

AM1090 Inleiding Programmeren
Deeltentamen 3
28 januari 2022, 9:00 – 11:00 uur

Normering: cijfer is (aantal punten + 2) / 2

Opgave 1 **Puntenverdeling: a) 3 punten; b) 3 punten**

Gebruik bij deze opgave het bestand `A1.py` en breid dit bestand uit. Vul je naam en studienummer in, bovenaan in de file.

We willen random dictionaries maken waarin alle letters 'a' t/m 'z' keys zijn. De values zijn hoofdletters. Elke hoofdletter ('A' t/m 'Z') komt precies 1 keer voor als value. Een voorbeeld van een dictionary die voldoet aan de gegeven regels is

```
dct = {'a': 'M', 'b': 'Z', 'c': 'O', 'd': 'B', 'e': 'J', 'f': 'U', 'g': 'T', 'h': 'Q', 'i': 'L', 'j': 'X', 'k': 'R', 'l': 'N', 'm': 'E', 'n': 'P', 'o': 'D', 'p': 'H', 'q': 'F', 'r': 'C', 's': 'S', 't': 'V', 'u': 'Y', 'v': 'W', 'w': 'G', 'x': 'I', 'y': 'A', 'z': 'K'}
```

- a. Implementeer een functie `maak_random_dict()` die een random dictionary teruggeeft die voldoet aan de bovenstaande regels.

Opmerking: Jouw functie mag een andere (pseudo) random dictionary opleveren dan de dictionary `dct` hierboven, mits de dictionary voldoet aan de gestelde regels. De random getallen/letters die je krijgt zijn namelijk afhankelijk van de manier waarop je de random generator gebruikt.

- b. Implementeer een functie `codeer(string, dct)` die `string` converteert naar een nieuwe string waarin elke kleine letter (zeg `let`) is geconverteerd naar de hoofdletter die de value is behorend bij key `let` in de dictionary `dct`. Andere tekens dan letters in `string` blijven in de nieuwe string ongewijzigd. De nieuwe string wordt door de functie teruggegeven.

Voorbeeld: Laat `string='Deze tekst!'` en `dct` de bovenstaande dictionary dan geeft `codeer(string, dct)` de string `'DJKJ VJRSV!'` terug.

Uitvoer van het testprogramma in `A1.py` bij correcte implementaties van de twee functies:

TEST a) `maak_random_dict`

De pseudo random dictionary is:

```
{'a': 'M', 'b': 'Z', 'c': 'O', 'd': 'B', 'e': 'J', 'f': 'U', 'g': 'T', 'h': 'Q', 'i': 'L', 'j': 'X', 'k': 'R', 'l': 'N', 'm': 'E', 'n': 'P', 'o': 'D', 'p': 'H', 'q': 'F', 'r': 'C', 's': 'S', 't': 'V', 'u': 'Y', 'v': 'W', 'w': 'G', 'x': 'I', 'y': 'A', 'z': 'K'}
```

TEST b) `codeer(string, dct)` met de test_dct dictionary

De gecodeerde string is:

```
VVIHGV IVTVOH EZM SVG TVWRXSG  
'SVIRMMVIRMT ZZM SLOOZMW' (1936) EZM SVMWIRP NZIHNZM:  
WVMPVMW ZZM SLOOZMW  
ARV RP YIVVWV IREVRVIM  
GIZZT WLLI LMVRMWRT  
OZZTOZMW TZZM,  
IRQVM LMWVMPYZZI  
RQOV KLKFORVIM  
ZOH SLLTV KOFRNVM  
ZZM WVM VRMWVI HGZZM,  
...
```

De gedecodeerde string is:
eerste regels van het gedicht
'herinnering aan holland' (1936) van hendrik marsman:
denkend aan holland
zie ik breede rivieren
traag door oneindig
laagland gaan,
rijen ondenkbaar
ijle populieren
als hoge pluimen
aan den einder staan,
...

TEST a) en b) met een random dictionary:
De gecodeerde string is:
JJCSVJ CJTJNS WMP QJV TJBLOQV
'QJCLPPJCLPT MMP QDNNMPB' (1936) WMP QJPBCLR EMCSEMP:
BJPRJPB MMP QDNNMPB
KLJ LR ZCJJB J CLWLJCJP
VCM MT BDDC DPJLPBLT
NMMTNMPB TMMP,
CLXJP DPBJPRZMMC
LXNJ HDHYNLJCJP
MNS QDDTJ HNYLEJP
MMP BJP JLPBJC SVMMP,
...

De gedecodeerde string is:
eerste regels van het gedicht
'herinnering aan holland' (1936) van hendrik marsman:
denkend aan holland
zie ik breede rivieren
traag door oneindig
laagland gaan,
rijen ondenkbaar
ijle populieren
als hoge pluimen
aan den einder staan,
...

Opgave 2 **Puntenverdeling: a) 3 punten; b) 1 punten; c) 3 punten; d) 2 punten**

Gebruik bij deze opgave het bestand A2.py en breid dit bestand uit. Vul je naam en studienummer in, bovenaan in de file.

Je gaat het spel **24 game** implementeren. De spelregels zijn:
Het doel van het **24 game** is om met vier getallen **24** te maken. De getallen die op het kaartje staan (bij deze opgave staan ze in een lijst van 4 getallen uit 1 t/m 9) moet je 1 maal gebruiken. Je mag de getallen vermenigvuldigen, delen, optellen of aftrekken van elkaar. De uitkomst van een som mag je weer gebruiken voor een volgende som. Alle operatoren +, -, * en / hebben dezelfde prioriteit.

Voorbeeld 1:
De getallen zijn 7, 1, 5, 9. Er zijn GEEN oplossingen.
Voorbeeld 2:
De getallen zijn 3, 5, 2, 9. Een oplossing is de string '9*3-5+2'
De lijst met alle oplossingen is:
['3*9+2-5', '3*9-5+2', '5-2*9-3', '5*9+3/2', '9*3+2-5', '9*3-5+2',
'9-5*3*2', '9*5+3/2', '9-5*2*3']

- Opmerkingen:
- De implementatie van het spel is object georiënteerd in 2 klassen Game (is geheel gegeven) en Opgave (is gedeeltelijk gegeven).

- In onze versie van het spel maken we steeds gebruik van 4 **verschillende** getallen uit 1 t/m 9.
- Er is al een methode `oplossingen(self)` geïmplementeerd die alle oplossingen berekent en in een lijst van strings teruggeeft.
- Alle operatoren hebben **gelijke prioriteit**.
- Bijvoorbeeld: antwoord string `'5-2*9-3'` betekent $5-2 = 3$; $3*9 = 27$; $27-3 = 24$
- Een tussenresultaat van een berekening mag een niet geheel rationaal getal zijn (een breuk).
- Bijvoorbeeld: string `'9/6*8*2'` is correct.
- Een tussenresultaat van een berekening mag ook een negatief getal zijn.
- Bijvoorbeeld: string `'1-3+5*8'` is correct.

- a. Implementeer de methode `__str__` die de `str` functie van Python overlaadt. Een Opgave object waarop de methode wordt toegepast moet de opgave als string teruggeven.

Voorbeeld: Met de volgende opdrachten:

```
opg2 = Opgave([7,1,5,9])
print(opg2, "\n")
```

moet het volgende worden geprint in de console of Python shell:

Uitkomst is 24

Met getallen 7,1,5,9

En operatoren +,-,*,/

- b. Implementeer een methode `heeftOplossing` die bij toepassing op een object van de klasse `Opgave` bepaalt of de opgave oplossingen heeft en `True` of `False` teruggeeft.

Voorbeelden:

Voor de opgave met getallen attribuut `[7, 1, 5, 9]` geeft de methode `False` terug.

Voor de opgave met getallen attribuut `[1, 3, 4, 7]` geeft de methode `True` terug. Een oplossing is bijvoorbeeld `'3+1*7-4'`.

- c. Implementeer een functie `randomOpgave()` buiten de klasse `Opgave`, die een random opgave teruggeeft, d.w.z. een opgave die (wanneer we de opgave zouden printen) er uit zou zien als:

Uitkomst is 24

Met getallen ?,?,?,?

En operatoren +,-,*,/

Met op de vraagtekens vier willekeurige **verschillende** getallen uit 1 t/m 9.

- d. Implementeer een functie `check(antwoord, opgave)` buiten de klasse `Opgave`, die voor een string antwoord bepaalt of antwoord een oplossing is van de 24 Game opgave.

Uitvoer van het testprogramma in `A2.py` bij correcte implementaties van alle methoden en functies:

TEST a) `__str__` methode:

`opg1` is

Uitkomst is 24

Met getallen 1,3,4,7

En operatoren +,-,*,/

`opg2` is

Uitkomst is 24

Met getallen 7,1,5,9

En operatoren +,-,*,/

#####

Met de al gegeven methode `oplossingen`:

Oplossingen van `opg1`:

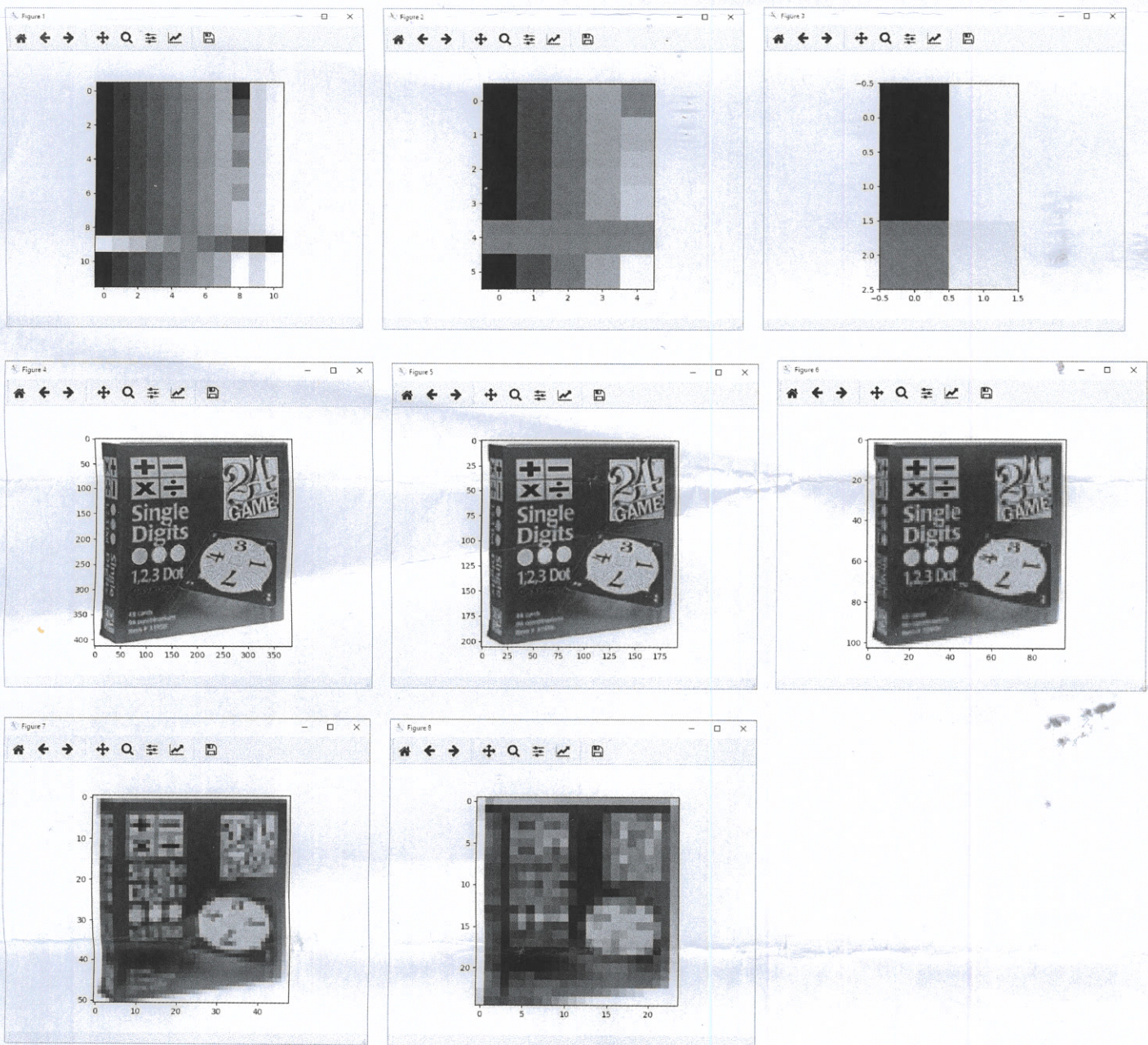
```
['1+3*7-4', '3+1*7-4', '3*7-1+4', '3*7+4-1', '4-1*7+3', '4*7-1-3', '4*7-3-1', '7*3-1+4', '7*3+4-1', '7*4-1-3', '7*4-3-1']
```

Oplossingen van `opg2`:


```
[[0.05 0.25 0.45 0.65 0.5 ]
 [0.05 0.25 0.45 0.65 0.7 ]
 [0.05 0.25 0.45 0.65 0.75]
 [0.05 0.25 0.45 0.65 0.8 ]
 [0.5  0.5  0.5  0.5  0.5 ]
 [0.05 0.25 0.45 0.65 0.95]]
```

```
[[0.15  0.55 ]
 [0.15  0.55 ]
 [0.325 0.525]]
```

Het testprogramma genereert tevens de afbeeldingen:



Inleveren van je werk

Lees het laatste gedeelte van het instructieblad en voer alle handelingen uit om je werk in te leveren. Je werk wordt uitsluitend aan de hand van de .py files met je naam en studienummer beoordeeld. Controleer of er drie .py files met je naam en studienummer op Private (P:) staan.

Einde tentamen.