NTNU
Kunnskap for ei betre verd

Institutt for datateknikk og informasjonsvitskap

Eksamensoppgåve i TDT4110 Informasjonsteknologi - grunnkurs

Fagleg kontakt under eksamen:	Guttorm Sindre	Mobil: +47 944 30245
	Aleksander Rognhaugen	Mobil: +47 901 14409
	Sigbjørn Lund Olsen	Mobil: +47 414 73502

Eksamensdato:	**2014-12-06**
Eksamenstid (frå-til):	**09:00 – 13:00**
Hjelpemiddelkode/Tillatne hjelpemiddel:	**Godkjend kalkulator**

Annan informasjon:

Oppgåvesettet inneheld 4 oppgåver. Det er oppgitt i prosent kor mykje kvar oppgåve og kvar deloppgåve tel ved sensur. Les igjennom heile oppgåvesettet før du byrjar å løyse oppgåvene. Disponer tida godt! Gjer rimelege antakingar der du meiner oppgåveteksten er ufullstendig, skriv kort kva du antar.

Svar kort og klart, og skriv tydeleg. Er svaret uklart eller lenger enn naudsynt, trekkjer dette ned.

Målform/språk:	**Nynorsk**
Tal på sider:	**17 (inkl. framside, svarark og appendiks)**

Innhald:

Kontrollert av:

_____

Dato Sign

Oppgåve 1: Fleirvalsoppgåve (25%)

Bruk dei to vedlagde svarskjemaa for a˚ svarepå denne oppgåva (ta vare på det eine sjølv). Du kan fa˚ nytt ark av eksamensvaktene dersom du treng dette. Berre eitt svar er heilt riktig. For kvart spørsmål gir korrekt avkryssing 1 poeng. Feil avkryssing eller meir enn eitt kryss gir −1/2 poeng. Blankt svar gir 0 poeng. Du får ikkje mindre enn 0 poeng totalt på denne oppgåva. Der det er spesielle uttrykk, står den engelske oversetjinga i parentes.

1. Kva for ein fundamental aktivitet innan programvareutviklingsprosessen fokuserer på å endre programvaren for å møte endra kunde- og marknadskrav?
    a. Programvarespesifikasjon
    b. Programvareutvikling
    c. Programvarevalidering
    d. Programvareevolusjon

2. Kva er eit analogt signal?
    a. Eit kontinuerleg signal der den variable eigenskapen er gitt av ein diskret funksjon, som gir verdiar frå eit definert og avgrensa område.
    b. Eit kontinuerleg signal der ein variabel eigenskap (t.d. amplitude eller frekvens) representerer informasjonen som blir overført.
    c. Eit diskret signal som blir representert ved hjelp av nullar og einarar.
    d. Ein kombinasjon av alternativ a og b.

3. Kva for ein type løkkestruktur er garantert å utføre handlinga minst éin gong?
    a. pre-test løkke (pretest loop).
    b. post-test løkke (posttest loop).
    c. begge typar.
    d. ingen av typane.

4. Omtrent kor mange gonger raskere er ein 1 GHz - prosessor i forhold til ein på 2 MHz.
    a. Halvparten så rask.
    b. Like rask.
    c. Dobbelt så rask.
    d. 500 gonger så rask.

5. Kva seier Nyquist-regelen?
    a. at samplingsrate ved lyd må vere minst det dobbelte i forhold til høgste frekvensen.
    b. at lyd over 20000Hz ikkje kan høyrast av mennesket sitt øyre.
    c. at tapsfri komprimering ikkje er mogeleg for lyd.
    d. at lyddata tapsfritt kan komprimerast med maksimalt ein faktor 2*pi.

6. Kva for ein programvareprosessmodell bør veljast for eit prosjekt der det skal utviklast eit heilt nytt system der eksisterande komponentar ikkje finst og kunden er usikker på korleis systemet skal vere?
    a. Vassfallsmodellen.
    b. Inkrementell utvikling.
    c. Gjenbruksorientert systemutvikling.
    d. Havmodellen.

7. Kva er føremålet med ein paritetsbit i digitale signal?
    a. Fortel kvar meldinga skal sendast.
    b. Gjer meldingane raskare å overføre (komprimering).
    c. Hjelper til å oppdage feil i digitale signal.
    d. Krypterer signal så overføringa av data blir sikrare.

8. Kompleksiteten til sortering ved innsetjing (insertion sort) er
    a. $\Theta(n)$.
    b. $\Theta(n \log n)$.
    c. $\Theta(n^2)$.
    d. $\Theta(2n)$.

9. Ein moderne prosessor er typisk bygd opp av mange millionar små…
    a. Diodar.
    b. Magnetar.
    c. Transistorar.
    d. Kondensatorar.

10. Ein byte med minne i datamaskinen kan lagre kor mykje?
    a. 16 bits.
    b. 8 flyttal.
    c. fire ASCII-teikn.
    d. ein heiltalsverdi mellom 0 og 255.

11. Kva for ein av dei følgjande er ein kjent fordel med vassfallsmodellen?
    a. Tar omsyn til brukarkrav som endrar seg i løpet av prosjektet.
    b. Gjer prosessen synleg og enklare å monitorere for prosjektleiaren.
    c. Får tidlege versjonar av systemet raskt ut til kunden.
    d. Opnar for kontinuerleg tilbakemelding frå brukarane av systemet.

12. Morsekode representerer bokstavar som sekvensar av prikk og strek som er
    a. like lange for alle bokstavar i alfabetet.
    b. kortare for bokstaver tidleg i alfabetet, lenger for bokstaver sist i alfabetet.
    c. kortare for vokalar, lenger for konsonantar.
    d. kortare for bokstaver som førekjem hyppig i vanleg tekst, lenger for sjeldnare bokstavar.

13. Kva for ein av desse er ei korrekt attgjeving av teoriboka sin definisjon av ei algoritme? "Ei algoritme er eit ordna sett av…"
    a. "… eintydige, utførbare steg som definerer ein terminerande prosess" (unambiguous, executable steps that defines a terminating process).
    b. "… eintydige, effektive steg som definerer ein utførbar prosess" (unambiguous, efficient steps that defines an executable process).
    c. "… velforma, effektive steg som definerer ein terminerande prosess" (well-formed, efficient steps that defines a terminating process).
    d. "…velforma, utførbare steg som definerer ein effektiv prosess" (well-formed, efficient steps that defines an efficient process).

14. En datamaskin går i ei uendelig løkke som blir kalla
    a. Det naturlige krinslaupet.
    b. Hent-Utfør krinslaupet.
    c. Det evige krinslaupet.
    d. Utreknings- krinslaupet.

15. Kva er korrekt binær representasjon av 'NTNU' i 8 bits ASCII?
    a. `01001110 01010100 01001110 01010101`.
    b. `01100001 01100100 01110011 01100110`.
    c. `01101110 01110101 01110100 01001110`.
    d. `01100010 01010101 01010010 01010000`.

16. I kva høve er det mest nyttig å bruke gjenbruksorientert systemutvikling?

a. Når det finst tilgjengeleg programvare som kan gjere oppgåver systemet skal utføre.
b. Når det skal lagast programvare for å handtere resirkulering av søppel eller liknande system.
c. Når det skal gjenbrukast idear frå tidligare prosjekt.
d. Når det skal gjenbrukast systemutviklarar og systemdesignarar frå tidligare prosjekt.

17. Kva står forkortinga ISP for?
    a. Internet Service Provider.
    b. Information Security Protocol.
    c. Internet Security Protocol.
    d. Information Super Pool.

18. Kompleksiteten til binærsøk er
    a. $\Theta(n)$ dersom lista er sortert og $\Theta(n \log n)$ dersom den er usortert.
    b. $\Theta(\log n)$ dersom lista er sortert og $\Theta(2 \log n)$ dersom lista er usortert.
    c. $\Theta(\log n)$ dersom lista er sortert og $\Theta(n)$ dersom lista er usortert.
    d. $\Theta(\log n)$ dersom lista er sortert. Binærsøk er ubrukeleg dersom lista er usortert.

19. RAM
    a. Hugsar alle verdiane når straumen blir kutta.
    b. Er alltid delt inn i blokker på 1 kilobyte.
    c. Betyr Random Access Memory.
    d. Kan trygt fjernast utan at maskinen sluttar å fungere.

20. Kva står bokstavane i RGB for?
    a. Red, Green, Blue.
    b. Readable Graphics Byte.
    c. Raster Grayscale Balance.
    d. Realtime GPU Backlog.

21. Kva kallar vi aktiviteten som har fokus på å identifisere den overordna strukturen for eit system og kva sub-system det skal ha?
    a. Hovuddesign.
    b. Arkitekturdesign.
    c. Interfacedesign.
    d. Komponentdesign.

22. MODEM er ei forkorting for
    a. MOdulator / DEModulator.
    b. Massive Online Digital Electric Messaging.
    c. MOnitored Data EMission.
    d. Mapping Of Digital Electronic Mail.

23. ASCII-kode representerer bokstavane A til Z som sekvensar av 0 og 1 som er
    a. like lange for disse bokstavane i alfabetet.
    b. kortare for bokstavar tidleg i alfabetet, lenger for bokstavar sist i alfabetet.
    c. kortare for vokalar, lenger for konsonantar.
    d. kortare for bokstaver som førekjem hyppig i vanleg tekst, lenger for sjeldnare bokstavar.

24. Eit nettverk som knyt saman datamaskiner og utstyr i eit avgrensa område som eit kontor, bygning eller i ein bustad blir omtala med forkortinga:

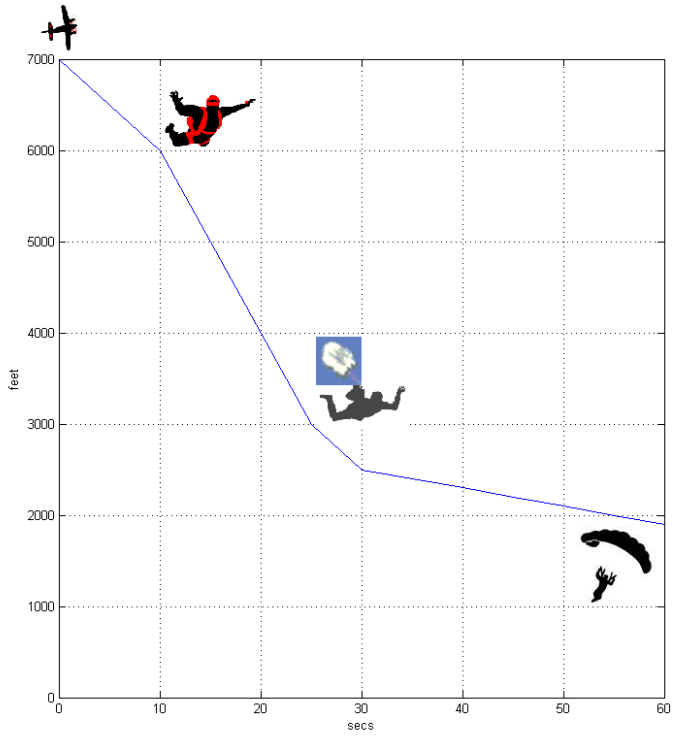
a. LAN.
b. MAN.
c. PAN.
d. WAN.

25. VPN (Virtual Private Network) kan gi mottakaren inntrykk av at ein reisande tilsett sin berbare PC er seg innanfor bedrifta sitt nettverk ved at meldingar frå denne PC'en
    a. blir plasserte inni ein kryptert datapakke for ekstern oversending.
    b. blir sende med ei tidsforseinking.
    c. blir sende ekstra raskt, med høg prioritet.
    d. blir sende med ei falsk avsendaradresse som inneheld eit virus.

Oppgåve 2 Programmering Fallskjerm (25%)

(I denne oppgåva kan det vere gunstig å kalle funksjonar som du har laga i tidligare deloppgåver. Sjølv om du ikkje har fått til den tidligare oppgåva, kan du kalle funksjon derifrå som om den verkar som spesifisert i oppgåveteksten.)

NTNU Fallskjermklubb treng hjelp til å lage eit nytt opplærings- og administrasjonsprogram. Dei har bede firmaet ditt (ITGK) om hjelp, og du har fått jobben med å programmere funksjonar som skildra i deloppgåvene under.
Vi nyttar ein litt forenkla versjon av Jorda sine fysiske lover:



Figur 1. Hopp frå 7000 fot.

Ein fallskjermhoppar fell (med konstant/gjennomsnittleg fart) 100 fot pr. sekund dei 10 første sekunda, og deretter med konstant toppfart på 200 fot pr. sekund til skjermen må opnast i 3000 fots høgde (sjå figur 1). Om ein mot normalt hoppar ut under 3000 fot må skjermen utløysast med éin gong (etter 0 sekund).

Medlemsdatabasen til NTNU-FSK ligg lagr på ei fil 'members.txt', med følgjande format:

NAMN ; ID ; VEKT ; SKJERMST.
Døme på innhaldet i fila:

```
Frank Stank;D-49334;75;120
Bjarne Stor;C-49335;95;150
Dumbo Ear;D-50105;450;750
Peter Pan;A-12345;30;100
```

Oppgåve 2a (5%)
Lag ein funksjon `inputPerson` som les inn namn, id, vekt og skjermstorleik frå tastaturet, og returnerer ei liste med verdiar for namn, id, vekt og skjermstorleik. Merk at namn og id er tekststrengar, medan vekt og skjermstorleik er heiltal. Du kan anta at brukaren skriv inn lovlege verdiar, og det trengst ikkje unnatakshandtering.
Døme på køyring (tekst med understreking blir skriven inn av brukaren):

```
>>> person = inputPerson()
Name: Fredrik Olsen
ID: B-77777
KG: 80
Size: 240
>>> person
['Fredrik Olsen', 'B-77777', 80, 240]
>>>
```

Oppgåve 2b (5%)
Lag ein funksjon `readDbFile` som les inn heile medlemsbasen til ein tabell (liste av lister) med følgjande element: `name`, `id`, `weight` og `size` (sjå forklaring ovanfor). `name` og `id` er tekststrengar, medan `weight` og `size` er heiltal. Du kan anta at fila finst, at det ikkje oppstår nokon problem ved opning/lukking, og at fila ikkje inneheld nokon blanke eller ugyldige liner. Funksjonen skal ha inn-parameter `filename` og returnere tabellen (liste av liste).
Døme på køyring:

```
>>> db=readDbFile('members.txt')
>>> for line in db:
      print(line)

['Frank Stank', 'D-49334', 75, 120]
['Bjarne Stor', 'C-49335', 95, 150]
['Dumbo Ear', 'D-50105', 450, 750]
['Peter Pan', 'A-12345', 30, 100]
>>>
```

Oppgåve 2c (5%)
Lag ein funksjon `printMemberList` som skriv ut følgjande overskrifter og innhaldet av tabellen `db` (lista av lister som forklart ovanfor) på skjermen på følgjande format:
NAMN (avsett 15 teikn) ID-NR (avsett 9 teikn) VEKT (avsett 5 siffer/teikn) kg.
SKJERMSTORLEIK (avsett 4 siffer/teikn) kvadratfot
Inn-parameter `db`, ingen retur-verdi.
Døme på bruk, gitt at db inneheld tabellen vist i oppgåve 2b:

```
>>> printMemberList(db)
NAMN            ID-NR   VEKT kg. SKJERMSTORLEIK
Frank Stank     D-49334   75 kg  120 kvadratfot
Bjarne Stor     C-49335   95 kg  150 kvadratfot
Dumbo Ear       D-50105  450 kg  750 kvadratfot
Peter Pan       A-12345   30 kg  100 kvadratfot
>>>
```

Oppgåve 2d (5%)
Lag ein funksjon `addPerson` med inn-parameteren `filename` (der databasen er lagra). Funksjonen skal be brukaren skrive inn informasjon om ein person med namn, id, vekt og skjermstorleik og lagre dette i variabelen `person`. Funksjonen skal så lese inn databasen som ligg lagra i fila `filename` til datastrukturen `db` som er ein tabell (liste av lister) som forklart i oppgåve 2b. Deretter skal opplysningane om den nye personen leggjast til i `db` og i fila `filename`. Bruk riktig format: sjå «Døme på innhaldet i fila» ved figur 1 (sjå side 6). Returverdien `db` skal innehalde den oppdaterte databasen.
Døme på køyring (alt som er understreka er informasjon skrive inn av brukaren, og alt som er i feit skrift er køyring av funksjonen og endringa av resultatet ved køyring):

```
>>> db=addPerson('members.txt')
Name: Santa Klaus
ID: H-12345
KG: 155
Size: 380
>>>
>>> printMemberList(db)
NAMN             ID-NR   VEKT kg. SKJERMSTORLEIK
Frank Stank      D-49334   75 kg  120 kvadratfot
Bjarne Stor      C-49335   95 kg  150 kvadratfot
Dumbo Ear        D-50105  450 kg  750 kvadratfot
Peter Pan        A-12345   30 kg  100 kvadratfot
Santa Klaus      H-12345  155 kg  380 kvadratfot
>>>
```

Oppgåve 2e (5%)
For ein fallskjermhoppar er det veldig viktig å vere klar over kor mange sekund ein kan vente før ein må opne fallskjermen (sjå figur 1). Lag ein funksjon `feet2seconds` som reknar ut kor mange sekund det tar å falle frå ei oppgitt høgd i fot ned til 3000 fot (inn-parameter `feet`, og retur-verdi `seconds`). Bruk informasjon gitt i starten av oppgåve 2 (forklaringa til figur 1) til å rekne ut riktig tid. Dersom høgda i fot er under 3000 skal funksjonen returnere 0.
Døme på bruk:

```
>>> feet2seconds(12500)
52.5
>>> feet2seconds(7000)
25.0
>>> feet2seconds(2000)
0
>>>
```

Oppgåve 3 Programmering Vêrstasjon (30%)

Du skal behandle data frå ein vêrstasjon for ei rekkje dagar. Data ligger lagra som flyttal i ein tabell (liste av lister) som heiter `weatherData`. Kvar rekkje (kvar liste i lista) representerer måledata for *éin dag* og har tre element av ulike typar måledata. Dei tre typane med måledata er maksimumstemperatur, minimumstemperatur og nedbørsmengd. Vi refererer til dagane slik at data i første rekkje (`weatherData[0]`) er for dag nr 1, data for andre rekkje (`weatherData[1]`) er for dag nr 2 osb. Døme på utskrift av liner i `weatherData` kan vere:

```
>>> for row in weatherData:
      print(row)

[12.0, 2.4, 8.2]
[6.1, 0.6, 11.9]
[8.3, -3.5, 0.0]
[11.6, -5.2, 0.0]
[15.3, 2.8, 14.3]

>>> weatherData[0]
[12.0, 2.4, 8.2]
>>> weatherData[1]
[6.1, 0.6, 11.9]
>>>
```

Oppgåve 3a (10%)

a) Skriv ein funksjon `weatherStats` som tar inn `weatherData` som parameter. Funksjonen skal gå igjennom datalista og utifrå det skrive eit samandrag som vist under. Det vil seie at den skal skrive ut talet på dagar i perioden, total nedbørmengd i heile perioden, og vise den aller lågaste og aller høgste temperaturen saman med *nummeret* på dagane for desse temperaturane.
Døme på køyring dersom data er som vist i den grå boksen ovanfor:

```
>>> weatherStats(weatherData)
There are 5 days in the period.
The highest temperature was 15.3 C on day number 5
The lowest temperature was -5.2 C on day number 4
There was a total of 34.4 mm rain in the period
>>>
```

Oppgåve 3b (10%)

b) Skriv ein funksjon `coldestThreeDays` som tar inn parameteren `weatherData` (som definert over). Funksjonen skal finne den perioden av tre samanhengande dagar som hadde den lågaste gjennomsnittlege minimumstemperaturen. Den skal returnere nummeret på første dagen i denne tredagarsperioden. Dersom det er fleire periodar som er like kalde, så skal den returnere berre den siste av desse periodane. Eit døme på ei køyring av denne funksjonen for `weatherData` som definert tidlegare i oppgåva gir:

```
>>> coldestThreeDays(weatherData)
2
>>>
```

Oppgåve 3c (10%)
Vêrstasjonen har nettopp rapportert data for ytterlegare ein dag. Den kjem som ein tekststreng lagra i variabelen `extraData` som har formatet som vist under:

```
>>> extraData
'max=23.5, min=9.3, 5.1mm'
>>>
```

Skriv ein funksjon `addNewDay` som tar `extraData` samt `weatherData` som parameterar, og som returnerer ein ny versjon av `weatherData` som er oppdatert med dei nye dataa på slutten av tabellen. Under er vist eit døme på køyring (sjølve køyringa og endringar er i feit skrift):

```
>>> for row in weatherData:
      print(row)

[12.0, 2.4, 8.2]
[6.1, 0.6, 11.9]
[8.3, -3.5, 0.0]
[11.6, -5.2, 0.0]
[15.3, 2.8, 14.3]
>>>
>>> extraData
'max=23.5, min=9.3, 5.1mm'
>>>
>>> weatherData = addNewDay(extraData,weatherData)
>>> for row in weatherData:
      print(row)

[12.0, 2.4, 8.2]
[6.1, 0.6, 11.9]
[8.3, -3.5, 0.0]
[11.6, -5.2, 0.0]
[15.3, 2.8, 14.3]
[23.5, 9.3, 5.1]
>>>
```

Oppgåve 4 Kodeforståing (20%)

Oppgåve 4a (5%)
Kva blir returnert ved køyring av funksjonen `myst([1,2,3,3,2,1])` med kode som vist under? (3 poeng)
Forklar med ei setning kva funksjonen `myst()` gjer? (2 poeng)

```
def myst(A):
  L=len(A)
  if (L>1):
    B=A[0]*A[L-1]
    return B+myst(A[1:L-1])
  return 0
```

Oppgåve 4b (5%)
Kva blir skrive ut når ein køyrer koden nedanfor? (3 poeng)
Forklar med ei setning kva funksjonen `myst_b( )` gjer? (2 poeng)

```
def myst_b(W):
  # create a 2d-list with W x W zeros
  table = [[0 for i in range(W)] for j in range(W)]
  for a in range(W):
    table[a][a]=1
    b=0
    while a-b > 0 and a+b < W-1:
      b+=1
      table[a-b][a+b]=1
      table[a+b][a-b]=1
  return table

for line in myst_b(4):
  print(line)
```

Oppgåve 4c (5%)
Kva blir returnert ved køyring av funksjonen `myst_c('RBHOOASDUEØGNGEBLSOIURMNGTD')` med kode som vist under? (3 poeng)
Forklar med ei setning kva funksjonen `myst_c()` gjer? (2 poeng)

```
def myst_c(A):
  B=''
  for x in range(0,len(A),3):
    B=B+A[x]
  return B
```

Oppgåve 4d (5%)
I eit program som skal trene ungdomsskuleelevar i matematikk, treng vi ein funksjon for å sjekke korrekt nøsting av parentesar i uttrykk der tre ulike parentestypar er tillatne. Funksjonen treng ikkje å sjekke at uttrykket elles er fornuftig, berre at parentesar kjem i lovleg rekkjefølgje og går opp mhp talet på start- og sluttparentesar. Vi har også skrivet tre print-setningar som kallar funksjonen for å teste om den verkar. Koden er vist her (linenummer til venstre ikkje del av koden men er tatt med så du lettare kan vise til spesifikke kodeliner i svaret ditt):

```
1     PAREN=['(','[','{',')',']','}']
2
3     def test(expression):
4       parentheses_list = []
5       for char in expression:
6         if char in PAREN[:3]: # start-parenthesis found
7           # put corresponding end-parenthesis in the beginning of the list
8           parentheses_list.append(PAREN[PAREN.index(char)+3])
9         elif char in PAREN[3:]: # end-parenthesis found
10          if char not in parentheses_list: # not matched start-parenthesis
11            return False
12          else:
13            parentheses_list.remove(char)
14      return parentheses_list
15
16    def main():
17      print('A:', test('{a+4*[b-2*(c+5)]/11}'))   #should give True
18      print('B:', test('{a+4*[b-2*(c+5])/11}'))   #should give False
19      print('C:', test('{a+4*[b-2*(c+5])/11}}'))  #should give False
```

Som kommentarane seier, skulle utskrifta frå dei tre print-setningane ha blitt True, False og False, men når vi køyrer `main`-funksjonen får vi:

```
>>> main()
A: []
B: []
C: False
>>>
```

Det er to feil i koden:

(1) funksjonen returnerer tom liste (`[]`) i staden for boolske variable for både A og B. Dersom denne feilen blir retta, vil køyring av programmet indikere den andre feilen:

```
>>> main()
A: True
B: True
C: False
>>>
```

(2) som vist over: funksjonen returnerer True for somme uttrykk som skulle gitt False, som for linje B her

Spørsmål: Forklar kva for kodeliner som er årsak til feil (1) og feil (2) og korleis dei enklast kan rettast. I begge høve skal det vere mogeleg å rette feilen berre ved å endre noko i eksisterande kodeliner, det skal ikkje vere naudsynt å leggje til nye kodeliner.

Appendiks: Nyttige funksjonar

Built-in:

format(numeric_value, format_specifier)
: Formats a numeric value into a string according to the format specifier, which is a string that contains special characters specifying how the numeric value should be formatted. Examples of various formatting characters are "f=floating-point, e=scientific notation, %=percentage, d=integer". A number before the formatting character will specify the field width. A number after the character "." will format the number of decimals.

%
: Remainder (or modulo operator): Divides one integer by another and gives the remainder

len(s)
: Return the length (the number of items) of a string, tuple, list, dictionary or other data structure.

int(x)
: Convert a string or number to a plain integer.

float(x)
: Convert a string or a number to floating point number.

str([object])
: Return a string containing a nicely printable representation of an object.

String methods:

s.isalnum()
: Returns true if the string contains only alphabetic letters or digits and is at least one character of length. Returns false otherwise.

s.isalpha()
: Returns true if the string contains only alphabetic letters, and is at least one character in length. Returns false otherwise.

s.isdigit()
: Returns true if the string contains only numeric digits and is at least one character in length. Returns false otherwise.

s.isspace()
: Returns true if the string contains only whitespace characters, and is at least one character in length. Returns false otherwise. (Whitespace characters are spaces, newlines (\n), and tabs (\t)).

s.ljust(width)
: Return the string left justified in a string of length width.

s.rjust(width)
: Return the string right justified in a string of length width.

s.lower()
: Returns a copy of the string with all alphabetic letters converted to lowercase.

s.upper()
: Returns a copy of the string with all alphabetic letters converted to uppercase.

s.strip()
: Returns a copy of the string with all leading and trailing white space characters removed.

s.strip(char)
: Returns a copy of the string with all instances of *char* that appear at the beginning and the end of the string removed.

s.split(str)
: Returns a list of all the words in the string, using str as the separator (splits on all whitespace if left unspecified).

s.endswith(substring)
: The substring argument is a string. The method returns true if the string ends with substring.

s.startswith(substring)
: The substring argument is a string. The method returns true if the string starts with substring.

s.find(substring)
: The substring argument is a string. The method returns the lowest index in the string where substring is found. If substring is not found the method returns -1.

s.replace(old, new)
: The old and new arguments are both strings. The method returns a copy of the string with all instances of old replaced by new.

List operations:

s[i:j:k]
Return slice starting at position i extending to position j in k steps. Can also be used for strings.

item in s
Determine whether a specified item is contained in a list.

min(list)
Returns the item that has the lowest value in the sequence.

max(list)
Returns the item that has the highest value in the sequence.

s.append(x)
Append new element x to end of s.

s.insert(index,item)
Insert an item into a list at a specified position given by an index.

s.index(*item*)
Return the index of the first element in the list containing the specified item.

s.pop()
Return last element and remove it from the list.

s.pop(i)
Return element i and remove it from the list.

s.remove(item)
Removes the first element containing the item.

s.reverse()
Reverses the order of the items in a list.

s.sort()
Rearranges the elements of a list so they appear in ascending order.

Dictionary operations:

d.clear()
Clears the contents of a dictionary

d.get(key, default)
Gets the value associated with a specific key. If the key is not found, the method does not raise an exception. Instead, it returns a default value.

d.items()
Returns all the keys in a dictionary and their associated values as a sequence of tuples.

d.keys()
Returns all the keys in a dictionary as a sequence of tuples.

d.pop(key, default)
Returns the value associated with a specific key and removes that key-value pair from the dictionary. If the key is not found, the method returns a default value.

d.popitem()
Returns a randomly selected key-value pair as a tuple from the dictionary and removes that key-value pair from the dictionary.

d.values()
Returns all the values in dictionary as a sequence of tuples.

Files

open()
Returns a file object, and is most commonly used with two arguments: open(filename, mode). Mode can be ‘r’ (read only), ‘w’ (writing only), ‘a’ (appending), ‘r+’ (both reading and writing).

f.read(size)
Reads data from file and returns it as a string. Size is an optional and if left out the whole file will be read.

f.readline()
Reads a single line from the file (reads until newline character (\n) is found), and returns it as a string.

f.readlines()
Reads data from the file and returns it as a list of strings.

f.write(string)
Writes the contents of string to file.

f.close()
Close the file and free up any system resources taken up by the open file.

Svarskjema fleirvalsoppgåve

Kandidatnummer: _____ Program: _____

Fagkode: _____ Dato: _____

Antall sider: _____ Side: _____

Oppgåvenr	*A*	*B*	*C*	*D*
1.1				
1.2				
1.3				
1.4				
1.5				
1.6				
1.7				
1.8				
1.9				
1.10				
1.11				
1.12				
1.13				
1.14				
1.15				
1.16				
1.17				
1.18				
1.19				
1.20				
1.21				
1.22				
1.23				
1.24				
1.25				

Denne sida er med vilje blank!

Svarskjema fleirvalsoppgåve

Kandidatnummer: _____ Program: _____

Fagkode: _____ Dato: _____

Antall sider: _____ Side: _____

Oppgåvenr	*A*	*B*	*C*	*D*
1.1				
1.2				
1.3				
1.4				
1.5				
1.6				
1.7				
1.8				
1.9				
1.10				
1.11				
1.12				
1.13				
1.14				
1.15				
1.16				
1.17				
1.18				
1.19				
1.20				
1.21				
1.22				
1.23				
1.24				
1.25				