Code:

1. Import the turtle library
2. Make a variable called “file\_name”
   1. It asks the user for the name of the file they want to use
3. Make Function named “barcode\_split”
   1. Use one pramer
      1. This pramer is going call variable named “filename”
   2. Then we set up a Turple called “barcodelines”
      1. Which will hold the value for the barcode read
   3. Then we make a variable called “lines”.
      1. Which will open the file by call the variable named “filename”
   4. Then we will make a while statement
      1. Which says if True do the follow next line after this line and if not True than do the follow line after “else:”.
         1. If the while sty true when the loop continues but if false than the loop breaks
   5. Then we will make the variable called “thelines”.
      1. Which read each line in the file given
   6. When we will make an if statement.
      1. Which says if the length of the variable named “theline” is 0, then break.
   7. Next we will set the variable named “theline” to the variable called “theline” but with all part that show \n replace with spaces.
   8. Next we will make the variable called “DBars ”
      1. Which is second number to second last number of variable called “theline”.
   9. Next we will make variable called “Modulo\_Check\_Character”>
      1. Which is the last number of variable called “theline”.
   10. Next we make a line that will append the variable named “DBars” to the turple named “barcodelines”.
   11. Next we make a line that will append the variable named “Modulo\_Check\_Number” to the turple named “barcodelines”
   12. Then we return the value of the turple named “barcodelines”
4. Make a Function named “trueORfalse”.
   1. First we will make a turlpe named “TRUEORFALSE” and set to nothing
   2. Next make a variable named “A” and set it to zero
   3. Next make a variable named “B” and set it to zero
   4. Next make a variable named “index” and set it to zero
   5. Make a while loop
      1. Which says if length of index is less than length of the paramer then run but if not stop running the loop.

while int(index) < len(turple\_of\_barcode):

if index % 2 == 0:

Barcode = str(turple\_of\_barcode[index])

else:

CheckNumber = str(turple\_of\_barcode[index])

Even = 0 #A varable for even#s

Odd = 0 #A varable for odd#s

index2 = 0 #A varable to index

while int(index2) < len(Barcode): # a while so that the code does not error message

if index2 % 2 == 0: # an if statement that finds the second, fourth, sixth, and eigth number in the string

Odd += int(Barcode[index2]) # stores the sum of the odd elements

else:

Even += int(Barcode[index2]) # stores the sun of the even elements

index2 += 1 # add one to the index

Odd2 = Odd \* 3 #Odd2 is a varable for mutipling the odd numbers by 3

Total1 = Odd2 + Even #Adds the total sum for even numbers and odds numbers

Total2 = Total1 % 10 # find the remainer for the first total

Total3 = 10 - int(Total2) # subtracts 10 by the second total

if int(Total3) == int(CheckNumber): # an if statement that take the total and compairs it to the value of the Check

print str(CheckNumber)+ ' ' + str(Total3) + 'True' # if the compair is true, then print true, if not print false.

T = "True"

TRUEORFALSE.append(T)

else:

print str(CheckNumber) + ' ' + str(Total3) + 'False'

F = "False"

TRUEORFALSE.append(F)

index += 1

return TRUEORFALSE

def barcodetogether(A):

"""Take all the split parts of the barcodes and puts them togther"""

Listofbarcodes = []

index = 0

while int(index) < len(A):

if index % 2 == 0:

List = ('')

P1 = A[index]

indexA = index + 1

P2 = A[indexA]

List += str(P1) + str(P2)

Listofbarcodes.append(List)

index += 1

return Listofbarcodes

def Dictionary(A):

"""Decodeds the Bracode into Binart Code"""

BarcodeDecodedList = []

for i in A:

BarcodeC = str(i)

PL = BarcodeC[0:6]

PR = BarcodeC[6:12]

PartLeft = {0:'0001101',1:'0011001',2:'0010011',3:'0111101',4:'0100011',5:'0110001',6:'0101111',7:'0111011',8:'0110111',9:'0001011'}

PartRight = {0:'1110010',1:'1100110',2:'1101100',3:'1000010',4:'1011100',5:'1001110',6:'1010000',7:'1000100',8:'1001000',9:'1110100'}

PLdecoded = ('')

PRdecoded = ('')

for A in PL:

number = int(A)

PLdecoded += str(PartLeft[number])

for A in PR:

number = int(A)

PRdecoded += str(PartRight[number])

BarcodeDecoded = str(PLdecoded) + str(PRdecoded)

BarcodeDecodedList.append(BarcodeDecoded)

return BarcodeDecodedList

def printturtle(A, B):

"""Prints all barcodes and Nonbarcodes but for Nonbarcodes the screen prints an error message saying 'Error: This barcode is not acceptable (Check function trueORfalse)'"""

wn=turtle.Screen()

barcode=turtle.Turtle()

barcode.color("black")

for oneorzero in A:

if oneorzero == 1:

barcode.right(90)

barcode.forward(10)

barcode.right(180)

barcode.forward(10)

barcode.right(90)

else:

barcode.forward(1)

#-------------------------------------------------------------------------------

#Calling Zone:

file\_name = raw\_input("file name") #Asks the user which file they should use.

#look for file name "upc-input1.txt, upc-input2.txt"

BarcodeCheckNumber = barcode\_split(file\_name) #set the turple to the a varable called BarcodeCheckNumber

print BarcodeCheckNumber #For testing

(Barcode1,Check1,Barcode2,Check2,Barcode3,Check3,Barcode4,Check4,Barcode5,Check5,Barcode6,Check6,Barcode7,Check7,Barcode8,Check8,Barcode9,Check9,Barcode10,Check10,Barcode11,Check11,Barcode12,Check12) = BarcodeCheckNumber

#Takes the turple and sets every element as a varable

print trueORfalse(barcode\_split(file\_name))

# check all possablity to see if the each string is truely a barcode

print barcodetogether(barcode\_split(file\_name))

#For testing

print Dictionary(barcodetogether(barcode\_split(file\_name)))

'''This line input every mathfied tranlated barcode into a tranlate binary code string'''

#printturtle(Dictionary(i)

'''This comitted out line is to preform the turtle'''