Danmarks Tekniske Universitet

Kursusnavn	Introduktion til programmering og databehandling
Kursusnummer	02631, 02632, 02633, 02634, 02692
Hjælpemidler	Alle hjælpemidler er tilladt
Varighed	2 timer
Vægtning	Opgaverne vægtes ens

Indhold

Aflevering

Du skal aflevere dine løsninger elektronisk:

1. Du kan uploade dine løsninger individuelt på CodeJudge

   https://dtu.codejudge.net/prog-aug20/assignments/

   under *Exam*. Når du afleverer en løsning på CodeJudge bliver det test-eksempel som fremgår i opgavebeskrivelsen kørt på din løsning. Hvis din løsning består denne ene test, vil den fremgå som *Submitted*. Det betyder at din løsning består på dette ene test-eksempel. Du kan uploade til CodeJudge så mange gange som du ønsker under eksamen.

2. Du skal uploade alle dine løsninger på DTU's Onlineeksamen site. Hver assignment skal uploades som en separat .py fil med samme navn som funktionen i opgaven:

   (a) `double_sort.py`
   (b) `compile.py`
   (c) `pack_clips.py`
   (d) `navigate.py`
   (e) `nonpalindromic_prime.py`

   Filerne skal afleveres separat (*ikke* som en zip-fil) og skal have præcis disse filnavne.

Efter eksamen vil dine løsninger fra DTU Inside blive automatisk evalueret på CodeJudge på en række forskellige tests for at undersøge om de generelt fungerer korrekt. Bedømmelsen af din aflevering foretages udelukkende på baggrund af hvor mange af de automatiserede test der bestås.

- Du skal sikre dig at din kode følger specifikationerne nøje.
- Hver løsning må ikke indeholde nogen yderligere kode udover den specificerede funktion, dog kan `import`-sætninger inkluderes.
- Husk at du kan tjekke om dine løsninger følger specifikationerne ved at uploade dem til CodeJudge.
- Bemærk at alle vektorer og matricer anvendt som input eller output skal være numpy arrays.

Assignment A Double sort

Givet en matrix ønsker vi at sortere elementer i hver kolonne i stigende orden baseret på størrelsen (den absolutte værdi) og derefter sortere kolonnerne i stigende orden baseret på værdi (med fortegn) af elementet i den første række. Hvis elementerne i den første række er ens, så skal kolonnerne med de ens elementer sorteres efter rækkefølgen i den oprindelige matrix.

Problemdefinition

Skriv en funktion kaldet `double_sort` som tager som input $\mathbf{X}$ der er en matrix. Funktionen skal returnere en sorteret matrix $\mathbf{Y}$.

Løsningsskabelon

```
def double_sort(X):
  #insert your code
  return Y
```

Input

`X`	Matrix med en eller flere kolonne og en eller flere rækker.

Output

`Y`	Matrix med en eller flere kolonner og en eller flere rækker.

Eksempel

Betragt følgende matrix

$$\mathbf{X} = \begin{bmatrix} 7 & -1 & 0 & 5 \\ 2 & 5.2 & 4 & 2 \\ 3 & -2 & 1 & 4 \end{bmatrix}$$

Først sorteres hver række individuelt baseret på størrelse

$$\begin{bmatrix} 2 & -1 & 0 & 2 \\ 3 & -2 & 1 & 4 \\ 7 & 5.2 & 4 & 5 \end{bmatrix}$$

Så sorteres kolonnerne baseret på elementerne i den første række

$$\mathbf{Y} = \begin{bmatrix} -1 & 0 & 2 & 2 \\ -2 & 1 & 3 & 4 \\ 5.2 & 4 & 7 & 5 \end{bmatrix}.$$

Denne matrix returneres.

A

Assignment B Compile

Vi ønsker at konstruere en kompiler der tager en streng som repræsenterer et lille program som input og kompilerer og eksekverer det. Strengen, der repræsenterer et program, indeholder positive heltal og fire aritmetriske operationer: "A", "S", "M" og "D" for henholdsvis addering, subtraktion, multiplicering og division.

Programmet kan være givet som et enkelt heltal, flere heltal med en af de fire operationer imellem. Heltal og operationer er adskilt af et mellemrumstegn. Operator-forrangen betyder at operationerne skal udføres med den operation der er længst til venstre først, så at for eksempel "2 A 3 M 4" betyder $(2+3)\times 4$.

Det kan antages at kun korrekte programmer er givet som input.

Problemdefinition

Skriv en funktion kaldet `compile` som tager et input: Programmet som en streng. Funktionen skal returnere det numeriske resultat med en evaluering af programmet.

Løsningsskabelon

```
def compile(program):
  # insert your code
  return result
```

Input

`program`	Streng med programmet

Output

`result`	Decimal- eller heltal med resultatet af beregningen.

Eksempel

Betragt programmet "2 A 3 M 4 D 8". Dette skal fortolkes som $((2+3)\times 4)/8$ og dette skal resultere i $(5\times 4)/8 = 20/8 = 2{,}5$. Så decimaltallet 2,5 returneres.

B

Assignment C Pack clips

Vi ønsker at passe audioklip af en vis varighed ind i et specificeret tidsvindue med en total varighed. Hver varighed er givet i et heltal minutter og sekunder, og den totale varighed af tidsvinduet er også givet som minutter og sekunder.

Problemdefinition

Skrev en funktion kaldet `pack_clips` der tager som det første input en $(N \times 2)$-matrix hvor hver række er en varighed for et audioklip og den første kolonne specificerer antallet af minutter for klippet og den anden kolonne specificerer antallet af sekunder for klippet. Som andet argument tager funktionen varigheden af tidsvinduet som en 2-element vector med det første element der specificerer minutter og det andet element sekunder. Funktionen skal returnere det maximale antal af klip hvor summen af varighederne er mindre end eller lig med den totale varighed af tidsvinduet. Klippene skal vælges i rækkefølge startende med den første række. Hvis ingen kan passes ind, så skal funktionen returnere nul. Hvis alle kan passe ind, så skal funktionen returnere tallet N

Løsningsskabelon

```
def pack_clips(durations, total):
  # insert your code
  return clips
```

Input	
`durations`	$(N \times 2)$-matrix med varigheder. $N > 0$
`total`	2-element vektor med totale varighed af tidsvinduet.

Output	
`clips`	Heltal skalar med antallet af audioklip der passer i tidsvinduet.

Eksempel

Antag at den totale varighed er 10:20 og at 6 audioklips varighed er

```
3:10 4:20 0:50 1:50 10:10 1:00
```

Varighederne er repræsenteret i en matrix

$$\begin{bmatrix} 3 & 10 \\ 4 & 20 \\ 0 & 50 \\ 1 & 50 \\ 10 & 10 \\ 1 & 0 \end{bmatrix} \tag{1}$$

mens den totale varighed er repræsenteret i en vektor $[10, 20]$.

1. Første klip har varigheden 3:10 og passer i tidsvinduet.
2. De første to klip har varigheden 7:30 og passer i tidsvinduet.
3. De første tre klip har varigheden 8:20 og passer i tidsvinduet.
4. De første fire klip har varigheden 10:10 og passer i tidsvinduet.
5. De første fem flip har varigheden 20:10 og passer ikke i tidsvinduet.

Derfor skal funktionen returnere 4.

C

Assignment D Navigate

Vi ønske at navigere på DTU's campus hvor vi har en mål-bygning repræsenteret med et 3-cifret tal og en start-bygning også repræsenteret med et 3-cifret tal, hver i en af fire kvadranter. Bygninger i 100-rækken (100-199) er i nordøst-kvadranten, 200'ernes bygninger i nordvest, 300'ernes bygninger i sydvest og 400'ernes bygninger i sydøst.

Vi ønsker en retningsbeskrivelse med start-bygningens nummer, retningen og mål-bygningens nummer. Specifikationen af retning er S, N, W, E for syd, nord, vest og øst, eller SW, NW, SE, NE for sydvest, nordvest, sydøst og nordøst.

Hvis vores start-bygning er 413 og målet er 202, ønsker vi en streng "413NW202", der indikerer retning fra bygning 412 (sydøst) til bygning 2020 (nordvest). En start-mål kombination med 414 og 101 skal være "414N101".

Hvis starten og målet er i samme kvadrant, så skal retningen være "H", f.eks. 303 og 306 skal resultere i retningsbeskrivelsen "303H306".

Problemdefinition

Skriv en funktion kaldet `navigate` der tager to heltal input repræsenterende bygningsnumre, the første starten, det andet målet. Den skal returnere en streng med en sammensætning af start-bygningen som en streng, retningen og mål-bygningen som en streng.

Løsningsskabelon

```
def navigate(origin, target):
    #insert your code
    return direction
```

Input

`origin`	3-cifret heltal der specificerer start-bygningen.
`target`	3-cifret helt der specificerer mål-bygningen.

Output

`direction`	Streng med start-bygningen, retningen (S, N, E, W, NW, NE, SW, SE eller H) og mål-bygningen sammensat.

Eksempel

Betragt tilfældet med start-bygningen som 413 og målet som 202, så skal strengen "413NW202" returneres.

D

Assignment E Nonpalindromic prime

Et palindromisk primtal er et primtal der også er et palindromtal. Et palindromtal er et tal der er det sammen når tallets cifre læses baglæns. Eksempler på palindromisk primtal er 2, 3, 11 og 101. Et ikke-palindrom-primttal er et primtal der ikke er et palindrom. Eksempler på ikke-palindrom-primtal er 13, 17, 31 og 47. Vi ønsker at have en funktion der returnerer det n'te ikke-palindrom-primtal.

Problemdefinition

Skriv en funktion kaldet `nonpalindromic_prime` som tager et heltal der specificerer indekset i ikke-palindrom-primtal rækken og returnerer primtallet.

Løsningsskabelon

```
def nonpalindromic_prime(n):
  # insert your code
  return prime
```

Input	
`n`	Heltal der specificerer hvilket n'te ikke-palindrom-primtal der skal returneres.

Output	
`prime`	Heltal som er det n'te ikke-palindrom-primtal.

Eksempel

For $n = 7$ lister vi de første primtal

$$2\ 3\ 5\ 7\ 11\ 13\ 17\ 19\ 23\ 29\ 31\ 37\ 41\ 43\ \ldots$$

Når palindromerne er fjernet får vi listen af ikke-palindrom-primtal

$$13\ 17\ 19\ 23\ 29\ 31\ 37\ 41\ 43\ \ldots$$

Det 7. ikke-palindom-primtal er $\underline{37}$ og det returneres.

E