

EKSAMENSFORSIDE

Skriftlig eksamen med tilsyn

Emnekode: IB 1020	Emnenavn: Python for beregning	
Dato: 1/12-2021	Tid fra / til: 09:00 – 12:00	Ant. timer: 3
Ansvarlig faglærer(e): Joakim Bjørk og Marius Lysaker		
Campus: Porsgrunn, Bakkenteigen og Kongsberg	Fakultet: Fakultet for teknologi, naturvitenskap og maritime fag	
Antall oppgaver: 4	Antall vedlegg: 0	Ant. sider inkl. forside og vedlegg: 5
Tillatte hjelpemidler (jfr. emnebeskrivelse): Ingen		
Opplysninger om vedlegg:		
Merknader: Alle delspørsmål blir vektet likt i sensureringen.		

Kryss av for type eksamenspapir

Ruter ☐Linjer ☐

OPPGAVE 1) FUNKSJONER OG FOR-LØKKE

- a) Overflatearealet og volumet av en kule er gitt ved følgende formler:

$$K_{overf} = 4\pi r^2 \quad \text{og} \quad K_{vol} = \frac{4}{3}\pi r^3.$$

Lag en funksjon med navn 'kule' med inn-argument r , hvor r angir radius på kula. Funksjonen skal så beregne kulas overflate og volum og returnere disse to verdiene.

Hint: tallverdien for π kan du få tilgang på ved å skrive `np.pi`

- b) Lag et program som bruker ei *for-løkke* til å beregne overflaten og volumet til en kule dersom radiusen r tar følgende verdier: $r = 150, 145, 140, \dots, 5$. Svarene skrives til skjerm.

Merk: denne deloppgaven kan besvares selv om du ikke fikk til deloppgave 1a).

OPPGAVE 2) FOR/WHILE OG IF-ELSE

Gitt en array med 500 elementer. Elementene i `arr1` består kun av heltallene 1, 2 og 3, og disse heltallene står i vilkårlig rekkefølge.

Du kan her bruke kommandoen: `arr1 = np.random.randint(1, 4, size=500)` til å opprette `arr1` med egenskapene beskrevet over.

- a) Skriv et program som søker gjennom `arr1` og hver gang et element i `arr1` er identisk lik 3 (dvs. element på plass nummer i er identisk lik 3), skrives indeks i til skjerm.
- b) Skriv et program som bruker en *for-løkke* og en *if/elif/else-test* til å finne antall 1'ere, antall 2'ere og antall 3'ere i `arr1`, og som så skriver svaret til skjerm.
- c) Skriv et program som visualiserer antall 1'ere, 2'ere og 3'ere i `arr1` ved bruk av et søylediagram (bar).

Fra deloppgave 2b) kan du anta følgende fordeling av henholdsvis 1'ere, 2'ere og 3'ere:
fordeling = [160, 145, 195]

Følgende hjelpetekst/dokumentasjon kan gi deg hint til hvordan man får plottet et søylediagram (bar):

matplotlib.pyplot.bar

```
matplotlib.pyplot.bar(x, height, width=0.8, bottom=None, *, align='center', data=None, **kwargs)
```

[\[source\]](#)

Make a bar plot.

The bars are positioned at *x* with the given *alignment*. Their dimensions are given by *height* and *width*. The vertical baseline is *bottom* (default 0).

Many parameters can take either a single value applying to all bars or a sequence of values, one for each bar.

Parameters:

x: float or array-like

The x coordinates of the bars. See also *align* for the alignment of the bars to the coordinates.

height: float or array-like

The height(s) of the bars.

width: float or array-like, default: 0.8

The width(s) of the bars.

- d) Lag en funksjon med navn **to_treere** som tar `arr1` som inn-argument. Funksjonen skal så bruke en *while-løkke* til å undersøke om to påfølgende elementer i `arr1` har verdi 3. Dersom to påfølgende elementer i `arr1` har verdi 3 avsluttes søket (dvs. løkka) og følgende tekst skrives til skjerm:

I `arr1` finnes det to påfølgende elementer som begge har verdi 3.

Dersom det ikke finnes to påfølgende elementer i `arr1` som begge har verdi 3 skrives følgende tekst til skjerm:

I `arr1` finnes det ikke to påfølgende elementer som begge har verdi 3.

Oppgave 3) SANNSYNLIGHET

Et idrettslag har gjennomført et loddsalg. Totalt er det solgt 2178 lodd, nummerert fra lodd nr. 1 og opp til lodd nr. 2178. Finn sannsynligheten for at dersom man trekker to lodd, så er begge lodd nr. større eller lik 2000.

Merk: samme lodd nr. kan ikke trekkes flere ganger.

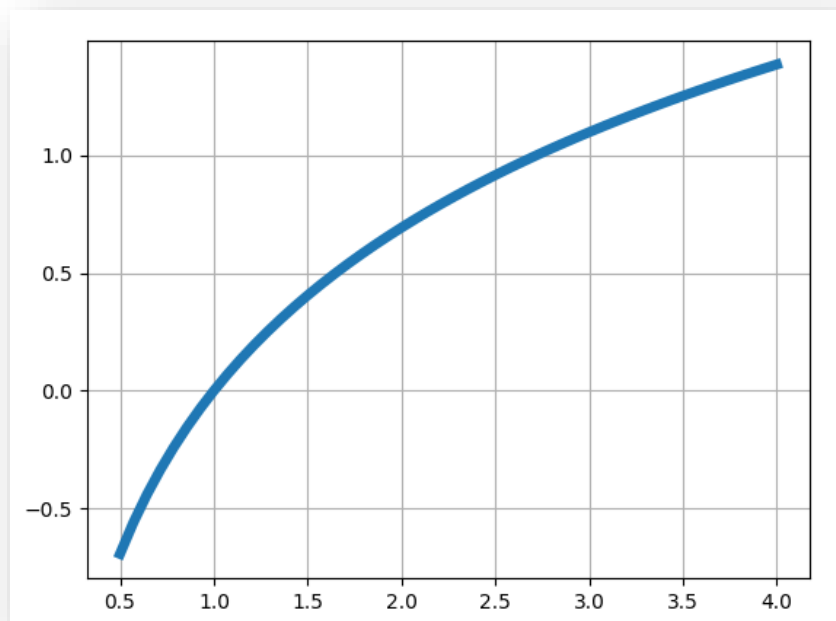
Koden under viser hvordan man kan trekke to tilfeldig heltall på intervallet $[1, 2178]$ uten 'tilbakelegging'. Du står fritt til å bruke denne teknikken (eller andre) for å trekke de to tilfeldige tallene.

```
import numpy as np

sample_space = np.linspace(1, 2178, 2178)
tall = np.random.choice(sample_space, 2, False)
```

OPPGAVE 4) OPTIMERING

Gitt funksjonen $f(x) = \ln(x)$ for $x \in [0.5, 4]$. I figuren under har vi plottet funksjonen $f(x)$ på det gitte intervallet.



KANDIDATEN MÅ SELV KONTROLLERE AT OPPGAVESETTET ER FULLSTENDIG

Skriv et program som estimerer minste avstand fra origo (punktet $(0, 0)$) til kurven f .

Hint1: $\ln(x)$ er gitt ved koden `np.log(x)` i Python.

Hint2: Avstanden fra $(0,0)$ til et vilkårlig punkt (x, y) er gitt ved $\sqrt{x^2 + y^2}$.

Lykke til!