# Uitwerking opdracht

Opdracht : *Nobelprijzen*

Weeknummer : *week 7*

Studentnummer : *s1101951*

Naam student : *Reza Naser*

Specialisatie : *SE*

Pogingnummer : *1*

# Vraagstelling

*Een functie prijzen te schrijven, die kan gebruikt worden om een aantal prijzen uit het CSV-bestand uit te schrijven die voldoen aan een reeks vooropgestelde criteria. De locatie van het CSV-bestand moet als verplicht argument aan de functie doorgegeven worden. Daarnaast heeft de functie nog een reeks criteria die opleggen waaraan de prijzen moeten voldoen om uitgeschreven te worden*

# Specificatie

## Invoer

* *prijs (str): prijs moet tot opgegeven soort (nobelprijs, abelprijs of turing award) behoren*
* *discipline (str): prijs werd in opgegeven discipline uitgereikt*
* *jaar (int): prijs werd in opgegeven jaar uitgereikt*
* *nationaliteit (str): minstens één van de laureaten had de opgegeven nationaliteit*
* *laureaten (int): het aantal laureaten waaraan de prijs werd uitgereikt is gelijk aan het opgegeven aantal*
* *motivering (str): motivering van de prijs bevat het tekstfragment dat aan dit argument werd doorgegeven*

## Uitvoer

*Voor elke prijs die aan bovenvernoemde criteria voldoet moet het soort prijs worden uitgeschreven, gevolgd door de tekst " voor de ", de discipline met een hoofdletter vooraan en de rest kleine letters, nog een spatie en het jaar tussen rond haken. Daarna volgt een dubbelpunt, een spatie en de namen van de laureaten zoals ze in het gegeven CSV bestand worden beschreven. Bij het uitschrijven van de velden moeten spaties vooraan en achteraan verwijderd worden. De oorspronkelijke volgorde waarmee de prijzen staan opgelijst in het gegeven CSV bestand moet bij het uischrijven behouden blijven. Bij criteria waarbij er een vergelijking tussen strings moet gemaakt worden, moet men geen onderscheid maken tussen hoofdletters en kleine letters.*

## Verband tussen in- en uitvoer

*Het programma berekent hoeveel kolommen in een rij overeenkomen met de zoektermen. Als aantal gevonden kolommen overeenkomt met aantal zoektermen dan wordt dat rij toegevoegd aan een lijst van gevonden rijen. Als alle rijen is het CSV-bestand zijn gecontroleerd, worden de gevonden rijen getoond op het scherm.*

## Beperkingen

* *prijs (str): prijs moet tot opgegeven soort (nobelprijs, abelprijs of turing award) behoren*
* *discipline (str): prijs werd in opgegeven discipline uitgereikt*
* *jaar (int): prijs werd in opgegeven jaar uitgereikt*
* *nationaliteit (str): minstens één van de laureaten had de opgegeven nationaliteit*
* *laureaten (int): het aantal laureaten waaraan de prijs werd uitgereikt is gelijk aan het opgegeven aantal*
* *motivering (str): motivering van de prijs bevat het tekstfragment dat aan dit argument werd doorgegeven*

## Voorbeelden (testscenario’s)

*Invoer: prijzen('prijzen.csv', prijs='nobelprijs', jaar=1994)*

*Uitvoer: Nobelprijs voor de Economie (1994): John Harsanyi (VS), John Forbes Nash (VS), Reinhard Selten (Dui)*

*Nobelprijs voor de Geneeskunde (1994): Alfred G. Gilman (VS), Martin Rodbell (VS)*

*Nobelprijs voor de Literatuur (1994): Kenzaburo Oë (Jap)*

*Nobelprijs voor de Natuurkunde (1994): Bertram Neville Brockhouse (Can)*

*Nobelprijs voor de Natuurkunde (1994): Clifford Glenwood Shull (VS)*

*Nobelprijs voor de Scheikunde (1994): George Andrew Olah (VS)*

*Nobelprijs voor de Vrede (1994): Yasser Arafat (Pal), Shimon Peres (Isr), Yitzhak*

*Invoer: prijzen('prijzen.csv', nationaliteit='bel')*

*Uitvoer: Nobelprijs voor de Vrede (1904): Institut de Droit International (Bel)*

*Nobelprijs voor de Vrede (1909): Auguste Beernaert (Bel)*

*Nobelprijs voor de Literatuur (1911): Maurice Maeterlinck (Bel)*

*Nobelprijs voor de Vrede (1913): Henri La Fontaine (Bel)*

*Nobelprijs voor de Geneeskunde (1919): Jules Bordet (Bel)*

*Nobelprijs voor de Geneeskunde (1938): Corneille Jean FranÃ§ois Heymans (Bel)*

*Nobelprijs voor de Vrede (1958): Georges Pire (Bel)*

*Nobelprijs voor de Geneeskunde (1974): Albert Claude (Bel), Christian de Duve (Bel), George E. Palade (Roe)*

*Nobelprijs voor de Scheikunde (1977): Ilya Prigogine (Bel)*

*Nobelprijs voor de Vrede (1999): Artsen zonder Grenzen (Bel)*

*Abelprijs voor de wiskunde (2008): John G. Thompson (VS), Jacques Tits (Bel)*

# Ontwerp

1. *Geef de zoektermen mee in de prijzen functie als argumenten.*
2. *De methode controleert hoeveel keer de zoektermen overeenkomen met de kolommen in één rij.*
3. *Als aantal gevonden kolommen overeenkomen met de aantal zoektermen wordt dat rij toegevoegd in een lijst met de gevonden rijen.*
4. *De gevonden rijen woorden getoond op het scherm.*

# Pseudocode

*import csv*

*import re*

*def main():*

*prijzen('prijzen.csv', prijs='turing award', nationaliteit='gri', discipline='INformatica')*

*def prijzen(location, prijs=None, discipline = None, jaar=None, laureaten=None, motivering=None, nationaliteit = None):*

*#Met deze open ik het CSV bestand*

*with open(location, 'r') as csvfile:*

*#De ingevoerde prijs moet één van deze prijzensoorten zijn*

*prijzenSoort = ['nobelprijs', 'abelprijs', 'turing award']*

*for prijsInLijst in prijzenSoort:*

*if prijs != None and prijs == prijsInLijst:*

*prijs = prijs.title()*

*if (jaar != None):*

*jaar = str(jaar)*

*if discipline != None:*

*discipline = discipline.lower()*

*#Zet de waarde van ingevoerde nationaliteit om naar de HoofdLetters*

*if(nationaliteit != None):*

*nationaliteit = nationaliteit.upper()*

*#Een lijst van alle argumenten*

*arguments=[prijs, discipline, jaar,laureaten,motivering, nationaliteit]*

*#Een lijst van argumenten die een waarde hebben behalve None*

*argumentsAfterCheck = []*

*for arg in arguments:*

*if(arg != None):*

*argumentsAfterCheck.append(arg)*

*#Een lijst van alle gevonden rijen die voldoen aan de zoektermen*

*rijenVoldoenAanZoekTermen = geefGezochteRijen(csvfile, argumentsAfterCheck)*

*#Deze forloop print alle gevonden rijen*

*for item in rijenVoldoenAanZoekTermen:*

*eenRij = item[0] + ' voor de ' + item[1].title() + ' (' + str(item[2]) + '): ' + str(item[3])*

*#De spaties vooraan en achteraan van de string worden verwijderd.*

*eenRij = eenRij.lstrip()*

*print(eenRij.lstrip())*

*#Deze functie krijg het CSV bestand en de ingevoerde argumenten mee en controlleert*

*#Welke rijen in het bestand overeenkomen met de zoektermen. Daarna voegt die dat rij toe aan een*

*#Lijst. Als alle rijen zojn doorgelopen stuurt deze functie dat lijst terug.*

*def geefGezochteRijen(csvfile, argumentsAfterCheck):*

*readCSV = csv.reader(csvfile, delimiter=';')*

*rijen = []*

*#Deze forloop doorloopt alle rijen in csvfile*

*for row in readCSV:*

*#Deze splits de winnaars van de prijs. Als er meerder winnars zijn wordt er een*

*#Aparte object van gemaakt.*

*lijstLaureaten = row[3].split(",")*

*lengteLaureaten = len(lijstLaureaten)*

*#Deze variabele krijgt alle nationaliteiten die zich in een rij bevinden.*

*nationaliteiten = geefNationaliteitenInEenRij(lengteLaureaten, row[3], lijstLaureaten)*

*#Deze variable krijgt aantal overeenkomende kolomen.*

*aantalOvereenKomendeArgumenten = geefAantalOvereenkomendeKolomen(argumentsAfterCheck, row, lengteLaureaten, nationaliteiten)*

*#Als de lengte gelijk is aan de lengte van de gevraagde argumenten*

*if (aantalOvereenKomendeArgumenten == len(argumentsAfterCheck)):*

*#Deze rij word toegevoegd aan de lijst*

*rijen.append(row)*

*return rijen*

*#Deze functie knipt de nationaliteiten in een rij en zet ze in een string.*

*#Dit doet door de waardes binnen de parentheses te knippen.*

*#Aan het einde stuurt deze functie de strings met alle nationaliteiten in een rij terug.*

*def geefNationaliteitenInEenRij(lengteLaureaten, rij, lijstLaureaten):*

*nationaliteiten = ''*

*nationaliteitTeller = 0*

*while nationaliteitTeller < lengteLaureaten:*

*if (rij != 'laureaat'):*

*nationaliteiten = nationaliteiten + re.search('\(([^)]+)', lijstLaureaten[nationaliteitTeller]).group(*

*1) + ' '*

*nationaliteiten = nationaliteiten.upper()*

*nationaliteitTeller += 1*

*return nationaliteiten*

*#Deze functie controlleert hoeveel kolomen in een rij overeenkomen met de zoektermen.*

*#Deze wordt bijgehouden door elke keer als er iets overeenkomt de variabele*

*#aantalOvereenKomendeArgumenten met een opgehoogd. Aan het eind stuurt deze functie dat variabele terug.*

*def geefAantalOvereenkomendeKolomen(argumentsAfterCheck, row, lengteLaureaten, nationaliteiten):*

*teller = 0*

*aantalOvereenKomendeArgumenten = 0*

*while (teller < len(argumentsAfterCheck)):*

*for kolom in row:*

*if row.index(kolom) == 3:*

*if argumentsAfterCheck[teller] == lengteLaureaten:*

*aantalOvereenKomendeArgumenten += 1*

*elif type(argumentsAfterCheck[teller]) != int \*

*and argumentsAfterCheck[teller] in nationaliteiten:*

*aantalOvereenKomendeArgumenten += 1*

*elif row.index(kolom) == 4 and type(argumentsAfterCheck[teller]) != int and \*

*argumentsAfterCheck[teller] in kolom:*

*aantalOvereenKomendeArgumenten += 1*

*elif type(argumentsAfterCheck[teller]) != int and \*

*argumentsAfterCheck[teller] == kolom:*

*aantalOvereenKomendeArgumenten += 1*

*teller += 1*

*return aantalOvereenKomendeArgumenten*

*if \_\_name\_\_ == '\_\_main\_\_':*

*main()*

# Code

**import** csv  
**import** re  
  
**def** main():  
 prijzen(**'prijzen.csv'**, prijs=**'turing award'**, nationaliteit=**'gri'**, discipline=**'INformatica'**)  
  
**def** prijzen(location, prijs=**None**, discipline = **None**, jaar=**None**, laureaten=**None**, motivering=**None**, nationaliteit = **None**):  
 *#Met deze open ik het CSV bestand* **with** open(location, **'r'**) **as** csvfile:  
  
 *#De ingevoerde prijs moet één van deze prijzensoorten zijn* prijzenSoort = [**'nobelprijs'**, **'abelprijs'**, **'turing award'**]  
 **for** prijsInLijst **in** prijzenSoort:  
 **if** prijs != **None and** prijs == prijsInLijst:  
 prijs = prijs.title()  
 **if** (jaar != **None**):  
 jaar = str(jaar)  
 **if** discipline != **None**:  
 discipline = discipline.lower()  
  
 *#Zet de waarde van ingevoerde nationaliteit om naar de HoofdLetters* **if**(nationaliteit != **None**):  
 nationaliteit = nationaliteit.upper()  
 *#Een lijst van alle argumenten* arguments=[prijs, discipline, jaar,laureaten,motivering, nationaliteit]  
  
 *#Een lijst van argumenten die een waarde hebben behalve None* argumentsAfterCheck = []  
 **for** arg **in** arguments:  
 **if**(arg != **None**):  
 argumentsAfterCheck.append(arg)  
  
 *#Een lijst van alle gevonden rijen die voldoen aan de zoektermen* rijenVoldoenAanZoekTermen = geefGezochteRijen(csvfile, argumentsAfterCheck)  
  
 *#Deze forloop print alle gevonden rijen* **for** item **in** rijenVoldoenAanZoekTermen:  
 eenRij = item[0] + **' voor de '** + item[1].title() + **' ('** + str(item[2]) + **'): '** + str(item[3])  
  
 *#De spaties vooraan en achteraan van de string worden verwijderd.* eenRij = eenRij.lstrip()  
  
 print(eenRij.lstrip())  
  
*#Deze functie krijg het CSV bestand en de ingevoerde argumenten mee en controlleert  
#Welke rijen in het bestand overeenkomen met de zoektermen. Daarna voegt die dat rij toe aan een  
#Lijst. Als alle rijen zojn doorgelopen stuurt deze functie dat lijst terug.***def** geefGezochteRijen(csvfile, argumentsAfterCheck):  
 readCSV = csv.reader(csvfile, delimiter=**';'**)  
 rijen = []  
  
 *#Deze forloop doorloopt alle rijen in csvfile* **for** row **in** readCSV:  
 *#Deze splits de winnaars van de prijs. Als er meerder winnars zijn wordt er een  
 #Aparte object van gemaakt.* lijstLaureaten = row[3].split(**","**)  
 lengteLaureaten = len(lijstLaureaten)  
  
 *#Deze variabele krijgt alle nationaliteiten die zich in een rij bevinden.* nationaliteiten = geefNationaliteitenInEenRij(lengteLaureaten, row[3], lijstLaureaten)  
  
 *#Deze variable krijgt aantal overeenkomende kolomen.* aantalOvereenKomendeArgumenten = geefAantalOvereenkomendeKolomen(argumentsAfterCheck, row, lengteLaureaten, nationaliteiten)  
  
 *#Als de lengte gelijk is aan de lengte van de gevraagde argumenten* **if** (aantalOvereenKomendeArgumenten == len(argumentsAfterCheck)):  
  
 *#Deze rij word toegevoegd aan de lijst* rijen.append(row)  
 **return** rijen  
  
*#Deze functie knipt de nationaliteiten in een rij en zet ze in een string.  
#Dit doet door de waardes binnen de parentheses te knippen.  
#Aan het einde stuurt deze functie de strings met alle nationaliteiten in een rij terug.***def** geefNationaliteitenInEenRij(lengteLaureaten, rij, lijstLaureaten):  
 nationaliteiten = **''** nationaliteitTeller = 0  
 **while** nationaliteitTeller < lengteLaureaten:  
 **if** (rij != **'laureaat'**):  
 nationaliteiten = nationaliteiten + re.search(**'\(([^)]+)'**, lijstLaureaten[nationaliteitTeller]).group(  
 1) + **' '** nationaliteiten = nationaliteiten.upper()  
 nationaliteitTeller += 1  
 **return** nationaliteiten  
  
  
*#Deze functie controlleert hoeveel kolomen in een rij overeenkomen met de zoektermen.  
#Deze wordt bijgehouden door elke keer als er iets overeenkomt de variabele  
#aantalOvereenKomendeArgumenten met een opgehoogd. Aan het eind stuurt deze functie dat variabele terug.***def** geefAantalOvereenkomendeKolomen(argumentsAfterCheck, row, lengteLaureaten, nationaliteiten):  
 teller = 0  
 aantalOvereenKomendeArgumenten = 0  
 **while** (teller < len(argumentsAfterCheck)):  
 **for** kolom **in** row:  
 **if** row.index(kolom) == 3:  
 **if** argumentsAfterCheck[teller] == lengteLaureaten:  
 aantalOvereenKomendeArgumenten += 1  
 **elif** type(argumentsAfterCheck[teller]) != int \  
 **and** argumentsAfterCheck[teller] **in** nationaliteiten:  
 aantalOvereenKomendeArgumenten += 1  
 **elif** row.index(kolom) == 4 **and** type(argumentsAfterCheck[teller]) != int **and** \  
 argumentsAfterCheck[teller] **in** kolom:  
 aantalOvereenKomendeArgumenten += 1  
 **elif** type(argumentsAfterCheck[teller]) != int **and** \  
 argumentsAfterCheck[teller] == kolom:  
 aantalOvereenKomendeArgumenten += 1  
 teller += 1  
 **return** aantalOvereenKomendeArgumenten  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 main()

# Test

*Invoer: prijzen('prijzen.csv', prijs='nobelprijs', jaar=1994)*

*Uitvoer: Nobelprijs voor de Economie (1994): John Harsanyi (VS), John Forbes Nash (VS), Reinhard Selten (Dui)*

*Nobelprijs voor de Geneeskunde (1994): Alfred G. Gilman (VS), Martin Rodbell (VS)*

*Nobelprijs voor de Literatuur (1994): Kenzaburo Oë (Jap)*

*Nobelprijs voor de Natuurkunde (1994): Bertram Neville Brockhouse (Can)*

*Nobelprijs voor de Natuurkunde (1994): Clifford Glenwood Shull (VS)*

*Nobelprijs voor de Scheikunde (1994): George Andrew Olah (VS)*

*Nobelprijs voor de Vrede (1994): Yasser Arafat (Pal), Shimon Peres (Isr), Yitzhak*

*Invoer: prijzen('prijzen.csv', nationaliteit='bel')*

*Uitvoer: Nobelprijs voor de Vrede (1904): Institut de Droit International (Bel)*

*Nobelprijs voor de Vrede (1909): Auguste Beernaert (Bel)*

*Nobelprijs voor de Literatuur (1911): Maurice Maeterlinck (Bel)*

*Nobelprijs voor de Vrede (1913): Henri La Fontaine (Bel)*

*Nobelprijs voor de Geneeskunde (1919): Jules Bordet (Bel)*

*Nobelprijs voor de Geneeskunde (1938): Corneille Jean FranÃ§ois Heymans (Bel)*

*Nobelprijs voor de Vrede (1958): Georges Pire (Bel)*

*Nobelprijs voor de Geneeskunde (1974): Albert Claude (Bel), Christian de Duve (Bel), George E. Palade (Roe)*

*Nobelprijs voor de Scheikunde (1977): Ilya Prigogine (Bel)*

*Nobelprijs voor de Vrede (1999): Artsen zonder Grenzen (Bel)*

*Abelprijs voor de wiskunde (2008): John G. Thompson (VS), Jacques Tits (Bel)*