Handwriting recognition

Optical character recognition, usually abbreviated to OCR, is the mechanical or electronic conversion of scanned or photographed images of typewritten or printed text into machine-encoded/computer-readable text. It is a common method of digitizing printed texts so that they can be electronically edited, searched, stored more compactly, displayed on-line, and used in machine processes such as machine translation, text-to-speech, key data extraction and text mining. OCR is a field of research in pattern recognition, artificial intelligence and computer vision.

Assignment

In this exercise we work with text files of which the lines represent a number of handwritten characters. The manuscript is printed in a font that meets the following conditions:

- all lines have the same length (including spaces)
- equal handwritten characters are always displayed in the same way
- different handwritten characters are never displayed in the same way
- the representation of a handwritten character never contain empty columns
- there are one or more blank columns between two successive handwritten characters
- before the first and after the last hand-written character there are zero or more blank columns

An empty column is a column of the lines with handwritten text consisting only of spaces.

The following example displays the content of a text file that contains a handwritten version of the letters ocrrococo. Click here to view a graphical representation of the segmentation of the text file. These segments are indicated with a dark gray background, and the lines that are not included in the string representation of segments are crossed out with a red line.

```
 ##  ### # ## # ##  ##  ### ##  ### ##
# ##  ##  ## ## # ##  # ##  # #
# ##  #  #  # ##  # ##  # #
# ##  #  #  # ##  # ##  # #
# ##  #  #  # ##  # ##  # #
 ##  ### #  #  ##  ### ##  ### ##
```

The challenge is to convert the contents of the text file to the corresponding series of ASCII characters. To do this, just follow these steps:

- OCR often starts with *segmenting* handwritten text, where the handwriting is divided into individual handwritten characters (in this context referred to as *segments*). Write a function segmentation to which the location of a text file with handwritten text is to be passed as an argument. The function should return a list of all the segments that appear in the text file. A segment is formed by one or more consecutive non-empty columns which are enclosed between two blank columns. Also consecutive non-blank columns in front and at the end of

the text file form a segment. A segment is represented as a string by putting all the rows of the segment after one another and separating them by *newline* characters. Rows in which all the characters in the original text file are spaces (not only within the segment, but the entire line) are not included in the string representation of the segment. It is the string representation of a segment that should be included in the list that is returned by the function.

- Use `segmentation` to write a function `OCR` that can be used to convert a handwritten text to to the corresponding ASCII characters. The handwritten text is stored in a text file, of which the location must be passed to the function as an argument. The part of the file location for the first item is a word whose ASCII characters also form the initial characters of the handwritten text. The file `ocr.txt` yields than for example gives the word `ocr`. The remaining characters of the handwritten text also occur in the word. The function is required to return a string containing ASCII characters that correspond to the word that is composed of the remaining handwritten characters.

Example

In the following example session we assume that the example file ocr.txt, sportmannen.txt and romanheld.txt are in the current directory.

```
>>> segment = segmentation('ocr.txt')
>>> segment[0]
' ## \\n#  #\\n#  #\\n#  #\\n#  #\\n ## '
>>> print(segment[0])
 ##
#  #
#  #
#  #
#  #
 ##
>>> print(segment[1])
 ###
#
#
#
#
 ###
>>> print(segment[2])
# ##
## #
#
#
#
#
>>> print(segment[3])
# ##
## #
#
#
#
#
>>> print(segment[-1])
 ##
#  #
#  #
```

```
# #
# #
 ##

>>> OCR('ocr.txt')
'rococo'
>>> OCR('sportmannen.txt')
'marmer'
>>> OCR('romanheld.txt')
'emerald'
```

Click on the links below to view a graphical representation of the segmentation of the text files. These segments are indicated with a dark gray background, and the lines that are not included in the string representation of segments crossed out with a red line.

- ocr.txt
- sportmannen.txt
- romanheld.txt

Optical character recognition (OCR) is een techniek die kan gebruikt worden om ingescande afbeeldingen van handgeschreven, getypte of afgedrukte tekst om te zetten naar ASCII-tekst die leesbaar is voor een computer. Deze techniek wordt vaak gebruikt om afgedrukte teksten te digitaliseren zodat men ze elektronisch kan doorzoeken, compacter kan opslaan, online kan weergeven, of ze automatisch kan laten vertalen of voorlezen door een computer. OCR vormt e e n onderzoeksdomein binnen de patroonherkenning, artificiële intelligentie en computervisualisatie.

Opgave

In deze opgave werken we met tekstbestanden, waarvan de regels een aantal handgeschreven karakters voorstellen. Het handschrift staat telkens in een lettertype dat voldoet aan volgende voorwaarden:

- alle regels hebben dezelfde lengte (inclusief spaties)
- gelijke handgeschreven karakters worden steeds op dezelfde manier weergegeven
- verschillende handgeschreven karakters worden nooit op dezelfde manier weergegeven
- de voorstelling van een handgeschreven karakter bevat nooit lege kolommen
- tussen twee opeenvolgende handgeschreven karakters staan één of meer lege kolommen
- voor het eerste en na het laatste handgeschreven karakter staan nul of meer lege kolommen

Een lege kolom is een kolom van de regels met handgeschreven tekst die enkel bestaat uit spaties.

Hieronder wordt bijvoorbeeld de inhoud weergegeven van een tekstbestand dat een handgeschreven versie van de letters ocrrococo bevat. Klik hier om een grafische voorstelling van de segmentatie van dit tekstbestand te bekijken. Hierbij worden de segmenten aangegeven met een donkergrijze achtergrond, en worden de regels die niet opgenomen worden in de stringvoorstelling van segmenten met een rode lijn doorstreept.

```
 ##  ### # ## # ##  ### ##  ### ##
# # #  ## # ## # # #  # # #  # #
# # #  #  #  # # #  # # #  # #
# # #  #  #  # # #  # # #  # #
# # #  #  #  # # #  # # #  # #
 ##  ### #  #  ##  ### ##  ### ##
```

De opgave bestaat erin om de inhoud van het tekstbestand om te zetten naar de corresponderende reeks ASCII karakters. Hiervoor ga je als volgt te werk:

- OCR start vaak met het *segmenteren* van de handgeschreven tekst, waarbij het handschrift wordt opgedeeld in individuele handgeschreven karakters (in deze context *segmenten* genoemd). Schrijf daarvoor een functie segmentatie waaraan de locatie van een tekstbestand met handgeschreven tekst als argument moet doorgegeven worden. De functie moet een lijst teruggeven met alle segmenten die in het tekstbestand voorkomen. Daarbij wordt een segment gevormd door één of meer opeenvolgende niet-lege kolommen die tussen twee lege kolommen ingesloten zitten. Ook opeenvolgende niet-lege kolommen vooraan of achteraan het tekstbestand vormen een segment. Een segment wordt voorgesteld als een string door alle rijen van het segment achter elkaar te zetten en van elkaar te scheiden door *newline* karakters. Rijen waarop in het oorspronkelijke tekstbestand alle karakters spaties zijn (niet enkel binnen het segment, maar over de volledige regel), worden niet opgenomen in de stringvoorstelling van het segment. Het is de stringvoorstelling van een segment die moet opgenomen worden in de lijst die door de functie wordt teruggegeven.
- Gebruik de functie segmentatie om een functie OCR te schrijven die kan gebruikt worden om een handgeschreven tekst om te zetten naar de corresponderende ASCII karakters. De handgeschreven tekst zit opgeslagen in een tekstbestand, waarvan de locatie als argument aan de functie moet doorgegeven worden. Het deel van de bestandslocatie voor het eerste punt vormt een woord, waarvan de ASCII karakters meteen ook de beginkarakters vormen van de handgeschreven tekst. De bestandsnaam ocr.txt levert dan bijvoorbeeld het woord ocr op. De resterende karakters van de handgeschreven tekst komen allemaal ook in het woord voor. De functie moet een string teruggeven die de ASCII karakters bevat die corresponderen met het woord dat bestaat uit de resterende handgeschreven karakters.

Voorbeeld

Bij onderstaande voorbeeldsessie gaan we ervan uit dat de tekstbestanden ocr.txt, sportmannen.txt en romanheld.txt zich in de huidige directory bevinden.

```
>>> segment = segmentatie('ocr.txt')
>>> segment[0]
' ## \\n#  #\\n#  #\\n#  #\\n#  #\\n ## '
>>> print(segment[0])
 ##
# #
# #
# #
# #
 ##
>>> print(segment[1])
 ###
```

```
#
#
#
#
 ###
>>> print(segment[2])
# ##
## #
#
#
#
#
>>> print(segment[3])
# ##
## #
#
#
#
#
>>> print(segment[-1])
 ##
#  #
#  #
#  #
#  #
 ##

>>> OCR('ocr.txt')
'rococo'
>>> OCR('sportmannen.txt')
'marmer'
>>> OCR('romanheld.txt')
'emerald'
```

Klik op onderstaande links om een grafische voorstelling van de segmentatie van de tekstbestanden te bekijken. Hierbij worden de segmenten aangegeven met een donkergrijze achtergrond, en worden de regels die niet opgenomen worden in de stringvoorstelling van segmenten met een rode lijn doorstreept.

- ocr.txt
- sportmannen.txt
- romanheld.txt