

Aufgabe 2: Nummernmerker

Team-ID: #####

Team-Name: ###

Bearbeiter/-innen dieser Aufgabe:
Snocember

23. November 2019

Inhaltsverzeichnis

Lösungsidee.....	1
Umsetzung.....	1
Beispiele.....	2
Quellcode.....	2

Lösungsidee

Die Aufgabe lautet, Sarahs Mutter zu helfen, indem man die Nummernketten, die eingelesen werden, in Blöcke aufteilt, die zwei bis vier Ziffern besitzen. Dabei soll die Anzahl der Blöcke, die mit Null anfangen, möglichst gering sein.

Mein Lösungsansatz ist, dass eine Zahlenkette eingelesen wird und nacheinander getestet wird, welche Blöcke (2 bis 4-Ziffer-Blöcke) man verwenden kann, damit möglichst keine Null vor einem Block ist. z.B. dass aus ,33300‘ ,33-300‘ wird, damit die Null vor einem Block vermieden wird (,3330-0‘). Dann wird durch weitere Funktionen vermieden, dass der String nicht mit nur einer Ziffer endet, oder dass etwas nicht aufgeht.

Umsetzung

Die Lösungsidee wird in Python 3.7.0 implementiert. Es wurde MacOS Mojave mit Terminal.app genutzt.

Die Textdatei ,nummern.txt‘ wird eingelesen und deren Nummernketten in eine Liste eingetragen. Alle Nummernketten werden nacheinander in einer For-Schleife betrachtet, in der eine While-Schleife sich befindet. Die While-Schleife arbeitet sich durch die Zahlenkette bis sie komplett neu aufgeteilt ist. Wenn das Programm am Ende des Zahlenstrings ist, tritt ein IndexError auf, der abgefangen wird. Darauf startet die For-Schleife erneut die While-Schleife mit einem anderen Zahlenstring.

Das Programm möchte vorrangig 4er-Blöcke (damit ist z.B. ,1111‘ gemeint) bilden. Ist nach einer potentiellen nächsten 4er-Kette eine Null (,1000-0‘), wird versucht ob nach einem 3er-Block auch noch eine Null ist (,100-0‘), das selbe mit einem 2er-Block (,10-0‘). Ist nach einem dieser Möglichkeiten keine Null, wird der jeweilige Block gebildet und das Programm springt weiter in dem Zahlenstring. Können die drei Möglichkeiten aufgrund einer darauffolgenden Null nicht genutzt werden, wird trotz einer darauffolgenden Null ein 4er-Block verwendet.

Tritt der Fall auf, dass man fast am Ende des Strings ist und ein 4er-Block möglich wäre, aber dahinter nur noch eine Ziffer ist (z.B. 9876-12345), werden die letzten übrigen 5 Ziffern in einen 2er und 3er-Block umgewandelt.

(z.B 9876-12-345) Ist hinter dem 2er-Block eine Null, wird die Reihenfolge der Blöcke zu zuerst 3er und dann einen 2er-Block geändert. (z.B. 9876-12034 –> 9876-120-34)

Beispiele

Wir rufen das Programm auf. (Die fünfte Zahlenkette ist ein Testbeispiel von mir.)

```
$ python3 nummernmerker.py
```



EINGABE

```
['005480000005179734', '03495929533790154412660', '5319974879022725607620179',  
'9088761051699482789038331267', '01100000001100010011111101011']
```

ERGEBNIS (Gelb markiert sind alle Blöcke, die mit 0 anfangen.)

```
0054-8000-0005-1797-34  
0349-5929-5337-9015-4412-660  
5319-9748-7902-2725-6076-20-179  
9088-7610-5169-9482-7890-3833-1267  
01-1000-0000-1100-0100-1111-1110-1011
```

Quellcode

```
# -*- coding: utf8 -*-  
#!/usr/bin/env python3  
  
# ----#---- Importe und Initialisierungen ----#----  
import sys                                     # sys.exit() falls Datei nicht gefunden.  
cyan = "\033[96m"  
[reset, pink, gelb] weitere ascii-farb-codes  
  
# ----#----#---- EINLESEN ----#----  
try:  
    file = open("./nummern.txt")               # Einlesen der Datei  
    inhalt = file.readlines()                 # "  
except FileNotFoundError:                     # Falls die Datei nicht gefunden wird, Fehler  
    [...]                                     # Falls die Datei nicht gefunden wird, endet das Programm  
    sys.exit(0)  
print(pink+"EINGABE"+reset)  
for i in range(0, len(inhalt)):  
    inhalt[i] = inhalt[i].strip()             # Leerzeichen um den Strings entfernen  
print(inhalt)                                # Ausgabe der eingelesenen Datei  
# ----#----#---- AUSWERTEN ----#----  
print(pink+"ERGEBNIS"+reset+" (Gelb markiert sind alle Blöcke, die mit 0 anfangen.)")  
anzahl = len(inhalt)                          # Die Anzahl der Nummernketten wird bestimmt  
for u in range(0, anzahl):                   # For-Schleife, die alle Nummern nacheinander bearbeitet  
    ok = 1                                    # ,ok' um die schleife aufrechts zu erhalten  
    nr = 0                                    # ,nr' = aktuelle Stelle im String  
    string = ""
```

```

while ok:                                     # Die Schleife läuft so lange, bis die Zahlenreihe bearbeitet ist
    try:                                       # Try, except-Block um IndexError abzufangen
        if inhalt[u][nr] == "0": string = string+gelb # Wenn der String mit 0 beginnt, wird er gelb gefärbt
        if inhalt[u][nr+4] != "0":           # Wenn NICHT nach einem 4er-Block eine 0 ist:
            if (len(inhalt[u])-1)-(nr+3) == 1: # Falls hinter einem 4er-Block nur noch eine Zahl ist:
                if inhalt[u][nr+2] == "0":     # Falls die 3te stelle ein 0 ist:
                    string = string+inhalt[u][nr]+inhalt[u][nr+1]+inhalt[u][nr+2]+"-"+inhalt[u][nr+3]
                                                +inhalt[u][nr+4]+reset    # - > 110-22
                else:
                    string = string+inhalt[u][nr]+inhalt[u][nr+1]+"-"+inhalt[u][nr+2]+inhalt[u][nr+3]
                                                +inhalt[u][nr+4]+reset    # - > 11-222
            ok = 0                             # Da Zahlenkette zuende ist; ok=0; while-Schleife endet
        else:                                 # Normaler 4er Block:
            string = string+inhalt[u][nr]+inhalt[u][nr+1]+inhalt[u][nr+2]+inhalt[u][nr+3]+reset+"-" # 1111
            nr = nr+4                          # Die Position im String wird verändert.
        elif inhalt[u][nr+3] != "0":          # Wenn NICHT nach einem 3er-Block eine 0 ist:
            string = string+inhalt[u][nr]+inhalt[u][nr+1]+inhalt[u][nr+2]+reset+"-" # 111
            nr = nr+3                          # Die Position im String wird verändert.
        elif inhalt[u][nr+2] != "0":          # Wenn NICHT nach einem 2er-Block eine 0 ist:
            string = string+inhalt[u][nr]+inhalt[u][nr+1]+reset+"-" # 11
            nr = nr+2                          # Die Position im String wird verändert.
        else:                                 # Wenn die oberen Möglichkeiten nicht zutreffen:
            if (len(inhalt[u])-1)-(nr+3) == 1: # Wenn hinter dem nächsten 4er-Block noch 1 Zahl ist:
                if inhalt[u][nr+2] == "0":     # Wenn die 3te Ziffer eine 0 ist:
                    string = string+inhalt[u][nr]+inhalt[u][nr+1]+inhalt[u][nr+2]+"-"+gelb+inhalt[u][nr+3]
                                                +inhalt[u][nr+4]+reset    # - > 110-22
                else:
                    string = string+inhalt[u][nr]+inhalt[u][nr+1]+"-"+inhalt[u][nr+2]+inhalt[u][nr+3]
                                                +inhalt[u][nr+4]+reset    # - > 11-222
            ok = 0                             # ok = 0, die Schleife endet
        else:                                 # Falls alles nicht zutrifft ein 4er-Block
            nr = nr+4                          # Die Position im String wird verändert.
            string = string+inhalt[u][nr-4]+inhalt[u][nr-3]+inhalt[u][nr-2]+inhalt[u][nr-1]+reset+"-" # 1111
    except IndexError:                        # Ein IndexError wird abgefangen, wenn das Ende des...
        for k in range((anzahl)-(anzahl-nr), len(inhalt[u])): # ... Strings erreicht ist.
            string = string+inhalt[u][k]      # Dann wird der Rest des Strings angehängt
        string = string+reset
        ok = 0                               # ok = 0, die schleife endet
# ----#----#---- ERGEBNIS ----#----
print(string)                               # Der fertig umgewandelte String einer Zahlenkette wird
                                           # ausgegeben. z.B. 110-1245-693-80

```