is scanned the first 12 digits are again used to calculate the check digit. If this does not match the existing digit in the code then, then the code is invalid.

The task is to write a program that will allow the user to either calculate a check digit from an input 12 digit code or to verify that a full 13 digit code is valid (has the correct check digit).

Any such system should include validation of the data entry and provide clear messages to the user.

# Success Criteria

My success criteria are:

1. Menu allows user to select task
2. Selected task runs when chosen
3. Codes are entered and validated as necessary
4. Correct check digit is calculated
5. Codes are verified correctly
6. Appropriate messages are provided

# Algorithms

## Menu:

Print (Select from the following choices

1. Calculated Check digit
2. Verify code)

ValidChoice=False

While ValidChoice=False:

Menuchoice=input()

If Menuchoice=”1”:

ValidChoice=True

Else if Menuchoice=”2”:

ValidChoice=True

Else:

Print(“Not a valid choice. Please enter again”)

If Menuchoice=”1”:

Calculate Check Digit #See algorithm below

Else:

Verify Code #See algorithm below

## Calculate Check digit

validcode=False

While validecode=False:

Code=input(“Please enter 12 digit code”)

If len(Code)!=12:

Print(“Code must be exactly 12 digits long”)

Else:

If Code.isdigit()=False:

Print(“Code must contain only numerical characters”)

Else:

Validcode=True

Total=0

For position in range (0,12):

Value=code[position]

If position MOD 2=1:

Value=Value\*3

Total=Total+Value

Remainder = Total MOD 10

CheckDigit=10-Remainder

FullCode=Code+Str(CheckDigit)

## Verify Code

validcode=False

While validecode=False:

Code=input(“Please enter 13 digit code”)

If len(Code)!=13:

Print(“Code must be exactly 12 digits long”)

Else:

If Code.isdigit()=False:

Print(“Code must contain only numerical characters”)

Else:

Validcode=True

Total=0

For position in range (0,12):

Value=code[position]

If position MOD 2=1:

Value=Value\*3

Total=Total+Value

Remainder = Total MOD 10

CheckDigit=10-Remainder

If CheckDigit=Code[12]:

Print(“Code is valid”)

Else:

Print(“Code is Invalid”)

|  |  |
| --- | --- |
| Variables used | Data type |
| Validchoice | Boolean |
| MenuChoice | String |
| code | string |
| Validcode | Boolean |
| Value | Integer |
| Total | Integer |
| Remainder | Integer |
| Checkdigit | Integer |
| Fullcode | string |

## Test Plan

|  |  |  |
| --- | --- | --- |
| What I am testing | Data Used | Expected result |
| Menu | 1,2,4,A | 1 and 2 should work. 4 and A should expect user to enter choice again |
| Valid code | 123456789012 or 1234567890123 | Code should be accepted |
| Invalid code | 1234567890  12345678901234  123456789o12  123456789o123 | Codes should be rejected as too short, too long or containing letters |
| Calculating check digit | 123456789012  222222222222 | Should return check digit of 8 and 2 respectively |
| Verifying code | 1234567890128  1234567890129 | Should return valid code and invalid code respectively |