

Institutt for datateknikk og informasjonsvitenskap

Eksamen i TDT4110 IT-grunnkurs, med Python

Faglig kontakt under eksamen:	Terje Rydland: 957 73 463 Guttorm Sindre: 944 30 245
Eksamensdato:	2016-12-06
Eksamenstid (fra-til):	09:00 – 13:00
Hjelpemiddelkode/Tillatte hjelpemidler:	D – Bare bestemt, enkel kalkulator

Annen informasjon:

Oppgavesettet inneholder 4 oppgaver. Det er angitt i prosent hvor mye hver oppgave og hver deloppgave teller ved sensur. Les igjennom hele oppgavesettet før du begynner å løse oppgavene. Disponer tiden godt! Gjør rimelige antagelser der du mener oppgaveteksten er ufullstendig, skriv kort hva du antar.

Svar kort og klart, og skriv tydelig. Er svaret uklart eller lenger enn nødvendig trekker dette ned.

HUSK å rive løs og levere et flervalgsoppgave svarark!

Innhold:

- Oppgave 1: Flervalgsoppgave (25%)
- Oppgave 2: Kodeforståelse (20%)
- Oppgave 3: Programmering Valg (25%)
- Oppgave 4: Programmering Penger (30%)
- Appendix: Potensielt nyttige funksjoner og metoder
- Svarskjema til flervalgsoppgave (2 eksemplarer)

Målform/språk:

Bokmål

Antall sider:

17 (inkl. Forside, Appendiks, og 2x svarark)

Informasjon om trykking av eksamensoppgave

Originalen er:

1-sidig 2-sidig

sort/hvit farger

skal ha flervalgskjema

Kontrollert av:

15. des. 2016

Dato

Sign.

Oppgave 1: Flervalgsoppgave (25%)

Bruk de to vedlagte svarskjemaene for å svare på denne oppgaven (ta vare på det ene selv). Du kan få nytt ark av eksamensvaktene dersom du trenger dette. Kun ett svar er helt riktig. For hvert spørsmål gir korrekt avkryssing 1 poeng. Feil avkryssing eller mer enn ett kryss gir $-1/2$ poeng. Blankt svar gir 0 poeng. Du får ikke mindre enn 0 poeng totalt på denne oppgaven. Der det er spesielle uttrykk står den engelske oversettelsen i parentes.

1. Hva er forventet klokkehastighet på en moderne datamaskin?
 - a. Litt over 1 kHz
 - b. Litt over 1 MHz
 - c. Litt over 1 GHz
 - d. Litt over 1 THz
2. Hva skiller en Solid-State Drive (SSD) fra andre Hard-Drives (HD)?
 - a. Den inneholder mer solide magnet-plater.
 - b. Den er basert på en gammeldags produksjonsteknikk.
 - c. Den har ingen bevegelige deler.
 - d. Den spinner mye raskere.
3. Hva er den vanligste bruken av Cyan-Magenta-Yellow-Key (CMYK)?
 - a. Blande farger i nøkkelmagasinet
 - b. Blande farger på en skjerm
 - c. Blande farger på en prosjektør
 - d. Blande farger på en utskrift
4. Hvor stor er de minste moderne transistorene i IC-kretser?
 - a. Ca. 10 mm
 - b. Ca. 10 mikrometer
 - c. Ca. 10 nanometer
 - d. Ca. 10 picometer
5. Hvilket steg i «fetch-execute cycle» flytter verdiene fra Minnet til ALU?
 - a. IF
 - b. ID
 - c. DF
 - d. EX
6. Hva er det største tallet som kan representeres med et 64-bits "signed integer" som «two's complement»?
 - a. $2^{64}-1$
 - b. $2^{63}-1$
 - c. 2^{65}
 - d. 2^{64}
7. Hva er tallverdien i desimalsystemet av det heksadesimale tallet 23F?
 - a. 575
 - b. 792
 - c. 1048
 - d. 312
8. I RGB-systemet med 24-bits fargekoder, hvilken farge er FFFF00
 - a. rødt
 - b. grønt
 - c. grått
 - d. gult

9. Hva er representasjon i binærtallsystemet av det desimale tallet 1234?
 - a. 0100 1101 0010
 - b. 0010 0101 1011
 - c. 0101 1011 0111
 - d. 1101 0011 0100
10. Hvordan påvirker sampleraten kvaliteten i en analog-til-digital-konvertering av lyd?
 - a. den påvirker ikke kvaliteten
 - b. høyere samplerate gjør at man får med høyere frekvenser
 - c. høyere samplerate gjør at man får med sterkere lyder
 - d. høyere samplerate gjør at man produserer mindre data
11. Hva er pakkeswitching?
 - a. En metode en maskin bruker for å motta meldinger
 - b. En protokoll en maskin bruker for å kontakte en annen maskin på nettet
 - c. En måte å dele opp meldinger i mindre pakker som kan sendes over nettet
 - d. En avleggs metode ARPANet brukte inntil 1970 til å sende meldinger på nettet
12. I hvilket av lagene i OSI-modellen finner vi IP-protokollen?
 - a. Application
 - b. Transport
 - c. Network
 - d. Link
13. Hvilken av disse feildeteksjonsmetodene er best til å oppdage "burst"-feil?
 - a. Paritet
 - b. Sjekksum
 - c. CRC
 - d. RAC
14. Hva menes med "den digitale dividende"?
 - a. betegnelse på forskjellen mellom en digital og en analog ressurs
 - b. betegnelsen på ledige (ikke-tildelte) frekvensressurser i det elektromagnetiske spektrum
 - c. forskjellen mellom en digitalt kompetent og en ikke digitalt kompetent person
 - d. uttrykk for en digital divisjonsoperasjon i en CPU
15. Hvordan kan vi forsikre oss om at avsender er den han/hun sier han/hun er?
 - a. Avsenderen har kryptert meldingen med sin private nøkkel
 - b. Avsenderen har kryptert meldingen med sin offentlige nøkkel
 - c. Avsenderen har brukt sjekksum og paritet
 - d. Mottagerens e-post program kan sammenligne mottatt e-post med den som ligger i avsenderens ut-post

Følgende koden viser en Python-implementasjon av en sorteringsalgoritme kjent som boblesortering. Innparameteren `li` er ei liste, og funksjonen skal returnere samme liste sortert. Spørsmål 16, 17, og 18 angår algoritmen vist i figuren.

```
def bubblesort(li):
    swapped = True
    while swapped:
        swapped = False
        for i in range(0, len(li)-1):
            if li[i] > li[i+1]:
                temp = li[i]
                li[i] = li[i+1]
                li[i+1] = temp
                swapped = True
    return li
```

16. Vil algoritmen fungere for å sortere elementene i lista li?
- Ja, den vil sortere stigende.
 - Ja, den vil sortere synkende.
 - Den vil bare fungere for lister av heltall, ikke flyttall.
 - Nei, den vil ikke fungere.
17. Forutsatt at algoritmen fungerer, hva er dens tidskompleksitet (worst case) hvis vi betegner lengda av lista som n?
- $\Theta(n)$
 - $\Theta(n \log n)$
 - $\Theta(n^2)$
 - $\Theta(n^3)$
18. Best case for denne algoritmen er hvis lista tilfeldigvis er riktig sortert allerede, hva er tidskompleksiteten da?
- $\Theta(\log n)$
 - $\Theta(n)$
 - $\Theta(n \log n)$
 - $\Theta(n^2)$
19. En funksjon utfører flyttallsmultiplikasjoner på tall som den får inn som argument i form av ei liste. Hvis antall element i lista er n, kan antall flyttallsmultiplikasjoner som utføres uttrykkes som $f(n) = 2n^3 - n^2 - 3n + 10$. Hva blir tidskompleksiteten for denne funksjonen?
- $\Theta(n)$
 - $\Theta(n \log n)$
 - $\Theta(n^2)$
 - $\Theta(n^3)$
20. Hva er betingelsen for å kunne bruke binærsøk på ei liste av tall?
- Tallene bruker bare 0 og 1 som siffer.
 - Lista er sortert stigende.
 - Lista er sortert, enten stigende eller synkende.
 - Lista har et odde antall elementer.
21. En kjent ulempe med inkrementell utvikling er at
- Tidskompleksiteten til systemet stiger kvadratisk med antall inkremitter
 - Kostnaden til systemet stiger kvadratisk med antall inkremitter
 - Inkrementelle algoritmer kun fungerer for sorterte data
 - Strukturen til systemet har en tendens til å gradvis forringes.

22. Med gjenbruk av programvare (software reuse) menes i pensum at ...
- flere brukere benytter samme abonnement for en app for å spare penger
 - en programmodul skrevet for ett system også kan benyttes i annet system
 - gamle mikroprosessorer kan smeltes om så materialene kan gjenbrukes
 - straks vi har skrevet et program, kan vi kjøre det om og om igjen
23. Fire sentrale designaktiviteter nevnt i teoriboka er arkitekturdesign, databasedesign, systemgrensesnittdesign (system interface design) og ...
- fildesign
 - komponentdesign
 - løkkedesign
 - matrisedesign
24. I systemutvikling (software engineering) betyr iterativ utvikling (iterative development) at
- Man benytter løkker i programkoden
 - Man benytter rekursjon i programkoden
 - At man gjør en ovenfra og ned (top-down) inndeling av systemet
 - At man leverer stadig nye versjoner av systemet, med gradvis økende funksjonalitet
25. Begrepet sort-boks testing (black box testing) innebærer at
- Man tester programmets kun ut fra dets ytre grensesnitt, ikke indre oppbygning
 - At man prøver å hacke programmet for å finne sikkerhetshull
 - At man tester programmet samtidig som det er i operasjon
 - At man tester programmets indre, uten tanke på brukergrensesnittet

Oppgave 2: Kodeforståelse (20%)

Oppgave 2a (5%)

Hva blir skrevet ut til skjerm når du kjører programmet vist under? (3%)

Forklar med en setning hva funksjonen `flopp` gjør (2%)

```
def flopp(mat):
    r = len(mat)
    c = len(mat[0])
    for i in range(r):
        for j in range(c):
            if mat[i][j] == 1:
                mat[i][j] = 0
            else:
                mat[i][j] = 1
    return mat

def main():
    M = [ [0, 1, 0], [1, 0, 1], [0, 1, 0] ]
    print(flopp(M))

main()
```

Oppgave 2b (5%)

Hva blir skrevet ut på skjermen når koden vist under blir kjørt? (3%)

Forklar med en setning hva funksjonen `compute` gjør (2%)

```
def compute(d, m, y):
    M = ('Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec')
    N = ('st', 'nd', 'rd', 'th')
    x = (d % 30) % 20 - 1
    if x > 3 or x == -1:
        x = 3
    return str(d) + N[x] + ' ' + M[m-1] + ' ' + str(y)

print(compute(30, 11, 1337))
```

Oppgave 2c (5%)

Hva blir skrevet ut på skjermen hvis koden vist under blir kjørt? (3%)

Forklar med en setning hva funksjonen `fr` gjør (2%)

```
def fr(s):
    f = [0] * 26
    for i in range(len(s)):
        if s[i] >= 'a' and s[i] <= 'z':
            f[ord(s[i]) - ord('a')] += 1
    a = max(f)
    b = chr(f.index(a) + ord('a'))
    return a, b

x, y = fr('abcdbcdbd')
print(x, y)
```

Oppgave 2d (5%)

Hva blir skrevet ut på skjermen hvis koden vist under blir kjørt? (3%)

Forklar med en setning hva funksjonen `f` gjør (2%)

```
def f(x):
    y = 0
    while x > 0:
        y = y + x % 10
        x = int(x / 10)
    if y >= 10:
        y = f(y)
    return y

print(f(32145))
```

Oppgave 3: Programmering Valg (25%)

I et land langt borte skal det avholdes valg. Hvert valgdistrikt telles manuelt. Du skal skrive kode for å regne sammen resultatene som rapporteres inn. Landet er delt inn i 92 valgdistrikt som hver velger én representant til parlamentet, samt at presidenten velges ved flertall blant de stemmeberettigete på tvers av alle valgdistriktene.

Du kan anta at innparameterne til funksjonene alltid inneholder lovlige verdier.

Oppgave 3a: Initialisering (5%)

Det er registrert fem ulike partier i landet, og navnene er definert i en liste kalt `parties`. Skriv en funksjon `initElection` som oppretter en tabell for å lagre stemmetall for hvert parti i hvert av de 92 valgdistriktene, og la funksjonen returnere denne tabellen initialisert til situasjonen før valget - dvs at alle har 0 stemmer.

Eksempel på bruk av funksjonen til å opprette tabellen med 92 linjer og en kolonne for hvert parti, pluss utlistering av de tre første linjene er vist her:

```
>>> parties = ['TeaParty', 'CoffeeParty', 'MilkParty', 'HouseParty', 'BeachParty']
>>> election = initElection(parties)
>>> election[:3]
[[0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0]]
>>>
```

Oppgave 3b: Oppdatering (5%)

Etter hvert som resultatene strømmer på, må de settes inn i tabellen som ble opprettet av funksjonen i oppgave (a). Skriv en funksjon `updateElection` som tar tre parametere inn: tabellen med valgdata, nummeret på valgdistriktet, og en liste med stemmetall for de deltakende partiene, i samme rekkefølge som de er listet i variabelen `parties`. Hvert valgdistrikt kan ha flere stemmelokaler, slik at det kan bli lagt inn stemmer for et distrikt flere ganger. Nye tall som kommer inn skal plusses sammen med tall som ligger der fra før. Funksjonen skal returnere en oppdatert tabell.

Eksempel på kjøring:

```
>>> election = initElection(parties)
>>> election = updateElection( election, 34, [123, 3321, 3442, 23, 1])
>>> election[34]
[123, 3321, 3442, 23, 1]
>>> election = updateElection( election, 34, [601, 2000, 3000, 50, 22])
>>> election[34]
[724, 5321, 6442, 73, 23]
```

Oppgave 3c: Presidentvalget (5%)

Skriv funksjonen `printLeadP` som tar resultatene i tabellen `election` med valgdata, og regner ut hvilket parti som totalt har fått flest stemmer, og som skriver ut at dette partiet leder.

I denne deloppgaven kan du gjøre en forenklet antagelse at funksjonen `printLeadP` kun kalles etter at det har blitt registrert noen stemmer (dvs. ikke hele tabellen er null), og at det ikke er delt ledelse mellom flere partier, dvs. ett unikt parti har flest stemmer.

Eksempel på kjøring (NB: eksemplene her og videre utover antar at det er satt inn vesentlig mer data i tabellen enn bare det som ble gjort i eksemplet i 3b):

```
>>> printLeadP(election, parties)
MilkParty is leading the election with 341433 votes
>>>
```

Oppgave 3d: Parlamentsvalget (10%)

Siden det kun er en person som velges i hver enmannskrets, er det partiet med flest stemmer i kretsen som får valgt inn sin kandidat.

Skriv funksjonen `printResults` som skriver ut på skjerm gjeldende mandatfordeling i parlamentet basert på data som hittil har kommet inn for valgdistriktene. For å få full pott må tallene komme ut høyrejustert på en jevn kolonne, som vist under. Hvis et parti kun har 1 delegat hittil, skal man skrive ordet `delegate` i entall, ellers `delegates` (også for null). Det kan antas at maksimalt antall delegater er tresifret. I denne deloppgaven skal det tas høyde for at to eller flere partier kan ligge likt i teten i et valgdistrikt. Hvis ett parti er alene i teten i et valgdistrikt, skal distriktet telles for dette partiet. Hvis flere partier deler ledelsen, skal distriktet telles som `Undecided (tied)`. Hvis det ennå ikke har blitt registrert noen stemmer i distriktet, skal det derimot telles som `Undecided (no votes)`.

Eksempel på kjøring:

```
>>> printResults(table)
TeaParty:                3 delegates
CoffeeParty:             12 delegates
MilkParty:                1 delegate
HouseParty:              4 delegates
BeachParty:              20 delegates
Undecided (tied):        2 delegates
Undecided (no votes):    50 delegates
>>>
```


Oppgave 4: Programmering Penger (30%)

Firmaer som setter opp kjøpskontrakter har ofte behov for å oppgi beløpet som skal betales både med tall og tekst. Store beløp blir en lang tekst og ansatte har lett for å skrive feil slik at tall og tekst ikke stemmer overens. Man ønsker derfor et program som automatisk kan oversette tall til tekst. I denne oppgaven skal tallene omformes til engelsk tekst. Du kan anta at det allerede er deklart to globale variable D og L som du kan benytte deg av i koden din hvis du ønsker det, men det er ikke noe krav at disse må brukes.

Skriv funksjonene under slik at du kan gjenbruke dem. Hvis du ikke klarer å løse en del-oppgave kan du likevel anta at funksjonen virker som oppgitt og gjenbruke den i en senere del-oppgave. Du kan anta at funksjonene alltid får inn lovlige verdier.

D og L inneholder følgende konstante verdier (skal ikke endres underveis).

```
D = {1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five',
     6: 'six', 7: 'seven', 8: 'eight', 9: 'nine', 10: 'ten',
     11: 'eleven', 12: 'twelve', 13: 'thirteen', 14: 'fourteen',
     15: 'fifteen', 16: 'sixteen', 17: 'seventeen',
     18: 'eighteen', 19: 'nineteen', 20: 'twenty', 30: 'thirty',
     40: 'forty', 50: 'fifty', 60: 'sixty', 70: 'seventy',
     80: 'eighty', 90: 'ninety'}
```

```
L = [ 1000000, ' million', 1000, ' thousand', 1, '']
```

Oppgave 4a: 2-digit (7%)

Skriv en funksjon `i2_txt` som kan ta som argument et positivt heltall med maksimalt to siffer (dvs. $1 \leq \text{tallet} \leq 99$) og returnere samme tall som engelsk tekst. Eksempel på bruk:

```
>>> i2_txt(5)
'five'
>>> i2_txt(30)
'thirty'
>>> i2_txt(68)
'sixty-eight'
```

Oppgave 4b: 3-digit (7%)

Skriv en funksjon `i3_txt` som kan ta som argument et positivt heltall med maksimalt tre siffer (dvs. $1 \leq \text{tallet} \leq 999$) og returnere samme tall som engelsk tekst. Tall med bare ett eller to siffer skal gi samme resultat som i (a). Eksempel på bruk med tresifrede tall:

```
>>> i3_txt(100)
'one hundred'
>>> i3_txt(345)
'three hundred forty-five'
```

Oppgave 4c: 9-digit (7%)

Skriv en funksjon `i9_txt` som kan ta som argument et positivt heltall med maksimalt ni siffer (dvs. $1 \leq \text{tallet} \leq 999\,999\,999$) og returnere samme tall i dets tekstlige form. Tall med tre eller færre siffer skal gi resultat som vist i (a) og (b). Eksempel på bruk med større tall:

```
>>> i9_txt(12000)
'twelve thousand'
>>> i9_txt(276900)
'two hundred seventy-six thousand nine hundred'
>>> i9_txt(67000020)
'sixty-seven million twenty'
```

Oppgave 4d: Sett inn tekst (9%)

Skriv en funksjon `add_words` som kan ta inn som argument en setning (tekststreng) som potensielt inneholder ett eller flere beløp gitt som en serie siffer, og returnere samme setning med tekstlige varianter av beløpene innskutt bak tallet omsluttet av tankestreker. Hvis man ikke finner noen tall, skal setningen være uendret, og du kan anta at det kun er snakk om heltall, ikke beløp med desimaler. Eksempler på bruk:

```
>>> add_words('Mr. X shall pay 9005100 dollars')
'Mr. X shall pay 9005100 - nine million five thousand one hundred -
dollars'
>>> add_words('C owes 91 pounds to D and 55 pounds to E')
'C owes 91 - ninety-one - pounds to D and 55 - fifty-five - pounds to E'
>>> add_words('The evildoer is hereby fined 945000000 yen')
'The evildoer is hereby fined 945000000 - nine-hundred forty-five million - yen'
```

Appendix: Potensielt nyttige funksjoner og metoder

Built-in:

`%`

Remainder (modulo operator): Divides one number by another and gives the remainder.

`except ExceptionName as variable:`

Catches an exception as default error message.

`except ExceptionName:`

Typical ExceptionNames are ValueError, ZeroDivisionError, IOError and Exception.

`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.

`int(x)`

Convert a string or number to a plain integer.

`float(x)`

Convert a string or a number to floating point number.

`len(s)`

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

`round(number[, ndigits])`

Returns the floating point value number rounded to ndigits digits after the decimal point. If ndigits is omitted, it returns the nearest integer to its input.

`range(start, stop[, step])`

Rather than being a function, range is actually an immutable sequence type, as documented in Ranges and Sequence Types — list, tuple, range.

`str([object])`

Return a string containing a nicely printable representation of an object.

String methods:

`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.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.join(list)`

Returns a string joining all the strings in the list into one string, with the string *s* inserted as a separator between each element.

`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()`

Closes the file and free up any system resources taken up by the open file.

Denne siden er med hensikt blank!

Svarskjema til flervalgsoppgave

Kandidatnummer: _____ Program: _____

Fagkode: _____ Dato: _____

Antall sider: _____ Side: _____

<i>Oppgavenr</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
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 siden er med hensikt blank!

Svarskjema flervalgsoppgave

Kandidatnummer: _____ Program: _____

Fagkode: _____ Dato: _____

Antall sider: _____ Side: _____

<i>Oppgavenr</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
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				