

i Instruksjon til kandidaten

Institutt for informatikk og datateknologi

Eksamensoppgave i IT6203 Grunnleggende Programmering for Lærere

Faglig kontakt under eksamen: **Majid Rouhani (hoved), Elise K Vonstad (støtte)**

Kontaktinfo: **majid.rouhani@ntnu.no (besvares innen 20 min)**

Tlf (skal brukes hvis det haster): **90074710 (Majid) / 41688134 (Elise)**

Eksamensdato: **10.12.2018**

Eksamenstid (fra-til): **09:00-13:00**

Hjelpemiddelkode/Tillatte hjelpemidler: **Alle trykte og håndskrevne**

Annen informasjon:

- **Husk å les oppgaveteksten nøye.**
- **Pass på å disponer tiden din, og legg merke til at noen oppgaver har flere oppnåelige poeng og tar derfor lengre tid å løse.**
- **Det er ikke tillatt å samarbeide med andre under eksamen. Bruk av Slack i tidsrommet eksamen avholdes anses som samarbeid.**
- **Det er ikke tillatt å kjøre oppgavene i del 2 i egen programvare.**

Merk! Studenter finner sensur i Studentweb. Har du spørsmål om din sensur må du kontakte instituttet ditt. Eksamenskontoret vil ikke kunne svare på slike spørsmål.

1 Følg pseudokoden del 1

Følg pseudokoden. 2 deloppgaver.

Her er det to deloppgaver, der deloppgave 2 bygger på koden fra deloppgave 1.

Du blir bedt om å lage et program som samler informasjon om eierne av sykler i et register. Dette programmet skal lagre informasjon fra bruker om navn og adresse for eierne av spesifikke sykler, og man skal kunne søke etter eierinformasjon ut fra rammenummer. Skriv programmet etter pseudokoden under.

Del 1:

```
# Opprett en tom dictionary med navn "sykler"
```

```
# Lagre listen "rammenr" under som en variabel
```

```
# For hvert element i listen "rammenr", gjør følgende:
```

```
# - Ta inn et navn fra bruker (navnet på eieren av hver sykkel)
```

```
# - Ta inn en adresse fra bruker (adressen til eieren av hver sykkel)
```

```
# - Legg "rammenummer" som key, og navnet og adressen du tok inn fra bruker som values, i dictionaryen "sykler"
```

```
rammenr = WBK0132K, VE01562512D, S5A001234, WN17632Z, TA78317B, ME7265T
```

Skriv ditt svar her...

1	
---	--

Maks poeng: 15

2 Følg pseudokoden del 2

Bruk koden fra del 1 som utgangspunkt for koden din i denne oppgaven. Det vil si at du kan referere til variabler, funksjoner osv som ble definert i koden din fra del 1. Skriv programmet etter pseudokoden under.

Del 2:

Definer funksjonen "sok_register", som skal ta inn argumentet "ramme" og gjøre følgende:

- Søke i dictionaryen "sykler" etter rammenummeret som blir gitt som argument

- Returnere resultatet av søket

Definer funksjonen "main" som skal:

- Be om input fra bruker om hvilket rammenummer han/hun vil søke opp eiers informasjon om

- Kalle funksjonen "sok_register" og legge resultatet i en variabel med navn

- Printe ut resultatet av søket til konsoll

Funksjonen "main" skal kalles hvis brukeren ønsker å søke på flere rammenummer.

Skriv ditt svar her...

1	
---	--

Maks poeng: 15

3 Kodeforståelse 1

3.1 I koden under er det to feil i "main()" som gjør at resultatet ikke blir riktig. Forklar hva feilene er og hvorfor de gjør at kjøringen blir feil. (10p)

```
# Dette programmet bruker en dictionary over planeter med deres respektive avstand (i millioner km) til sola.
# Programmet skal:
# Ved hjelp av funksjoner:
# - Sjekke avstanden til sola for hver planet
# - Hente de fire planetene som er nærmest sola i en egen liste
# - Hente de fire planetene som er lengst fra sola i en egen liste
# Print listen over de nærmeste planetene til konsoll
# Print listen over de planetene lengst unna til konsoll

planeter = {'uranus': 2871, 'venus': 108, 'jupiter': 778, 'jorda': 149, 'mars': 227, 'merkur': 57, 'neptun': 4497, 'saturn': 1427}

# Hent og sorter avstandene
def get_dist(planeter):
    avst = []
    for key,value in planeter.items(): # Iterer over keys og values i dictionaryen
        avst.append(value)

    avstSort = sorted(avst) # Sorter verdiene

    # Bruk den sorterte lista for å finne de fire nærmeste og fire lengst unna
    kortAvst = avstSort[:4]
    langAvst = avstSort[4:]

    return kortAvst, langAvst

# Ut fra listene med avstander, hent navnet på planetene som er nærmest og lengst unna
def get_names(kortAvst, langAvst):
    kortAvstNavn = []
    langAvstNavn = []

    for key,value in planeter.items(): # Iterer over keys og values i dictionaryen
        if value in kortAvst: # Sjekk om value'en eksisterer i listen over de korteste avstandene
            kortAvstNavn.append(key) # Legg key'en til listen
        else:
            langAvstNavn.append(key) # Alle value'ene som ikke er i lista over de korteste avstandene skal legges i denne lista

    return kortAvstNavn, langAvstNavn

def main():
    korteAvst, lengstAvst = get_dist() # Hent sorterte lister avstander fra dict'en "planeter"

    kortAvst, langAvst = get_names(lengstAvst, korteAvst) # Ut fra listene over, hent navnene på planetene

    # Print de to listene med navn på planetene til konsoll
    print('Planetene nærmest sola er: ', kortAvst)
    print('Planetene lengst fra sola er: ', langAvst)

main() # Kall main for å kjøre programmet
```

Skriv ditt svar her...

Format
-
B
I
U
x₂
x²
I_x
📄
📁
↶
↷
↺
≡
⋮
Ω
📊
✎
Σ
✖

Words: 0

Maks poeng: 10










4 Kodeforståelse 2

2.2 Resultat av kjøringen (10p)

Hva er resultatet av kjøringen av koden under gitt x? Forklar kort hva koden gjør.

```
x = ['p','g','e','j','p','s','p','v','e','p','r','m','k','t','a','b','k','t','e']
bak = ""
for i in range(0,len(x),2):
    bak += x[i]
print(bak)
```

Skriv ditt svar her...

Format | **B** | *I* | U | x_2 | x^2 | I_x |  |  |  |  |  |  | Ω |  |  | Σ | ABC | 

Words: 0

Maks poeng: 10

5 Programmering 1

4.1 Programmering (15p)

Dette programmet skal gi brukeren informasjon om hvorvidt en bil holdt en gjennomsnittsfart som var høyere enn tillatt (80 km/t) på en strekning på 3500 meter.

Programmet skal gjøre følgende:

- Lag en funksjon "speedFromTime" som tar inn to argumenter: distanse og tid. Funksjonen skal beregne og returnere hastighet (m/s) ut fra antall sekunder og distanse. Hastighet = distanse/tid.

- Lag en funksjon "speedCheck" som tar inn hastighet som argument. Denne skal sjekke om hastigheten var over 80 km/t eller ikke, og returnere True eller False basert på resultatet. Husk at du må gjøre om resultatet fra speedFromTime fra m/s til km/t.

- Lag en funksjon "main" som:

1) Henter inn informasjon fra bruker om registreringsnummer og tiden (antall sekunder) den gjeldende bilen brukte på å kjøre strekningen.

2) Kaller funksjonen "speedFromTime"

3) Kaller funksjonen "speedCheck"

4) Hvis hastigheten var over 80 km/t skal det skrives en setning til bruker om registreringsnummer og hastighet

- Så lenge bruker ønsker å skrive inn ny informasjon skal "main" kalles

Skriv ditt svar her

1	
---	--

Maks poeng: 15

6 Programmering 2

4.2 Programmering (15p)

Dette programmet skal generere en liste på 100 tilfeldige tall mellom 3-1000. Print lista ut til konsoll. Lag en funksjon som sjekker om tallene i lista er primtall (altså kun er delelig med 1 og seg selv), og legger primtallene i en ny liste. Print ut lista over primtall til konsoll.

Her er en funksjon som sjekker om tall i en liste er primtall:

```
def is_prime(n):  
    for i in range(3, n):  
        if n % i == 0:  
            return False  
    return True
```

Skriv ditt svar her...

1	
---	--

Maks poeng: 15

7 Programmering 3 del 1

4.3.1 Programmering

Dette er en oppgave med to deler. Koden fra del 1 skal også brukes i del 2.

Del 1(10p): Stave navn baklengs

Lag en funksjon som tar inn et navn. Funksjonen skal snu rekkefølgen på bokstavene i navnet, og returnere navnet stavet baklengs.

La brukeren skrive inn et navn som skal sendes til funksjonen.

Skriv ditt svar her...

1	
---	--

Maks poeng: 10

8 Programmering 3 del 2

4.3.2 Programmering

Del 2 (10p):

Denne delen av programmet skal bruke koden fra del 1. Det vil si at du kan referere til variabler, funksjoner osv som ble definert i koden din fra del 1.

Nå skal du skrive en funksjon som lar brukeren skrive inn navnet fra del 1 i koden baklengs, og så sjekker om rekkefølgen på bokstavene er riktig sammenlignet med resultatet fra funksjonen fra del 1. Hvis navnet blir stavet riktig baklengs skal det printes ut en beskjed om det til bruker, men hvis navnet er feil stavet så skal han/hun få mulighet til å prøve på nytt.

Skriv ditt svar her...

1	
---	--

Maks poeng: 10