Danmarks Tekniske Universitet

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

Indhold

Aflevering

Du skal aflevere to identiske kopier af dine løsninger:

1. Du skal uploade alle dine løsninger på CampusNet.
2. Du skal uploade dine løsninger individuelt på CodeJudge under *Exam*.

Bemærk at 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 kun at din løsning består på dette ene test-eksempel. Efter eksamen vil dine løsninger blive evalueret på en række forskellige tests for at undersøge om de generelt fungerer korrekt.

Assignment A Age groups

Den følgende tabel oplister navnene på forskellige aldersgrupper:

Age group	Age, x
`Infant`	$x < 1$
`Toddler`	$1 \leq x < 3$
`Child`	$3 \leq x < 13$
`Teenager`	$13 \leq x < 20$
`Adult`	$20 \leq x$

Problemdefinition

Skriv en funktion med navnet `computeAgeGroup` som tager en alder som et numerisk input og returnerer aldersgruppen som en streng (skrevet præcis som i tabellen ovenfor.)

Løsningsskabelon

```
def computeAgeGroup(age):
  #insert your code
  return ageGroup
```

Input	
`age`	Alder i år (decimaltal).

Output	
`ageGroup`	Aldersgruppe (streng).

Eksempel

Hvis alderen er $x = 2.9$ år skal strengen `Toddler` returneres. Hvis alderen er $x = 3$ år skal strengen `Child` returneres.

A

Assignment B Shirt sizes

Den følgende tabel oplister størrelserne på herre-trøjer som passer forskellige bryst- og taljemål.

Size	Chest	Waist
`Small`	38–42	30–35
`Medium`	40–44	32–37
`Large`	42–46	34–39
`X-Large`	44–48	36–41
`XX-Large`	46–50	38–43
`Not available`	–	–

Problemdefinition

Skriv en funktion med navnet `computeShirtSize` som returnerer trøje-størrelsen (Size) som en streng (skrevet præcis som i tabellen ovenfor) sådan at de givne brystmål (Chest) og taljemål (Waist) begge er inden for de angivne intervaller. Bemærk at intervallerne overlapper: Hvis det givne bryst- og taljemål passer med mere end en trøje-størrelse, skal funktionen returnere den mindste af de størrelser der passer. Hvis de givne bryst- og taljemål ikke passer med nogen af de givne trøje-størrelser, skal strengen `Not available` returneres.

Løsningsskabelon

```
def computeShirtSize(chest, waist):
  #insert your code
  return shirtSize
```

Input	
`chest`	Brystmål (decimaltal).
`waist`	Taljemål (decimaltal).

Output	
`shirtSize`	Trøje-størrelse (streng).

Eksempel

Forestil dig at brystmålet er 43.5 og taljemålet er 34.2. Brystmålet er inden for intervallet for både Medium og Large, og taljemålet er inden for intervallet for Small, Medium og Large. Da begge mål er inden for intervallerne for Medium og Large, vælges den mindste af disse to størrelser og strengen `Medium` skal returneres.

B

Assignment C Air conditioning system

Et avanceret computerstyret air conditioning system kan være i de forskellige tilstande som ses i tabellen nedenfor.

State code	Beskrivelse
0	Off
1	Auto
2	Manual
3	Disabled
4	Fault

Hver gang systemets tilstand ændres, logger systemet den nye tilstandskode (state code) samt tidspunktet for tilstandsændringen som et tidsstempel (time stamp) som måler antallet af sekunder siden loggens begyndelse. Når systemet resættes, logger det altid et time stamp 0 og state code 0 (Off).

Problemdefinition

Skriv en funktion med navnet `acState` som returnerer antallet af sekunder som systemet har været i hver af de fem tilstande, målt fra den først loggede tilstand frem til den senest loggede tilstandsændring. Input til funktionen er en vektor med N tilstandskoder og en vektor af samme længde med de tilsvarende tidsstempler. Funktionen skal fungere for vektorer af vilkårlig længde N.

Løsningsskabelon

```
def acState(state, timeStamp):
  #insert your code
  return stateTime
```

Input

`state`	Tilstandskode (vektor med længde N).
`timeStamp`	Tidsstempel (vektor med længde N).

Output

`stateTime`	Antal sekunder systemet har været i hver af de fem tilstande (vektor med længde 5).

Eksempel

Forestil dig den følgende tilstandssekvens og time stamps.

State code	0	1	2	3	2	3	1
Time stamp	0	486	849	1250	2340	3560	7045

Systemet har været

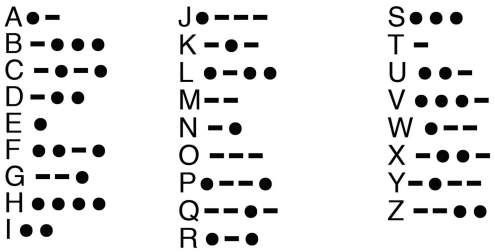
- i tilstand 0 i 486 sekunder,
- i tilstand 1 i $849 - 486 = 363$ sekunder,
- i tilstand 2 i $1250 - 849 + 3560 - 2340 = 1621$ sekunder,
- i tilstand 3 i $2340 - 1250 + 7045 - 3560 = 4575$ sekunder og,
- i tilstand 4 i 0 sekunder.

Funktionen skal således returnere vektoren [486, 363, 1621, 4575, 0].

C

Assignment D Morse code

Morse kode er en metode til at transmittere beskeder ved hjælpe af en serie af korte og lange signaler kaldet *dots* (prikker) og *dashes* (streger). Morsealfabetet er vist nedenfor:



I denne opgave repræsenterer vi en Morse kode besked som en streng, hvor et punktum repræsenterer en dot og en bindestreg repræsenterer en dash. Hvert bogstav er adskilt af et mellemrum og hvert ord er adskilt af to mellemrum.

Problemdefinition

Skriv en funktion som konverterer en streng skrevet i Morse kode til en streng skrevet i det almindelige alfabet med bogstaverne A–Z (og mellemrum mellem ord). Den returnerede tekst skal være skrevet med store bogstaver.

Løsningsskabelon

```
def morseToText(morseCode):
  #insert your code
  return text
```

Input	
`morseCode`	Besked skrevet med Morse-alfabetet (streng kun indeholdende punktum, bindestreg og mellemrums-tegn).

Output	
`text`	Besked skrevet i det normale alfabet (streng kun indeholdende bogstaverne A-Z samt mellemrums-tegn).

Eksempel

Forestil dig den følgende input-streng:

```
-- --- .-. ... .  -.-. --- -.. .
```

For klart at vise mellemrums-tegnene i strengen kan vi markere dem visuelt med et ␣ symbol:

```
--␣---␣.-.␣...␣.␣␣-.-.␣---␣-..␣.
```

For denne input-streng skal outputtet være strengen

```
MORSE CODE
```

D

Assignment E The D'Hondt method

D'Hondts metode er en algoritme som bruges til at fordele mandater til partier efter et valg, baseret på antallet af stemmer. Generelt set skal antallet af mandater være proportionelt med antallet af stemmer; men at opnå præcis proportionalitet kan dog ofte ikke lade sig gøre. I D'Hondts metode tildeles mandaterne til partierne et efter et til det parti der har den højeste kvotient, q, givet ved

$$q = \frac{V}{s+1}, \tag{1}$$

hvor V er antallet af stemmer partiet har fået og s er antallet af mandater tildelt partiet indtil nu. Hvis kvotienterne for to partier er ens, tildeles mandatet til det parti som har opnået det højeste stemmetal, og vi antager at ingen partier har identiske stemmetal.

Problemdefinition

Skriv en funktion som tager en vektor med antallet af stemmer for N partier som input samt antallet af mandater (seats) og returnerer mandatfordelingen beregnet med D'Hondts metode som beskrevet ovenfor.

Løsningsskabelon

```
def dhondt(votes, seats):
  #insert your code
  return seatsAllocated
```

Input	
`votes`	Antal stemmer for hvert af N partier (vektor med længde N).
`seats`	Antal mandater der skal fordeles (positivt heltal).

Output	
`seatsAllocated`	Antal mandater tildelt hvert af de N partier (vektor med længde N).

Eksempel

Forestil dig en situation hvor fem partier har opnået de følgende stemmetal, og der er syv mandater som skal tildeles. Tabellen viser kvotienterne for hvert af partierne efterhånden som mandaterne tildeles et efter et. Den højeste kvotient, som er markeret med fed skrift, giver et mandat til det respektive parti.

	Party A	Party B	Party C	Party D	Party E
Votes	340 000	280 000	160 000	60 000	15 000
Seat 1	**340 000**	280 000	160 000	60 000	15 000
Seat 2	170 000	**280 000**	160 000	60 000	15 000
Seat 3	**170 000**	140 000	160 000	60 000	15 000
Seat 4	113 333	140 000	**160 000**	60 000	15 000
Seat 5	113 333	**140 000**	80 000	60 000	15 000
Seat 6	**113 333**	93 333	80 000	60 000	15 000
Seat 7	85 000	**93 333**	80 000	60 000	15 000
Seats	3	3	1	0	0

Til at begynde med er kvotienten for hvert parti identisk med antallet af stemmer. Det første mandat tildeles Party A, og kvotientet for Party A opdateres til $\frac{340\,000}{2} = 170\,000$. Det andet mandat tildeles Party B som nu har den højeste kvotient, og kvotienten for Party B opdateres til $\frac{280\,000}{2} = 140\,000$. Det tredje mandat tildeles Party A, som nu igen har den højeste kvotient, og kvotienten for Party A opdateres til $\frac{340\,000}{3} = 113\,333$. Tildelingsprocessen fortsætter indtil alle mandater er fordelt. Outputtet fra funktionen i dette eksempel skal være en vektor med antallet af mandater for hvert parti,

$$[3,\ 3,\ 1,\ 0,\ 0]$$

E