

Nr. : A105X

Naam :

S-nummer :

Programmeren en dataverwerking II

Opdrachten Sessie 14 (Individuele opdracht)

Deze opdrachten worden individueel gemaakt en ingeleverd op Canvas. Bij het maken van de opdrachten is het toegestaan de documentatie behorende bij het vak en eerder gemaakte opdrachten te gebruiken. Het is NIET toegestaan internetbronnen te raadplegen.

Elke opdracht levert 1 .py file op die ingeleverd wordt in Canvas.

Het totaal aantal te behalen punten is 100. Je moet minimaal 60 punten scoren voor een VOLDAAN.

Opdracht 1 (5p). (lever in als: opdracht1.py)

Definieer een Numpy array met de waarden van $\sin(x)$ voor x in het interval $[-10, 10]$ en stapgrootte 0.01

Opdracht 2 (10p). (lever in als: opdracht2.py)

gegeven is de volgende Numpy array:

```
A : [[ 1  2  3  4  5],
      [ 2  6  9 12 13],
      [-23 45 12 35 36],
      [ 12 -34 56  1  2],
      [ 2  -3  5  2 23]]
```

Maak een script die door middel van slicing van A de volgende Numpy arrays definieert:

```
B: [ 2  6  9 12 13]
```

```
C: [[12 35 36]
     [56  1  2]
     [ 5  2 23]]
```

```
D: [[ 1  3  5]
     [-23 12 36]
     [ 2  5 23]]
```


Opdracht 3 (10p). (lever in als: opdracht3.py)

- Maak een script die door middel van list comprehensions een list k definieert bestaande uit de kwadraten i^2 van alle gehele **even** getallen $2 \leq i \leq 100$.
- Print alle waarden van de list naar de console mbv de iterator `range()`. Voorbeeld:
kwadraat van 2 is 4
kwadraat van 4 is 16
etc
- Idem als (b) maar nu met de `enumerate()` iterator.

Opdracht 4 (15p). (lever in als: opdracht4.py)

- a. Schrijf een functie `parabool(x, a, b, c)` die de waarden van een polynoom berekent volgens: $y = ax^2 + bx + c$ gegeven de waarde van x , a , b en c . De variabelen x , a , b en c zijn de argumenten van de functie, de waarde van y wordt teruggegeven. Schrijf de functie zodanig dat x en a verplichte argumenten zijn en dat b en c optioneel zijn en als default waarde 0 hebben. Bijvoorbeeld:

```
print(parabool(1,2,c=3))
```

geeft als resultaat:  5

- b. Maak een lijn plot van deze functie voor $a=1$, $b=2$ en $c=3$ op het interval $x=[-10,10]$. Kies zelf een geschikt aantal punten.
- c. Plot in dezelfde figuur de functie mbv rode markers (bijvoorbeeld een `*`) voor de gehele getallen van x .
- d. Voeg geschikte labels toe.
- e. Voeg geschikte tekst bij de assen toe en geef de figuur een titel.

Opdracht 5 (20p). (lever in als: opdracht5.py)

Schrijf een functie `passwordcheck(password)` die controleert of de string `password` voldoet aan de volgende eisen:

- Ten minste 1 letter [a..z]
- Ten minste 1 cijfer [0..9]
- Ten minste 1 teken [#@&]
- Ten minste 8 tekens lang

De functie geeft `True` als aan de eisen is voldaan anders `False`.

Bijvoorbeeld: `passwordcheck('ght23&ut')` geeft `True`; `passwordcheck('ert1trew')` geeft `False`.

Opdracht 6 (20p). (lever in als: `opdracht6.py`)

We willen het getal π benaderen met de volgende reeks:

$$\sum_{n=0}^N 4 \frac{-1^n}{2n+1}$$

- Schrijf een functie `pi_benadering(N)` die bovenstaande som bepaalt gegeven de waarde voor N .
- Welke waarde moet je kiezen voor N als je π op 3 decimalen nauwkeurig wilt bepalen? Schrijf hiervoor een script die gebruikmaakt van de functie die je hebt geschreven bij (a).

Opdracht 7 (20p). (lever in als: `opdracht7.py`)

We hebben een Numpy array met Bitcoin koersen in Euro's als functie van de tijd (oplopend).
Bijvoorbeeld:

```
btc_euro = np.array([102, 95, 110, 75, 43, 98, 101, 88])
```

Om winst te maken moet je eerst bitcoin kopen tegen bij voorkeur een lage koers en later weer verkopen als de koers hoger is. In bovenstaande voorbeeld kun je bijvoorbeeld kopen voor 102 en verkopen bij een koers van 110. Je hebt dan een winst van 8. Er is gegeven dat je maximaal 1 keer mag kopen en 1 keer mag verkopen. Gevraagd wordt de maximale te behalen winst te bepalen. In het voorbeeld is dat $101 - 43 = 58$.

Schrijf een functie `optimize_profit(btc_euro)` die de maximale winst bepaalt.

EINDE OPDRACHTEN