

INFO110 Laboppgaver 5

1. Lag en funksjon `siste(sekvens)` som returnerer det siste elementet i en sekvens-verdi (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 på en line, 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 beregner tverrsummen av et heltall, dvs summen av sifrene i tallet. For eksempel er tverrsummen $452 = 4+5+2 = 11$

(hint: `sifferHøyre(tall)` fra oppgavene til forelesning 4 om funksjoner eller `str`- og `int`-konvertering).

4.

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)
Gi et tall større enn 100: 132
132
>>>
>>> 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: 32
Gi et tall større enn 100: 99
Gi et tall større enn 100: 54 #siste sjanse
>>>
>>> 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
>>>
```

5.

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.

Eksempel:

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

6.

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

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!
```

7.

a) Skriv en funksjon `blanke(n)` som skriver ut n blanke tegn (mellomrom), uten linjeskift til slutt. Eksempel:

```
>>> for i in range(0,5):
      blanke(i)
      print(i)

0
 1
  2
   3
    4
```

b) Skriv en funksjon som gitt et siffer n (heltall mellom 1 og 9) skriver ut et kvadrat med sifrene opp til n , slik som dette:

```
>>> 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
```

c) Skriv en funksjon som gitt et siffer n skriver ut en skrånende ruter-figur med sifrene opp til n , slik som dette. (Hint: `blanke` fra punkt a) kan være nyttig)

```
>>> tegnSkråRuter(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
```

d) Skriv en tilsvarende funksjon som i c) men der figuren skråner den andre veien og sifrene skrives i omvendt rekkefølge, slik:

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

e) Gi funksjonene i b), c) og d) en opsjon til et innrykk, dvs et gitt antall blanke til venstre for figuren. Eksempel med innrykk=10:

```
>>> tegnSkråRuterOmvendt(6,10)
      6 5 4 3 2 1
        6 5 4 3 2 1
          6 5 4 3 2 1
            6 5 4 3 2 1
              6 5 4 3 2 1
                6 5 4 3 2 1
```

(Dette eksemplet kunne også vært kalt slik:

```
>>> tegnSkråRuterOmvendt(6,innrykk=10))
```

8. Lag et program som skriver ut alle permutasjonene av bokstavene a,b og c.
Dvs alle måter å stokke om 'abc' (inkludert 'abc' selv).

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