

INFO132 Laboppgaver 5

(Noen av oppgaver kan være basert på AI-generert tekst, og kvalitetssikret av en lærer)

Enkle oppgaver

1. Lag en funksjon `siste(sekvens)` som returnerer det siste elementet i en sekvens (som en tekststreng, liste eller intervall)

```
>>> siste('Python')
'n'
>>> siste([1,2,3,4,5])
5
>>> siste(range(90,100))
99
```

2. Lag en funksjon `skrivSekvens(sekvens)` som skriver ut elementene i en sekvens på en linje, adskilt av mellomrom. Eksempel:

```
>>> skrivSekvens('Python')
P y t h o n
>>> skrivSekvens([1,2,3,4,5])
1 2 3 4 5
>>> skrivSekvens(range(90,100))
90 91 92 93 94 95 96 97 98 99
```

3. Lag en funksjon som gitt en sekvens og et heltall N , skriver ut de N første elementene i sekvensen

```
>>> skrivFørste(['Per', 'Liv', 'Kari', 'Ola', 'Jon', 'Eva'],3)
Per
Liv
Kari
```

4. Lag en funksjon som skriver ut annenhver verdi i en sekvens

```
>>> annenhver(['Per', 'Liv', 'Kari', 'Ola', 'Jon', 'Eva'])
Per Kari Jon
```

5. Lag en funksjon `histogram` (*liste*) som gitt en liste av tall tegner liggende søyler av `*`-tegn tilsvarende hvert tall i listen

```
>>>histogram([3,5,7,2,9,8,1,5,0,7])
***
*****
*****
**
*****
*****
*
*****

*****
```

6.

a) Lag en funksjon `tellBokstaver` som tar en tekststreng og en bokstav som input og returnerer antall ganger bokstaven forekommer i strengen.

```
>>>tellBokstaver('programmering', 'm')
2
```

b) Lag en funksjon `tellBokstaverListe` som tar en liste med tekststrenger og en bokstav som input og returnerer antall ganger bokstaven forekommer totalt i listen. Bruk funksjonen fra punkt a)

```
>>>tellBokstaverListe(['matematikk','programmering'],'m')
4
```

7. Lag en funksjon `konkatener` (*lister*) som gitt en liste, der elementene selv er lister, slår de sammen til 1 liste

```
>>>konkatener([[1,2,3],['Bergen','Stavanger'],[9,8,7,6]])
[1, 2, 3, 'Bergen', 'Stavanger', 9, 8, 7, 6]
```

8. Skriv en funksjon `delbar_3` som tar et heltall N som argument. Funksjonen skal skrive ut alle tall fra og med 1 til og med N på hver sin linje. Hvis tallet er delelig med 3 skal det legges til "er delbar på 3". (Husk at tallet N er delbart på 3 dersom $N\%3==0$)

```
>>>delbar_3(7)
1
2
3 er delbar på 3
4
5
6 er delbar på 3
7
```

9. Lag en funksjon som gitt en liste og et heltall N returnerer en liste som inneholder de N første elementene.

```
>>>førsteElementer(['Per', 'Liv', 'Kari', 'Ola', 'Jon', 'Eva'],3)
['Per', 'Liv', 'Kari']
```

Middels vanskelige oppgaver

10. En ekvipasje består av en hest og en rytter. Tenk deg startlister for ridekonkurranse der de to første elementene utgjør den første ekvipasjen, de to neste er den 2. ekvipasjen og så videre, dvs lister på formen

$[rytter_1, hest_1, rytter_2, hest_2, \dots, rytter_N, hest_N]$

Lag funksjonene `hester(startliste)` og `ryttere(startliste)` som returnerer listene av hhv hester og ryttere fra en startliste

```
>>>startliste
['Kari','Blakken','Liv','Dølaprinsen','Eli','Sofus','Gro','Blåne']
>>>hester(startliste)
['Blakken', 'Dølaprinsen', 'Sofus', 'Blåne']
>>>ryttere(startliste)
['Kari', 'Liv', 'Eli', 'Gro']
```

11. Lag en funksjon `pyramide(høyde)` som gitt et tall tegner en pyramidefigur av *-tegn med gitt høyde (*høyde*= antall *-er i midten av figuren)

```
>>>pyramide(4)
*
**
***
****
***
**
*

```

Hint: bruk `histogram`-funksjonen fra forrige oppgave og husk at `range`-typer kan generere synkende tall vha negativt steg

12. Lag et program som skriver ut renteutviklingen av et beløp, år for år, inntil det samlede beløpet overstiger en gitt sum. Eksempel:

```
Startbeløp: 1000
Rentesats (%): 3
Ønsket beløp: 1200
år 1 : 1030.0
år 2 : 1060.9
år 3 : 1092.73
år 4 : 1125.51
år 5 : 1159.27
år 6 : 1194.05
år 7 : 1229.87
```

13. Lag en funksjon som kan hjelpe dørvakten på et utested å holde styr på hvor mange gjester som er sluppet inn og sikre at antallet ikke overstiger kapasiteten i lokalet. Kapasiteten skal være parameter til funksjonen, og vakten skal kunne angi antall personer i hver gruppe som ankommer eller forlater lokalet. For eksempel kan dialogen se slik ut:

```
>>> kapasitetVakt(10)
-- Bruksanvisning --
Noen ankommer - tast 1
Noen går - tast 2
Avslutt - 0
-----
Ledig kapasitet: 10
Hva skjer? 1
Hvor mange kommer? 5
Ledig kapasitet: 5
Hva skjer? 1
Hvor mange kommer? 4
Ledig kapasitet: 1
Hva skjer? 2
Hvor mange går? 2
Ledig kapasitet: 3
Hva skjer? 1
Hvor mange kommer? 5
For mange! Slipp inn 3
Ledig kapasitet: 0
Hva skjer? 2
Hvor mange går? 4
Ledig kapasitet: 4
Hva skjer? 0
Takk for nå
>>>
```

Vanskelige oppgaver

14. I denne oppgaven skal du lage programmer for å spille Stein-Saks-Papir mot maskinen over flere runder. To varianter: 'Best av N runder' og 'Førstemann til N seiere'

a) Brukeren skal oppgi et antall runder som skal spilles. Den som vinner flest runder, vinner sammenlagt. Eksempel:

```
Hvor mange runder? 3
Runde 1 - Velg (1=papir,2=saks eller 3=stein):2
Maskinen velger 1
Det ble seier
Runde 2 - Velg (1=papir,2=saks eller 3=stein):3
Maskinen velger 3
Det ble uavgjort
Runde 3 - Velg (1=papir,2=saks eller 3=stein):2
Maskinen velger 2
Det ble uavgjort
--- Sammenlagt -----
Du vant 1 - 0 ( 2 uavgjorte )
```

b) Brukeren skal oppgi et antall runder som må vinnes for å vinne sammenlagt. Eksempel:

```
Førstemann til? 3
Velg (1=papir,2=saks eller 3=stein):1
Maskinen velger 2
Stillingen er 0 - 1
Velg (1=papir,2=saks eller 3=stein):2
Maskinen velger 1
Stillingen er 1 - 1
Velg (1=papir,2=saks eller 3=stein):1
Maskinen velger 3
Stillingen er 2 - 1
Velg (1=papir,2=saks eller 3=stein):3
Maskinen velger 1
Stillingen er 2 - 2
Velg (1=papir,2=saks eller 3=stein):2
Maskinen velger 1
Stillingen er 3 - 2
-----
Du vant!
```

15. Lag en funksjon `gangetabell` som gitt et tall n skriver ut gangetabellen for tallene fra 1 til n , altså en tabell på formen

1	2	3	4	...	n
2	4	6	8	...	$2n$
3	6	9	12	...	$3n$
...					
n	$2n$	$3n$	$4n$...	n^2

Funksjonen skal ha en opsjon som angir hvor mange plasser som settes av til hvert tall, standardverdien er 3 plasser

```
>>>gangetabell(5)
  1  2  3  4  5
  2  4  6  8 10
  3  6  9 12 15
  4  8 12 16 20
  5 10 15 20 25
>>>
>>>gangetabell(3,2)
 1 2 3
 2 4 6
 3 6 9
```

16. Skriv om `histogram`-funksjonen fra oppgave 2 slik at histogrammet skrives ut stående i stedet for liggende.

```
>>>histogram([3,5,7,2,9,8,1,5,0,7])
 *
 **
* ** *
* ** *
** ** *
** ** *
*** ** *
***** *
***** *
```

17. Lag en funksjon for et gjette-spill. Gitt et tall N skal det generere et tilfeldig tall mellom 1 og N og så la brukeren gjette hvilket tall det er. Hvis brukeren gjetter riktig, skal funksjonen skrive ut "Gratulerer, du gjettet riktig!" og antall forsøk det tok. Hvis brukeren gjetter feil, skal funksjonen gi hint om tallet er høyere eller lavere, og la brukeren gjette på nytt. Programmet skal avsluttes når brukeren treffer riktig tall.

```
>>>gjette_spill(100)
Gjett et tall mellom 1 og 100
Gjett tallet: 50
Tallet er høyere.
Gjett tallet: 75
Tallet er lavere.
Gjett tallet: 63
Tallet er lavere.
Gjett tallet: 56
Tallet er lavere.
Gjett tallet: 53
Tallet er lavere.
Gjett tallet: 52
Gratulerer, du gjettet riktig! Det trengte 6 forsøk.
```

18. Lag et program som skriver ut alle permutasjonene av bokstavene a,b og c. Dvs alle måter å stokke om 'abc' (inkludert 'abc' selv). Bruk `for`-løkker til å generere permutasjonene.

```
abc
acb
bac
bca
cab
cba
```


Ekstraoppgaver

19. (litt vanskelig)

Lag en funksjon som beregner tverrrsummen av et heltall, dvs summen av sifrene i tallet. For eksempel er tverrrsummen av 452 = 4+5+2 = 11

```
>>>tverrrsum(452)
11
```

20. (vanskelig)

Skriv en funksjon `lesHeltall` (*meldinger*, *Min*, *Max*) som skal lese inn et heltall fra tastaturet, i intervallet fra *Min* til *Max*. Parameteren *meldinger* er en liste av ledetekster som skal brukes i de ulike forsøkene brukeren får til å taste inn en gyldig verdi. Dvs den første meldingen brukes innledningsvis, dersom brukeren ikke gir et gyldig tall får han/hun et nytt forsøk med den andre meldingen osv inntil alle meldingene er brukt. Da får ikke brukeren flere forsøk og `None` blir returnert. La *Min* og *Max* være valgfrie, dvs du kan sette begge grensene, en av dem eller ingen. Eksempler:

```
>>> lesHeltall(['Gi et tall større enn 100: ']*3, Min=100) # husk at /liste*n gir n kopier av liste-innholdet
Gi et tall større enn 100: 132
132
>>>
>>> lesHeltall(['Gi et tall større enn 100: ']*3, Min=100)
Gi et tall større enn 100: 32
Gi et tall større enn 100: 99
Gi et tall større enn 100: 54
>>>
>>> M=['Barnets alder:', 'Gi et tall mellom 0 og 13:', 'MELLOM 0 OG 13 SA JEG!', 'SISTE SJANSE:']
>>>
>>> lesHeltall(M, Min=0, Max=13)
Barnets alder: 24
Gi et tall mellom 0 og 13: 15
MELLOM 0 OG 13 SA JEG! 38
SISTE SJANSE: 40
>>>
```

21. (vanskelig)

Lag funksjoner som gitt et siffer n (heltall mellom 1 og 9) skriver ut et kvadrat med sifrene opp til n .

a) Slik:

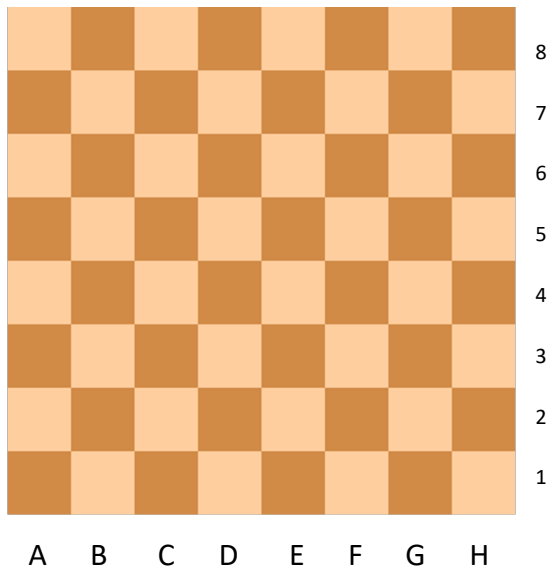
```
>>> tegnKvadrat(6)
1 1 1 1 1 1
2 2 2 2 2 2
3 3 3 3 3 3
4 4 4 4 4 4
5 5 5 5 5 5
6 6 6 6 6 6
```

b) Slik:

```
>>>tegnKvadrat(6)
1 2 3 4 5 6
1 2 3 4 5 6
1 2 3 4 5 6
1 2 3 4 5 6
1 2 3 4 5 6
1 2 3 4 5 6
```

22. (Vanskelig)

Lag et program som skriver ut fargene til alle rutene på sjakk-brettet



```
A1 : sort
A2 : hvit
A3 : sort
A4 : hvit
A5 : sort
A6 : hvit
A7 : sort
A8 : hvit
B1 : hvit
B2 : sort
B3 : hvit
B4 : sort
B5 : hvit
B6 : sort
...
```