

## Løsningsforslag, Korteeksamen august 2019, TDT4110

### Oppgave 1 - Teori

Hva står random i RAM for?	<b>At alle adresser kan hentes direkte</b>
Hva er det som benyttes i dag som har erstattet vacuum tubes (vakuumrør)?	<b>Transistorer</b>
Hvorfor er det ønskelig å bruke SSD fremfor en vanlig harddisk?	<b>I en SSD lagres data i elektroniske kretser. Det er ingen bevegelige deler, og dermed blir disken raskere og mer pålitelig.</b>
Hva er et operativsystem?	<b>Grunnleggende operasjoner som er nødvendig for å kunne bruke maskinen effektivt, men som ikke er bygd inn i maskinvaren.</b>
Hva er pipelining?	<b>En teknikk der en CPU kan utføre flere instruksjoner parallelt.</b>
Hva skjer om samplingsfrekvensen er for treg?	<b>Lydbølger kan forekomme mellom samplene, og man kan miste viktige segmenter av lyden</b>
Hvilken av følgende komprimeringer er loss-less?	<b>Run-length koding</b>
Hvor mange symboler kan representeres med 6 bit?	<b>64</b>
Hvor mange bytes bruker UNICODE i "worst-case"?	<b>4</b>
Hva er latency	<b>tiden det tar for informasjon å bli laget eller levert.</b>
Hva er unicast	<b>En type adressering der en enkelt datamaskin i nettverket blir identifisert og mottar pakken.</b>
Hva er en protokoll i nettverks-sammenheng?	<b>En protokoll beskriver prosedyrer for feilhåndtering og uventede hendelser.</b>
Hvilken metode av flytkontroll er mest effektiv?	<b>Sliding window</b>

Hva er sant om transportlaget i TCP/IP?	
	<b>Transportlaget sørger for at all data blir levert slik den ble sendt; komplett og i riktig rekkefølge.</b>
Hva er sant om syklisk sjekksum (Cyclic Redundancy Codes)?	
	<b>Syklisk sjekksum brukes i høyhastighetsnettverk fordi den godtar en melding med fast lengde og er ekstremt god til feildetektering.</b>
Hva er phishing	
	<b>Å opptre som en kjent nettside (f.eks. nettbank) for å få tak i personlig informasjon som f.eks. aksesskoder, kontonummer, etc.</b>
Hvordan kan man best sikre at data som kommer fram er identisk med de som ble sendt (sikre dataintegritet)?	
	<b>Bruke hashing</b>
Hvilken hensikt har brannmurer (firewalls)?	
	<b>De er ment til å beskytte nettsteds-integriteten.</b>
Hvordan fungerer kryptering med privat nøkkel (i motsetning til kryptering med offentlig nøkkel)?	
	<b>Partene deler en hemmelig nøkkel som brukes både for kryptering og dekryptering.</b>
Hva står VPN for	
	<b>Virtual private network</b>

## Oppgave 2 - kodeførståelse

- 2a **32**
- 2b **eksamen**
- 2c **4**
- 2d **35**
- 2e **21**
- 2f **tekst = list(tekst)**  
**tekst[6:6] = 'gh'**  
**tekst = ''.join(tekst)**

## Oppgave 3 - Programmering

Løsningene som er vist er bare en av måtene å programmere dette på. Alle programmeringsoppgaver kan løses på mange måter, så dette er bare forslag til kode som fungerer, og ikke fasit i den forstand at det er eneste måte å løse oppgavene på.

### Oppgave 3-1 Lesing fra fil: `read_file(filename)`

```
def read_file(filename):
    fil = open(filename, 'r')
    liste = fil.readlines()
    fil.close()
    return liste

def read_file(filename):
    fil = open(filename, 'r')
    liste = fil.readlines()
    fil.close()
    return liste
```

### Oppgave 3-2 Lag et sett med ingredienser: `fix_ingredients(string)`

```
def fix_ingredients(string):
    return string.strip().split(', ')

def fix_ingredients(streng):
    return streng.strip().split(', ')
```

### Oppgave 3-3: gjør om liste til dictionary: `make_dict(foodlist)`

```
def make_dict(foodlist):
    dikt = {}
    for line in foodlist:
        name, ingredients = line.strip().split(':')
        dikt[name] = fix_ingredients(ingredients)
    return dikt

def make_dict(foodlist):
    dikt = {}
    for line in foodlist:
        name, ingredients = line.strip().split(":")
        dikt[name] = fix_ingredients(ingredients)
    return dikt
```

### Oppgave 3-4: Skriv oppsummering og ingredienser for én spesifikk rett: `print_recipe()`

Her er det brukt try...except.  
Det kan også løses uten bruk  
av dette (se lenger nede)

```
def print_recipe(food_dict,rett):
    try:
        liste = food_dict[rett]
        antall = len(liste)
        print(rett,'has',antall,'ingredients:',',', '.join(liste))
    except:
        print('No dish called',rett)
```

```
def print_recipe(food_dict,rett):
    try:
        liste = food_dict[rett]
        antall = len(liste)
        print(rett,'has',antall,'ingredients:',',', '.join(liste))
    except:
        print('No dish called',rett)
```

```
def print_recipe(food_dict, dish):
    ingredients = food_dict.get(dish,[])
    if ingredients == []:
        print('No dish called',dish)
    # Now we know that there's a recipe:
    else:
        # f-strings will be part of the course from now on.
        print(f"{dish} has {len(ingredients)} ingredients: {'', '.join(ingredients)}")
```

```
def print_recipe(food_dict, dish):
    ingredients = food_dict.get(dish,[]) #verdi settes til [] hvis retten ikke finnes
    if ingredients == []:
        print('No dish called',dish)
    # Now we know that there's a recipe:
    else:
        # f-strings will be part of the course from now on.
        print(f"{dish} has {len(ingredients)} ingredients: {'', '.join(ingredients)}")
```

### Oppgave 3-5: Sorter retter etter ingredienser: `all_recipes_with(food)`

```
def all_recipes_with(food):
    nyDict = {}
    mengde = set()
    verdiListe = food.values()
    for liste in verdiListe:
        mengde = mengde.union(set(liste))
    for x in mengde:
        nyDict[x] = []
        for a in food.keys():
            if x in food[a]:
                nyDict[x] += [a]
    return nyDict
```

Her er det brukt mengder for å ekskludere duplikater i listen. En mengde inneholder ikke duplikater, noe en liste kan gjøre. Alternativ løsning nederst på siden.

```
def all_recipes_with(food):
    nyDict = {}
    mengde = set()
    verdiListe = food.values()
    for liste in verdiListe:
        mengde = mengde.union(set(liste))
    for x in mengde:
        nyDict[x] = []
        for a in food.keys():
            if x in food[a]:
                nyDict[x] += [a]
    return nyDict
```

```
def all_recipes_with(food):
    result_dict = {}
    for dish, ingredients in food.items():
        for ingredient in ingredients:
            # get an empty list on first iteration of ingredient:
            result_dict[ingredient] = result_dict.get(ingredient, [])
            # add dish
            result_dict[ingredient].append(dish)
    return result_dict
```

## Oppgave 6: Velg en tilfeldig matrett med en ingrediens og skriv den ut på en fin måte

Her trenger du ikke skrive noen ny funksjon, men bare sørge for at de funksjonene du har skrevet tidligere blir kalt på riktig måte. Et forslag til løsning:

```
import random
innhold = read_file('food.txt')
food_dict = make_dict(innhold)
recipe_dishes = all_recipes_with(food_dict)
liste = recipe_dishes['egg']
verdi = random.choice(liste)
print("Today you'll be eating",verdi)
print_recipe(food_dict,verdi)
```

Her importeres først random (for å kunne bruke random-funksjoner til å plukke ut en tilfeldig rett). Så hentes `recipe_dishes` for å kunne lage en liste over retter som inneholder 'egg'. Deretter hentes en tilfeldig rett fra denne listen (`verdi = random.choice(liste)`). I den siste linjen kalles så `print_recipe(food_dict,verdi)` for å skrive ut hvor mange ingredienser den valgte retten har.

## Oppgave 7: Beregn bokstavverdien til en matrett: `value_food(food)`

Her har jeg valgt å lage 2 nye funksjoner. Den ene (`value_food(food)`) tar inne en dictionary. Denne går gjennom denne dictionaryen og kaller funksjonen `beregn_verdi(liste)` med verdien til hver nøkkel i dictionaryen for å regne ut summen av bokstavene i hver rett.

```
def beregn_verdi(liste):
    vokaler = 'aeiouyæøå'
    summ = 0
    for element in liste:
        for bokstav in element:
            if bokstav in vokaler:
                summ += 5
            else:
                summ += 1
    return summ

def value_food(food):
    maks = 0
    for key in food.keys():
        verdi = beregn_verdi(food[key])
        if verdi > maks:
            maks = verdi
            mat = key
    print(mat, maks)
```

```
def beregn_verdi(liste):
    vokaler = 'aeiouyæøå'
    summ = 0
    for element in liste:
        for bokstav in element:
            if bokstav in vokaler:
                summ += 5
            else:
                summ += 1
    return summ
```

```
def value_food(food):
    maks = 0
    for key in food.keys():
        verdi = beregn_verdi(food[key])
        if verdi > maks:
            maks = verdi
            mat = key
    print(mat, maks)
```