Uitwerking opdracht

Opdracht :	*luchthavens*
Weeknummer :	*6*
Studentnummer :	*s1060679*
Naam student :	*Dion van den Berg*
Specialisatie :	*MEDT*
Pogingnummer :	*1*

1. Vraagstelling

Zet een set luchthavens om naar de afstand ertussen

2. Specificatie

Invoer

De gebruiker voegt 2 luchthavens in

Uitvoer

De gebruiker krijgt terug hoever ze uit elkaar liggen

Verband tussen in- en uitvoer

De uitvoer wordt gemaakt door de ingevoerde set aan luchthavens

Beperkingen

Foutieve invoer kan ingevoerd worden.

Voorbeelden (testscenario's)

Test 1

Invoer:

```
luchthavens['ADK']
```

Uitvoer:

```
(51.88, 176.65, 'Adak', 'AK')
```

3. Ontwerp

Hoe maak ik het bestand open?
Door het te openen met open() en daarna uit te lezen met .read()

Hoe ga ik alles omzetten?
Door alles in het bestand te splitsen op de spaties.

Hoe laat ik alles uiteindelijk zien ?
in een print statement de uitvoer te plakken

4. Pseudocode

Functie leesLuchthavens (bestandLocatie)
Open het bestand
Loop door het gehele bestand heen en vervang alle spaties
Voeg alles toe aan een lijst

Functie afstand (locatie1, locatie2, lijst)
Bereken de afstand tussen de 2 locaties
Gebruik de opgegeven formulie van de vraag website

Functie tussenlanding (locatie1, locatie2, lijst)
Kijk wat de snelste route is met 1 tussenlanding
Loop door alle luchthavens heen en bereken de kortste afstand.

5. Code

```
import math


def leesLuchthavens(file):
    # open het bestand van de opgegeven locatie.
    bestand = open(file)
    lijst = {}
    # loop door de regels in het bestand heen en strip alle losse regels.
    for l in bestand:
        sep = l.split("\t")
        if sep[0] == "Airport":
            continue
        lijst[sep[0].replace("[", '').replace("]", '')] = (float(sep[1]),
float(sep[2]), sep[3], sep[4].rstrip())
    return lijst


def afstand(c1, c2, lijst):
    # krijg de te vergelijken lijst mee en de 2 locaties.
    # bereken het begin en eindpunt
    b1 = math.radians(lijst[c1][0])
    b2 = math.radians(lijst[c2][0])
    l1 = math.radians(lijst[c1][1])
    l2 = math.radians(lijst[c2][1])
    r = 6372.795  # de opgegeven straal
    # voor de opgegeven berekening uit met de berekende waarde
    y = math.sqrt(math.pow(math.cos(b2) * math.sin(l1 - l2), 2) + math.pow(
        math.cos(b1) * math.sin(b2) - math.sin(b1) * math.cos(b2) *
math.cos(l1 - l2), 2))
    x = math.sin(b1) * math.sin(b2) + math.cos(b1) * math.cos(b2) *
math.cos(l1 - l2)
    return r * math.atan2(y, x)


def tussenlanding(c1, c2, l, r=4000):
    # bereken het optimale vluchtschame met 1 tussenlanding.
    # daarbij moet wel de afstand onder de 4000km liggen
    if afstand(c1, c2, l) <= 4000:
        return None
    woordenboekArray = {}
    for key in l:
        if afstand(c1, key, l) <= r and afstand(c2, key, l) <= r:
            woordenboekArray[afstand(c1, key, l) + afstand(c2, key, l)] =
key
    if len(woordenboekArray) > 0:
        return woordenboekArray[min(woordenboekArray.keys())]
    return None

# check het programma met de opgegeven test scenario's

luchthavens = leesLuchthavens('luchthavens.txt')
print(luchthavens)
print(luchthavens['ADK'])
print(luchthavens['DCA'])
print(luchthavens['4OM'])
print(afstand('P60', 'MSN', luchthavens))
print(afstand('ADK', 'DCA', luchthavens))
print(tussenlanding('ADK', 'DCA', luchthavens, 4000))
```

6. Test

Test 1

```
luchthavens = leesLuchthavens('luchthavens.txt')
print(luchthavens)
print(luchthavens['ADK'])
print(luchthavens['DCA'])
print(luchthavens['4OM'])
print(afstand('P60', 'MSN', luchthavens))
print(afstand('ADK', 'DCA', luchthavens))
print(tussenlanding('ADK', 'DCA', luchthavens, 4000))
```

```
{'AGN': (57.83, 134.97, 'Angoon', 'AK'), ...}
(51.88, 176.65, 'Adak', 'AK')
(38.85, 77.04, 'Washington/Natl', 'DC')
(48.42, 119.53, 'Omak', 'WA')
1694.545549951611
7295.503556775978
4OM
```