

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 en protokoll i kontekst av digital kommunikasjon?
 - a. Et sett av regler man kan velge fritt mellom som gjør kommunikasjon effektiv og rask.
 - b. Et internasjonalt lovverk som bestemmer hva man kan gjøre og hva man ikke kan gjøre på internett som bruker.
 - c. En stor datafil som inneholder hvordan alle nettverk i verden er koblet sammen.
 - d. Et regelverk som bestemmer hvordan kommunikasjon skal foregå og hvilke funksjoner som kan brukes.
2. Hva kalles den avsluttende aktiviteten i programvarevalideringsfasen?
 - a. Utviklingstesting.
 - b. Systemtesting.
 - c. Totaltesting.
 - d. Akseptansetesting.
3. Hvilken sikkerhetsutfordring er ”pharming”?
 - a. Bruker blir ledet til en falsk versjon av en offisielt nettside.
 - b. Bruker mottar en falsk epost som utgir seg fra å komme fra en avsender du stoler på.
 - c. Bruk av offisielle navn på institusjoner eller personer for å lure til seg konfidensiell informasjon.
 - d. Massivt angrep på webtjenere slik at tjenesten ikke blir tilgjengelig for tiltenkte brukere.
4. Hva står forkortelsen VPN for?
 - a. Virtual Protocol Node.
 - b. Viral Privacy Node.
 - c. Virtual Private Network.
 - d. Volatile Performance Network.
5. Hvis man skal sende data over nettet, kan elektrisk støy under transporten føre til at data utilsiktet endres. Bruk av en sjekksum (checksum) bakerst i en tallserie som skal overføres vil sikre at
 - a. tallene kommer uendret fram til mottager
 - b. mottager kan korrigere eventuelle endringer av tallene under overføring
 - c. mottager kan oppdage, men ikke korrigere, endringer av tallene under overføring
 - d. mottager kan oppdage endring såfremt kun ett tall er blitt endret
6. DSL-kommunikasjon over telefonlinjer har den egenskapen at
 - a. det dessverre er umulig å bruke vanlig telefon samtidig som noen bruker internettet.
 - b. at man kan bruke vanlig telefon samtidig, men med noe forstyrrelser i lyden.
 - c. at man også kan bruke vanlig telefon med grei lyd, men at data over nettet da går tregt.
 - d. at man kan bruke vanlig telefon samtidig, uten forstyrrelser verken for telefon eller internett.

7. I forbindelse med pensumstoffet om nettverkskommunikasjon står forkortelsen DSL for
 - a. Data Source Location.
 - b. Domain Support License.
 - c. Digital Subscriber Line.
 - d. Direct Signal Link.
8. Binærsøk er mer effektiv enn sekvensielt søk fordi
 - a. binærsøk også virker på sorterte lister, mens sekvensielt søk kun fungerer på usorterte lister.
 - b. binærsøk er rekursiv mens sekvensielt søk er iterativ.
 - c. binærsøk eliminerer halve datamengden for hvert oppslag mens sekvensielt søk bare eliminerer det ene elementet som ble testet.
 - d. binære søk virker på binærtall mens sekvensielt søk bare virker på tekst.
9. Anta at vi har fire mulige algoritmer for å løse et problem, hvorav en er $\Theta(n^3)$, en er $\Theta(n^2)$, en er $\Theta(n \log n)$ og en er $\Theta(2^n)$. Hvis vi skulle rangere disse med den mest effektive først, den minst effektive sist, blir det:
 - a. $\Theta(n \log n)$, $\Theta(n^2)$, $\Theta(n^3)$, $\Theta(2^n)$.
 - b. $\Theta(n^2)$, $\Theta(n \log n)$, $\Theta(2^n)$, $\Theta(n^3)$.
 - c. $\Theta(n \log n)$, $\Theta(n^2)$, $\Theta(2^n)$, $\Theta(n^3)$.
 - d. $\Theta(2^n)$, $\Theta(n^3)$, $\Theta(n \log n)$, $\Theta(n^2)$.
10. Alle eksterne enheter som skal kommunisere med en datamaskin krever en ... for å fungere
 - a. Driver.
 - b. ASCII-tabell.
 - c. Ledning.
 - d. Internett-forbindelse.
11. Sekundærminne.
 - a. Husker data etter at strømmen kuttes.
 - b. Er alltid flyktig.
 - c. Er alltid kjappere enn primærminnet.
 - d. Er kun skrivbart.
12. 129.241.103.4 er et eksempel på
 - a. En IP-adresse.
 - b. En MAC-adresse.
 - c. En datamaskin sitt telefonnummer.
 - d. Et TCP-nummer.
13. Ved hjelp av autentisering kan vi
 - a. Gjøre et usikkert nett helt sikkert.
 - b. Forsikre oss om at avsenderen er den han utgir seg for.
 - c. Skjule innholdet i en internett-melding.
 - d. Automatisk rette opp feil i dataoverføringen.
14. Hvordan representeres tallet -4 som toer-komplement?
 - a. 11111100.
 - b. 10000100.
 - c. 00110100.
 - d. 11111001.

15. Syv-bits kode for E er 1000101 og for S 1010011. Hvis man skal legge til en paritetsbit (parity bit) til disse to, blir denne paritetsbiten
 - a. 0 for både E og S
 - b. 0 for E, 1 for S
 - c. 1 for E, 0 for S
 - d. 1 for både E og S
16. I forbindelse med pensumstoffet om nettverkskommunikasjon står forkortelsen ISP for
 - a. Intelligent Security Protection
 - b. Internet Service Provider
 - c. Internet Security Policy
 - d. Information Standard Protocol
17. Hastigheten for internettaksess med DSL for en hjemmedatamaskin når det gjelder nedlasting (overføre data fra internett til hjemmemaskinen) og opplasting (overføre data fra maskinen til internett) er vanligvis
 - a. symmetrisk, dvs. nedlasting og opplasting går like raskt relativt til datamengde
 - b. asymmetrisk, nedlasting går kjappere enn opplasting
 - c. asymmetrisk, opplasting går kjappere enn nedlasting
 - d. asymmetrisk i perioder med mye nett-trafikk, symmetrisk i perioder med liten trafikk (for eksempel om natten)
18. Alle farger på en skjerm kan lages ved å kombinere fargene:
 - a. Turkis (Cyan), lilla (Magenta), gul (Yellow)
 - b. Sort, hvitt, turkis (Cyan), lilla (Magenta), gul (Yellow)
 - c. Sort, hvitt, rød, gul, blå
 - d. Rød, grønn, blå
19. Mindre kretskort som kan plugges i hovedkortet (Motherboard) kalles:
 - a. Mikroprosessor
 - b. Datter-kort
 - c. Barn-kort
 - d. RAM
20. Hvilken påstand stemmer om Solid State Drive (SSD)?
 - a. Tregere enn harddisk
 - b. Laget av "Flash Memory"
 - c. Består av en stabel av metallskiver med en arm som sveiper over/mellom skivene
 - d. Tåler dårlig støt

Oppgave 2 Kodeforståelse (15%)

- a (3%)** Hva blir skrevet ut når man kjører koden nedenfor?
b (2%) Forklar med en setning hva funksjonen `tricky_a()` gjør?

```
def tricky_a(v):
    w=[]+v
    if len(w)>1:
        w[0]=round((v[0]+v[1])/2,2)
        w[len(w)-1]=round((v[len(v)-1]+v[len(v)-2])/2,2)
        for i in range(1,len(v)-1):
            w[i]=round((v[i-1]+v[i]+v[i+1])/3,2)
    return w

print(tricky_a([20,30,10,20,10]))
```

- c (3%)** Hva blir skrevet ut når man kjører koden nedenfor?
d (2%) Forklar med en setning hva funksjonen `myst2()` gjør?

```
def myst2(A):
    for x in range(1,len(A)):
        if (A[x]==A[x-1]+1):
            return True
    return False

print(myst2([1,3,5,4,7,13,10,2,32,31]))
```

- e (3%)** Hva blir skrevet ut når man kjører koden nedenfor?
f (2%) Forklar med en setning hva funksjonen `myst3()` gjør?

```
def myst3(A,x):
    for r in range(0,len(A)):
        for c in range(0,len(A[0])):
            if (A[r][c]==x):
                return r*c
    return 0

print(myst3([[1,2,3],[4,5,6],[7,8,9]],8))
```

Oppgave 3 Programmering Yatzy (15%)

Du kan anta at alle funksjonene mottar gyldige argumenter (inn-verdier).

I denne oppgaven skal du skrive noen funksjoner til spillet Yatzy. I Yatzy spiller man med 5 terninger der målet er å få høyest poengsum på ulike sammensetninger av terninger som ett par, to par, tre like, fire like, liten straight, stor straight, hus, sjanse og yatzy.



Oppgave 3 a) (3%)

Lag funksjonen **throw** som har inn-parameteren **n**. Funksjonen skal returnere en liste med **n** tilfeldige verdier mellom 1 og 6.

Eksempel på kjøring av funksjonen og hva den returnerer:

```
>>> throw(5)
[5, 2, 6, 3, 3]
```

Oppgave 3 b) (3%)

Lag funksjonen **chance** som har inn-parameteren **dice** som er en liste med fem elementer med verdier mellom 1 og 6. Funksjonen skal returnere summen av alle elementene i lista og skal lages uten ved hjelp av innebygde funksjoner for å summere lister.

Eksempel på kjøring av funksjonen og hva den returnerer:

```
>>> dice = [5, 2, 6, 3, 3]
>>> chance(dice)
19
```

Oppgave 3 c) (4%)

Lag funksjonen **house** som har inn-parameteren **dice** som er en liste med fem elementer med verdier mellom 1 og 6. Funksjonen skal returnere summen av alle terningene hvis verdiene i **dice** har både 3 like og ett par (f.eks. 4,4,4,2,2 eller 1,1,6,6,6), hvis ikke skal verdien 0 returneres. Funksjonen skal også returnere verdien 0 hvis alle elementene i **dice** er like.

Eksempel på kjøring av funksjonen og hva den returnerer:

```
>>> dice=[1,3,1,1,3]
>>> house(dice)
9
>>> dice=[4,3,3,3,4]
>>> house(dice)
17
>>> dice=[2,2,2,2,2]
>>> house(dice)
0
>>> dice=[1,3,4,2,3]
>>> house(dice)
0
```

Oppgave 3 d) (5%)

Lag funksjonen **straight** som har inn-parameteren **dice** som er en liste med fem elementer med verdier mellom 1 og 6. Funksjonen skal undersøke om listen representerer en liten eller stor straight. Følgende skal returneres:

For en liten straight (liste som inneholder tallene 1,2,3,4,5) skal funksjonen returnere tallet 15. For en stor straight (liste som inneholder tallene 2,3,4,5,6) skal funksjonen returnere tallet 20. For en liste som er verken liten eller stor straight skal funksjonen returnere tallet 0.

Eksempel på kjøring av funksjonen og hva den returnerer:

```
>>> dice=[4,2,1,3,5]
>>> straight(dice)
15
>>> dice=[5,6,4,2,3]
>>> straight(dice)
20
>>> dice=[1,1,2,3,3]
>>> straight(dice)
0
```

Oppgave 4 Programmering vitneobservasjoner (45%)

I noen av oppgavene kan det være gunstig å kalle funksjoner som du har laget i tidligere deloppgaver. Selv om du ikke har fått til den tidligere oppgaven, kan du kalle funksjon derfra med antagelse om at den virker som spesifisert i oppgaveteksten.

Politiet trenger et system for å sjekke om vitneobservasjoner av kjøretøyer fra hendelser som etterforskes, stemmer med faktiske kjøretøyer i et register.

Oppgave 4 a) (5%)

Skriv en funksjon `les_inn_bilinfo()` som leser inn fra tastatur vitnets observasjon av bilmerke, modell og farge for et kjøretøy. Funksjonen skal returnere disse tre opplysningene i en liste.

Eksempel på kjøring (det i fete typer tastes inn av brukeren):

```
>>> les_inn_bilinfo()
Hvilket bilmerke var det? FIAT
Hvilken modell? Uno
Hvilken farge? Rød
['FIAT', 'Uno', 'Rød']
>>>
```

Oppgave 4 b) (5%)

Skriv en funksjon `sjekk_bil()` som sammenligner to lister som hver inneholder tre tekststrenger, der den ene lista representerer en vitneobservasjon og den andre en faktisk bil. I vitneobservasjonen kan felt inneholde '?' som betyr at vitnet var usikker på den informasjonen. Funksjonen skal returnere True eller False. True hvis det er full match eller hvis avvik kun gjelder '?', False hvis det fins avvik som ikke er '?'. Eksempler på kjøring:

```
>>> sjekk_bil(['FIAT', 'Uno', 'Rød'], ['FIAT', 'Uno', 'Rød'])
True
>>> sjekk_bil(['FIAT', 'Uno', 'Rød'], ['FIAT', 'Uno', 'Blå'])
False
>>> sjekk_bil(['FIAT', 'Uno', '?'], ['FIAT', 'Uno', 'Rød'])
True
>>> sjekk_bil(['FIAT', 'Uno', '?'], ['FIAT', 'Punto', 'Rød'])
False
>>>
```

Oppgave 4 c) (5%)

Gitt tuppelet `SKILTBOKSTAV = ('A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'J', 'K', 'L', 'N', 'P', 'R', 'S', 'T', 'U', 'V', 'X', 'Y', 'Z', '?')`

Dette inneholder bokstaver som er lov å bruke på moderne norske bilskilt, samt '?' helt bakerst (hvis vitnet ikke husker). Skriv en funksjon `les_gyldig_vitneskilt()` som leser inn fra tastatur en streng på nøyaktig 7 tegn, hvorav de 2 første tegnene skal være tegn fra tuppelet `SKILTBOKSTAV`, og de fem siste tegnene skal være tall eller ?. Ved feil input skal funksjonen be brukeren gjøre et nytt forsøk, inntil input er gyldig. Da skal funksjonen returnere strengen. Eksempel på kjøring:

```
>>> les_gyldig_vitneskilt()
Skriv inn skilt, 2 bokst + 5 tall (?=usikker) VFC1111
Fem siste tegn må være tall eller ?
Skriv inn skilt, 2 bokst + 5 tall (?=usikker) 8V12345
To første tegn må være lovlig skiltbokstav eller ?
Skriv inn skilt, 2 bokst + 5 tall (?=usikker) VF123456
Skiltnummer må være 7 tegn langt
Skriv inn skilt, 2 bokst + 5 tall (?=usikker) V???888
'V???888'
>>>
```


Oppgave 4d (5%)

Skriv en funksjon `match()` som skal sjekke om et vitneobservert skilt kan stemme overens med et faktisk skiltnummer. Funksjonen må ta inn de to strengene som skal sammenlignes som parametere. Returner `True` hvis det er en hel match (strengene er identiske) eller mulig match (de eneste forskjellene skyldes `?`), og `False` hvis de to ikke kan stemme overens. Eksempel på kjøring:

```
>>> match('VF12345', 'VF12355')
False
>>> match('V?1234?', 'VF12355')
False
>>> match('VF???55', 'VF12355')
True
>>> match('??12355', 'VF12355')
True
>>> match('??????', 'VF12355')
True
>>>
```

Oppgave 4 e) (5%)

Skriv en funksjon `match_liste()` som sammenligner ett vitneobservert skilt med en liste av faktiske skilt. Funksjonen skal returnere lista av alle skilt som *kan* stemme med det observerte skiltet. Eksempel på kjøring:

```
>>> match_liste('VF???55', ['VX33322', 'VF12355', 'VF77455', 'DA?????', 'VF10055'])
['VF12355', 'VF77455', 'VF10055']
>>>
```

Oppgave 4 f) (20%)

Anta at vi har en tekstfil `biler.txt` med format som vist i utdraget under, dvs. skiltnummer, bilmerke, modell, farge og navn på eier, hvor hvert element er adskilt med mellomrom.

```
DK21518 FIAT Panda Blå Os,Liss
GH70709 Ford Mondeo Blå Hansen,Jo
FB37769 FIAT Panda Brun Å,Ole
TD79641 Ford S-Max Grå Berg,Jo
PE66975 Toyota Avensis Gul Nes,Al
JV13133 VW Polo Brun Bø,Ole
CG74083 FIAT Panda Blå Hansen,Ann
ZG27056 Toyota Previa Grønn Berg,Bo
```

Skriv et skript eller en `main()`-funksjon som gjør følgende:

- Les inn data fra fila `biler.txt` og putter i en dictionary. Bruk unntaksbehandling for å unngå krasj hvis fila mangler.
- La brukeren sjekke den ene vitneobserverte bilen etter den andre opp mot det som fins i dictionary'en, inntil brukeren ønsker å slutte.
- For hver bil som sjekkes, skriv ut potensielle treff til skjerm, dvs. alle biler hvor de opplysningene som ikke var '?', matchet. Vis på skjerm skiltnummer og navn på eier.
- Hvis ingen kjøretøy matcher, skal programmet skrive ut 'Ingen match'

Du bestemmer selv om du vil skrive all koden for dette i skriptet / `main()`, eller om du vil dele det opp i flere funksjoner, men god oppdeling vil telle positivt der det er naturlig. Likeledes vil det telle positivt om du klarer å bruke funksjoner fra tidligere deloppgaver der det passer. Eksempel på kjøring (hvis fila kun besto av de linjene som var vist i utdraget ovenfor):

```
>>> main()
Fil lest
Hvilket bilmerke var det? FIAT
Hvilken modell? Panda
Hvilken farge? ?
Skriv inn skilt, 2 bokst + 5 tall (?=usikker) ???????
Mulige kjøretøyer er:
FB37769 Eier: Å,Ole
DK21518 Eier: Os,Liss
CG74083 Eier: Hansen,Ann
Vil du sjekke flere kjøretøyer? (J/N) J
Hvilket bilmerke var det? VW
Hvilken modell? Polo
Hvilken farge? Brun
Skriv inn skiltnummer 2 bokstaver + 5 tall (? for usikker) JV33333
Ingen match
Vil du sjekke flere kjøretøyer? (J/N) N
>>>
```

Appendix: Useful Functions and Methods

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 (modulo operator): Divides one number 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 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				