GCSE Computing Controlled Assessment

A453 PROGRAMMING PROJECT

Task 1

GITN BARCODE

Objectives

1. The program will take an input and validate it’s length and integers
2. The program will multiply the first 7 numbers alternately by 3,1
3. The program will total these results
4. The program will subtract this sum from its nearest highest multiple of 10
5. The program will compare this to the given 8th number

Pseudocode

```
User INPUT choice for calculate or verify
If Calculate:
        Number Length is 7
        INPUT GTIN
        Call Verify function
Else If Verify:
        Number Length is 8
        INPUT GTIN
        Call verify Function
Verify Function:
        If GTIN length = Length AND is all numeric:
                For a loop of 7 by step of 2:
                        Multiply GTIN at position of counter by 3 and add to total
                        If counter =6:
                                Round  total UP to nearest multiple of 10
                                Subtract total from rounded number = result
                                        If Length = 7:
                                                Print result
                                        Else:
                                                If GTIN at position Length = result:
                                                        Print GTIN is a valid number
                                                Else:
                                                Print GTIN is an invalid number
                        Else:
                                Multiply GTIN at position of counter+1 by 1 and add to total
        Else:
                Print error and return to GTIN input
```

Test Plan

1. Input strings of incorrect length. If rejected, it passes
2. Input strings of letters. If rejected, it passes.
3. Get program to run a valid input. Print out totals at each stage, and check them manually. If they are the same, it passes
4. Again, manually add the totals of a valid input. If they are the same, it passes
5. Manually round the total to the highest 10. If it is the same, it passes
6. Manually collect the distance rounded. If it is the same, it passes.
7. Run the program with a GTIN number taken from a product. If it correctly calculated and verified, it passes

Data Structure

Variable Name	Variable Description	Value
Ask	The choice whether the user wants to verify or calculate	‘c’ or ‘v’
gtin	The GTIN number used	0, and then user inputted
length	The length the GTIN should be	7 or 8
total	The running total of all the multiplications	0, and changes each iteration
checkdig	The 8th digit in a verification	Length -1
rounded	total rounded up to the nearest 10	n/a
result	The 8th digit as the program calculates it	rounded-total
again	The choice of whether the user wants to run the program again	‘n’ or ‘y’

Development

Objective 1 - The program will take an input and validate it's length and integers

```
def start():        ## Main process
 ask = input('Press [c] to calculate the 8th GTIN Number from 7 numbers. \nPress [v] to vertify an 8 digit GTIN
 if ask == 'c' or ask == 'C':
  gtin = 0
  length = 7
  check(length)
 elif ask == 'v' or ask == 'V':
  gtin = 0
  length = 8
  check(length)
 else: print('Error: Please enter either \'c\' or \'v\' ')
 start()
```

```
def check(length):
  print('Enter the', length, 'digit GTIN number')
  gtin = input(': ')
  if len(gtin) == length and gtin.isnumeric() == True:
    total = 0
```

```
else:
  print('Error: Only', length, 'numbers are allowed. Try again ')
  check(length)
```

- Asks the user if they want to calculate or verify
- If Calculate:
  - The length should be 7
- If Verify:
  - The length should be 8
- Go to 'Check' Function
- Input the GTIN number
- If the length is as specified and it is numeric:
  - Continue
- Else:
  - Return to input and try again

Objective 2 - The program will multiply the first 7 numbers alternately by 3,1

```
for counter in range(0, 7, 2):
  total = 0
  total = total + ((int(gtin[counter]))*3)
```

```
total = total + ((int(gtin[counter+1]))*1)
```

- For 7 iterations, stepping ‘counter’ by 2:
- Total is 0
- Total is total + GTIN position ‘counter’ x 3
- Total is total + GTIN position ‘counter+1’ x 1

Objective 3 - The program will total these results

```
total = int(total)
```

- As totals are added in line:
- This converts ‘total’ into an integer value

Objective 4 - The program will subtract this sum from its nearest highest multiple of 10

```
rounded = (int(math.ceil(total / 10.0)) * 10)
result = (rounded - total)
```

- ‘rounded’ is the integer of (the ceiling of (total / 10)) x 10
- ‘result’ is the rounded number - total

Objective 5 - The program will compare this to the given 8th number

```
checkdig = int(gtin[length-1])
```

```
if checkdig == result:
  print(gtin, 'is a Valid Number')
else:
  print(gtin, 'is an Invalid Number')
```

- ‘checkdig’ is the GTIN position of ‘length’ – 1
- If ‘checkdig’ is the same as ‘result’:
  - Display the number as valid
- Else:
  - Display the number as invalid

Testing

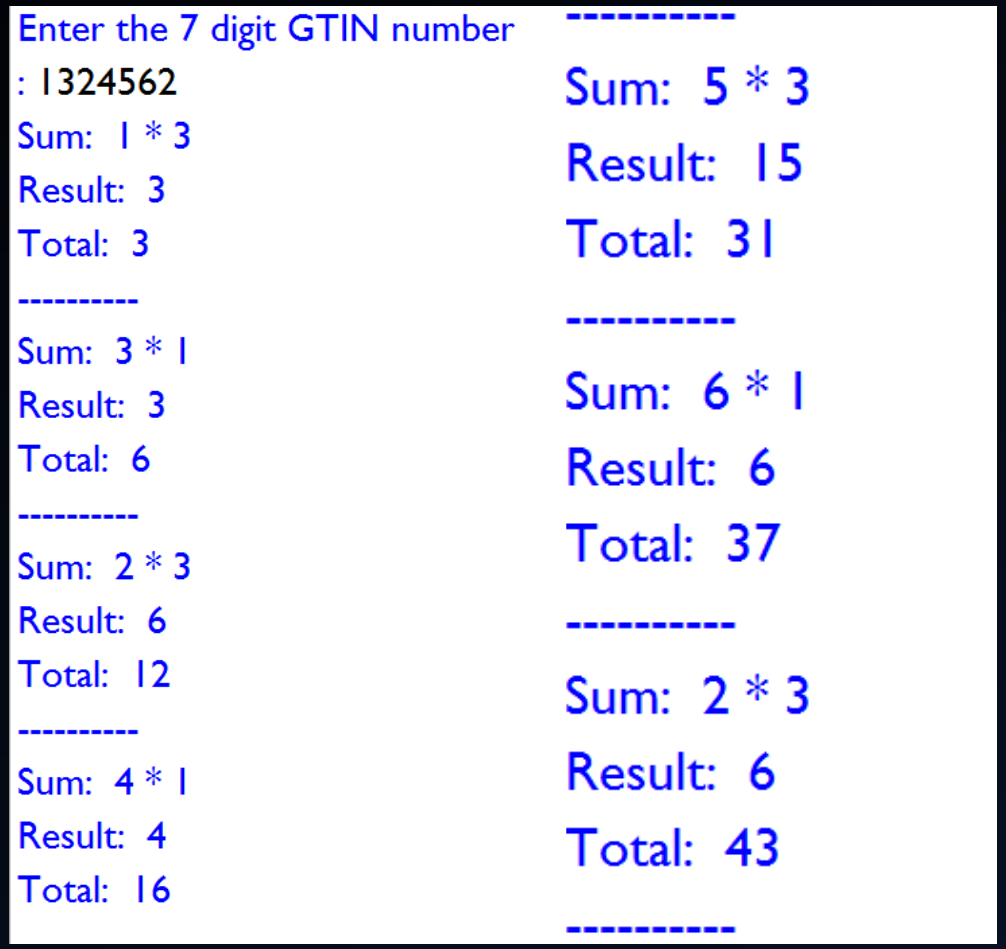
Objective 1 - The program will take an input and validate it's length and integers

```
Enter the 7 digit GTIN number
: 12345
Error: Only 7 numbers are allowed. Try again
Enter the 7 digit GTIN number
: 123456789
Error: Only 7 numbers are allowed. Try again
Enter the 7 digit GTIN number
: abcdefg
Error: Only 7 numbers are allowed. Try again
Enter the 8 digit GTIN number
: 1234567
Error: Only 8 numbers are allowed. Try again
```

Passed

- Input strings of incorrect length.
  - If rejected, it passes
- Input strings of letters.
  - If rejected, it passes.

Objective 2 - The program will multiply the first 7 numbers alternately by 3,1



- Get program to run a valid input. Print out totals at each stage, and check them manually.
  - If they are the same, it passes

Passed

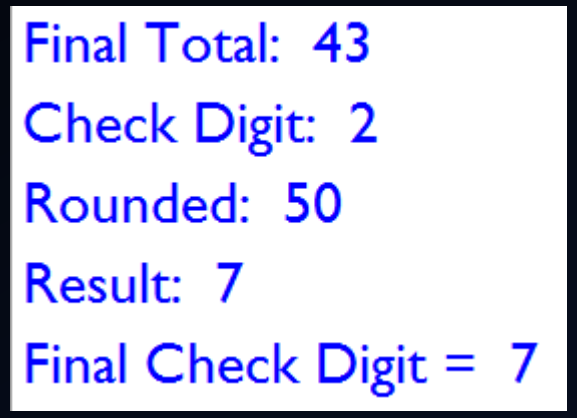
Objective 3 - The program will total these results

- Again, manually add the totals of a valid input.
  - If they are the same, it passes

Passed

Objective 4 - The program will subtract this sum from its nearest highest multiple of 10

- Manually round the total to the highest 10.
  - If it is the same, it passes
- Manually collect the distance rounded.
  - If it is the same, it passes.



Passed

Objective 5 - The program will compare this to the given 8th number


```
Enter the 8 digit GTIN number
:01412871
01412871 is a Valid Number
```

- Run the program with a GTIN number taken from a product.
  - If it correctly calculated and verified, it passes

Passed

Final Program

Raw data

```
import math
import sys
print('GCSE Controlled Assesment A453\nThomas Bass 4869\nTask 1')
def start():
  ask = input('Press [c] to calculate the 8th digit. \nPress [v] to vertify an 8 digit GTIN Number \n')
  if ask == 'c' or ask == 'C':
    length = 7
  elif ask == 'v' or ask == 'V':
    length = 8
  else:
    print('Error: Please enter either \'c\' or \'v\' ')
    start()
  check(length)
def check(length):
  print('Enter the', length, 'digit GTIN number')
  gtin = input(': ')
  if len(gtin) == length and gtin.isnumeric() == True:
    total = 0
    for counter in range(0, 7, 2):
      total = total + ((int(gtin[counter]))*3)
      if counter == 6:
        checkdig = int(gtin[length-1])
        rounded = (int(math.ceil(total / 10.0)) * 10)
        result = (rounded - total)
        if length == 7:
          print('Final Check Digit = ', result)
          print('Whole GTIN-8 Number = ', gtin,result)
          park()
        else:
          if checkdig == result: print(gtin, 'is a Valid Number')
          else: print(gtin, 'is an Invalid Number')
          park()
      else: total = total + ((int(gtin[counter+1]))*1)
  else:
    print('Error: Only', length, 'numbers are allowed. Try again ')
    check(length)
def park():
  again = input('Do you want to calculate or verify another number? \n[n] No [y] Yes: ')
  if again == 'n' or again == 'N':
    sys.exit()
  elif again == 'y' or again == 'Y':
    start()
start()
```

Formatted Photo

```
import math
import sys
print('GCSE Controlled Assesment A453\nThomas Bass 4869\nTask 1')
def start():
  ask = input('Press [c] to calculate the 8th digit. \nPress [v] to vertify an 8 digit GTIN Number \n')
  if ask == 'c' or ask == 'C':
    length = 7
  elif ask == 'v' or ask == 'V':
    length = 8
  else:
    print('Error: Please enter either \'c\' or \'v\' ')
    start()
  check(length)
def check(length):
  print('Enter the', length, 'digit GTIN number')
  gtin = input(': ')
  if len(gtin) == length and gtin.isnumeric() == True:
    total = 0
    for counter in range(0, 7, 2):
      total = total + ((int(gtin[counter]))*3)
      if counter == 6:
        checkdig = int(gtin[length-1])
        rounded = (int(math.ceil(total / 10.0)) * 10)
        result = (rounded - total)
        if length == 7:
          print('Final Check Digit = ', result)
          print('Whole GTIN-8 Number = ', gtin,result)
          park()
        else:
          if checkdig == result: print(gtin, 'is a Valid Number')
          else: print(gtin, 'is an Invalid Number')
          park()
      else: total = total + ((int(gtin[counter+1]))*1)
  else:
    print('Error: Only', length, 'numbers are allowed. Try again ')
    check(length)
def park():
  again = input('Do you want to calculate or verify another number? \n[n] No [y] Yes: ')
  if again == 'n' or again == 'N':
    sys.exit()
  elif again == 'y' or again == 'Y':
    start()
start()
```

Formatted Photo With Comments

```
import math     ## Imports Math and Sys librarys
import sys
print('GCSE Controlled Assesment A453\nThomas Bass 4869\nTask 1')
def start():        ## Defines 'start'
  ask = input('Press [c] to calculate the 8th digit from 7'
  '\nPress [v] to vertify an 8 digit GTIN Number \n')     ## Ask the user if they want to verify or calculate
  if ask == 'c' or ask == 'C':        ## If user chooses to calculate
    length = 7                         ## 'length' is 7
  elif ask == 'v' or ask == 'V':      ## If user chooses to verify
    length = 8                         ## 'length' is 8
  else:                                ## else
    print('Error: Please enter either \'c\' or \'v\' ')     ## Error message
    start()                            ## Return to 'start'
  check(length)                        ## Call 'check' function
def check(length):                     ## Define 'check' function
  print('Enter the', length, 'digit GTIN number')         ## Ask the user to input the GTIN
  gtin = input(': ')                   ## Input 'gtin'
  if len(gtin) == length and gtin.isnumeric() == True:      ## If the length of 'gtin' is equal to 'length' and 'gtin' is numeric
    total = 0                          ## 'total' is 0
    for counter in range(0, 7, 2):  ## For 7 iterations, stepping 'counter' by 2
      total = total + ((int(gtin[counter]))*3)            ## 'total' is 'total' plus 'gtin' position 'counter' times 3
      if counter == 6:                   ## If 'counter' is 6
        checkdig = int(gtin[length-1])                    ## 'checkdig' is 'gtin' position 'length' -1
        rounded = (int(math.ceil(total / 10.0)) * 10)   ## 'rounded' is ceil of 'total' /10.0 * 10 (rounded up to multiple of 10)
        result = (rounded - total)                        ## 'result' is 'rounded' - 'total'
        if length == 7:                                   ## If 'length' is 7
          print('Final Check Digit = ', result)           ## Print 'result'
          print('Whole GTIN-8 Number = ', gtin,result)    ## Print 'gtin'+'result'
          park()                                          ## Call 'park' function
        else:                                             ## Else
          if checkdig == result: print(gtin, 'is a Valid Number')       ## If 'checkdig' equals 'result': GTIN is valid
          else: print(gtin, 'is an Invalid Number')                     ## Else: GTIN is invalid
          park()                                          ## Call 'park' function
      else: total = total + ((int(gtin[counter+1]))*1)
  else:                                                   ## Else
    print('Error: Only', length, 'numbers are allowed. Try again ')     ## Print error message
    check(length)                                         ## Call 'check' function
def park():                                               ## Define 'park' function
  again = input('Do you want to calculate or verify another number? \n[n] No [y] Yes: ')  ## Ask the user to go again
  if again == 'n' or again == 'N':        ## If No:
    sys.exit()                             ## Close program
  elif again == 'y' or again == 'Y':      ## If Yes:
    start()                                ## Call 'start' function
start()                                    ## Call 'start' function
```