

Uitwerking opdracht

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

1. Vraagstelling

De gebruiker wil weten wat de beste tijd is om wakker te worden om zijn slaap cycli zo efficiënt mogelijk af te ronden.

2. Specificatie

Invoer

De gebruiker voert zijn huidige tijdstip in en het tijdstip dat hij wakker wil worden.

Uitvoer

De gebruiker krijg te zien op welke tijdstip hij zijn wekker het beste kan zetten. Dit zijn veelvoud van 90 minuten.

Verband tussen in- en uitvoer

De efficiëntste tijd wordt berekend met het tijdstip dat de gebruiker invoert

Beperkingen

De gebruiker kan rare tijdstippen invoeren die de datetime functie niet oppakt.

Voorbeelden (testscenario's)

Test 1

invoer:

Voer het huidige uur in: 11

voer de huidige minuut in: 24

Voer wakker wordt uur in: 18

Voer wakker wordt minuut in: 47

Uitvoer:

18:54

Test 2

Invoer:

Voer het huidige uur in:17

voer de huidige minuut in: 56

Voer wakker wordt uur in: 16

Voer wakker wordt minuut in: 40

Uitvoer:

17:56

3. Ontwerp

Hoe vang ik de invoer op van de gebruiker ?

Door de python functie input

Hoe ga ik om met tijdstippen die eerder zijn dan het eerste tijdstip ?

Dit zou betekenen dat hij een dag verder is. Dus moet de dag met 1 verhoogd worden.

Hoe laat ik alles uiteindelijk zien ?

in 1 regel met daarin de perfecte tijd om op te staan.

4. Pseudocode

```
Input uur
Input minuut
Input eruit uur
Input eruit minuut

Erin = combineren uur + minuut als H:M
Eruit = combineren eruit uur + minuut als H:M

While true
    If eruit < erin
        Print tijdstilvan wakker worden.

        Erin += tel er 90 minuten bij op.
```

5. Code

```
from datetime import datetime, timedelta

# gebruiker voer uur en minuut in waarin hij naar bed gaat.
invoerUur = input("Voer het huidige uur in:")
invoerMinuut = input("voer de huidige minuut in: ")

# gebruiker voert uur en minuut in waarin hij uit zijn bed wil.
invoerUurEruit = input("Voer wakker wordt uur in: ")
invoerMinuutEruit = input("Voer wakker wordt minuut in: ")

# zet de invoer om in 1 string die de datetime functie kan gebruiken.
erIn = datetime.strptime(invoerUur + ":" + invoerMinuut, '%H:%M')

# zet de invoer om in 1 string die de datetime functie kan gebruiken en
voeg toe dat hij overweg kan met extra dagen na 24 uur.
erUit = (datetime.strptime(invoerUurEruit + ":" + invoerMinuutEruit,
'%H:%M') + timedelta(days=1))

# vergelijk de 2 strings van tijden met elkaar.
# ga je later uit bed dan dat je erin gaat. Dus de volgende dag. Dan
telt hij daar iig 90 minuten bij op.
# dit doet hij tot je over je ingevoerde tijd heengaat dat je eruit
wilde gaan.
while True:
    if erUit < erIn:
        print("het beste tijdstip dat je wakker kan worden is: " +
datetime.strftime(erIn, '%H:%M'))
        break

    erIn += timedelta(minutes=90)
```

6. Test

Test 1

Voer het huidige uur in: 22

voer de huidige minuut in: 34

Voer wakker wordt uur in: 6

Voer wakker wordt minuut in: 07

het beste tijdstip dat je wakker kan worden is: 07:34

Test 2

Voer het huidige uur in: 6

voer de huidige minuut in: 07

Voer wakker wordt uur in: 22

Voer wakker wordt minuut in: 54

het beste tijdstip dat je wakker kan worden is: 00:07