NTNU
Kunnskap for en bedre verden

Institutt for datateknikk og informasjonsvitenskap

Løsningsforslag Kontinuasjonseksamen i TDT4110 Informasjonsteknologi - grunnkurs

Eksamensdato: **2017-08-XX**

Oppgave 1: Flervalgsoppgave (25%)

Oppgave	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Løsning	c	b	a	c	d	c	a	c	a	d	d	b	c	b	a	a	c	a	d	c

Oppgave 2 Programmering Priskrig (25%)

Oppgave 2a (5%)

Skriv funksjonen `file_to_list` som har en input-parameter `filename`. Denne funksjonen skal lese inn en tekstfil `filename` og returnere en tabell (liste av lister), der hver rekke inneholder navn på butikkjede, navn på vare, og pris på vare. Merk at pris på vare skal representeres som et flyttall (float).

Løsning:

```
def file_to_list(filename):
  dataList=[]
  f = open(filename)
  for line in f:
    lineList = line.split("\t")
    lineList[2] = float(lineList[2])
    dataList.append(lineList)
  f.close()
  return dataList
```

Oppgave 2b (4%)

Skriv funksjonen `list_stores` som har `dataList` som input-parameter. `dataList` er en tabell (liste av lister) lik den som blir returnert fra funksjonen `file_to_list` i oppgave 2a. Funksjonen skal returnere en komplett liste av butikkjeder den finner i tabellen `dataList`. Hver butikkjede skal kun ha ett innslag i lista. Merk også at man aldri vet hvilke butikkjeder som lista vil inneholde. Rekkefølgen på butikkjedene skal samsvare med rekkefølgen de kommer i tabellen `dataList`.

Løsning:

```
def list_stores(dataList):
  storeList = []
  for line in dataList:
    if line[0] not in storeList:
      storeList.append(line[0])
  return storeList
```

Oppgave 2c (5%)
Skriv funksjonen `sum_prices_stores` som har input-parameterne `dataList` og `storeList` (fra oppgave 2a og 2b). Funksjonen skal returnere en liste av totalsummen for alle varene per butikkjede. Rekkefølgen på totalsummene skal være den samme som rekkefølgen på butikkjedene i `storeList`.

Løsning:

```
def sum_prices_stores(dataList,storeList):
  sumStores = [0]*len(storeList)
  for line in dataList:
    storeNr = storeList.index(line[0])
    sumStores[storeNr] += line[2]
  return sumStores
```

Oppgave 2d (6%)
Skriv funksjonen `rank_stores` som har input-parameterne `storeList` og `sumStores` (fra oppgave 2b og 2c). Funksjonen skal returnere ei liste med navnene til butikkjedene sortert fra kjeden med lavest pris til høyest pris.

Løsning:

```
def rank_stores(storeList,sumStores):
  switch = True
  while(switch):
    switch = False
    for i in range(len(storeList)-1):
      if sumStores[i+1]<sumStores[i]:
        switch = True
        temp = sumStores[i]
        sumStores[i] = sumStores[i+1]
        sumStores[i+1] = temp
        temp = storeList[i]
        storeList[i] = storeList[i+1]
        storeList[i+1] = temp
  return storeList
```

Oppgave 2e (5%)
Skriv funksjonen `store_analysis` som har input-parameteren `filename`. Funksjonen skal laste inn ei fil med filnavnet `filename`, og deretter skrive ut summen for varene for hver butikk, og deretter skrive ut ranking av butikkjeder sortert etter der varene i fila `filename` er billigst. Funksjonen skal ikke returnere noe, men ha en utskrift som vist under.

Løsning:

```
def store_analysis(filename):
  dataList = file_to_list(filename)
  storeList = list_stores(dataList)
  sumStores = sum_prices_stores(dataList,storeList)
  print("The total price for shopping per store is:")
  for i in range(len(storeList)):
    print(storeList[i],":",sumStores[i],"kr")

  print("\nThe ranking of stores according to prices is:")
  rankedStores = rank_stores(storeList,sumStores)
  for i in range(len(rankedStores)):
    print(i+1,rankedStores[i])
```

Oppgave 3 Programmering Storskjerm (30%)

Her er kode for funksjonen som kan simulere framvisning på storskjerm (ikke en del av løsningen):

```
def show_display(content):
    print()
    if len(content)==6 and len(content[0])==30:
        print("####################################")
        print("#                                  #")
        for row in content:
            print('#  '+row.upper()+"  #")
        print("#                                  #")
        print("####################################")
    else:
        print("Error: Wrong dimensions")
```

Oppgave 3a (4%)
Skriv funksjonen `enter_line` som har to input-parametere `prompt` og `length`. Funksjonen skal spørre brukeren om å skrive inn en setning som skal returneres som en tekststreng. Setningen skal være av lengde spesifisert av input-parameteren `length`. Hvis setningen ikke er av spesifisert lengde, skal funksjonen gi feilmeldingen: ”The text must be `[length]` characters long”, og fortsette å spørre om en ny setning til brukeren har gitt en med korrekt lengde. Parameteren `prompt` spesifiserer hva brukeren skal spørres om.

Løsning:

```
def enter_line(prompt,length):
    text=""
    while(len(text)!=length):
        text = input(prompt)
        if len(text)!=length:
            print("The text must be",length,"characters long")
    return text
```

Oppgave 3b (4%)
Skriv funksjonen `adjust_string` som har to input-parametere `text` og `length`. Funksjonen skal returnere en ny utgave av tekststrengen `text` som har lengde `length`. Hvis strengen `text` har flere tegn enn `length`, skal den resterende teksten kuttes. Hvis strengen `text` har færre tegn enn `length`, skal teksten midtstilles og man skal legge til mellomrom (space) slik at lengden på strengen som returneres blir akkurat `length`.

Løsning:

```
def adjust_string(text,length):
    l = len(text)
    if l<length:
        text = " "*((length-l)//2)+text
        text+= " "*(length-len(text))
    elif l>length:
        text = text[:length]
    return text
```

Oppgave 3c (3%)
Skriv en smartere versjon av versjon av funksjonen `enter_line_smart` (fra oppgave 3a) som har to input-parametere `prompt` og `length`. Funksjonen skal ta imot input fra brukeren ved å bruke spørreteksten `prompt`, og returnere en streng på lengde `length`. Hvis teksten brukeren skriver inn er lengre enn `length` skal resterende teksten kuttes, og hvis teksten brukeren skriver inn er kortere skal teksten midtstilles og fylles ut med mellomrom (space) slik at teksten blir på `length` antall tegn.

Løsning:
```
def enter_line_smart(prompt,length):
    text = input(prompt)
    text = adjust_string(text,length)
    return text
```

Oppgave 3d (4%)
Skriv funksjonen `enter_show_text` som spør brukeren om å legge inn seks linjer med tekst på 30 tegn, og deretter viser innholdet på storskjermen. Funksjonen har ingen input-parametere og returnerer ingen ting. Hvis teksten som brukeren skriver inn er over 30 tegn, skal overflødig tekst kuttes bort. Hvis teksten som brukeren skriver inn er under 30 tegn, skal teksten midtstilles og fylles ut med mellomrom (space) slik at teksten blir på 30 tegn.

Løsning:
```
def enter_show_text():
    content = []
    for i in range(6):
        text = enter_line_smart('Line '+str(i+1)+": ",30)
        content.append(text)
    show_display(content)
```

Oppgave 3e (5%)
Skriv funksjonen `scroll_display` som har to input-parametere `content` og `line`. Funksjonen returnerer ingen ting. Parameteren `content` er ei liste bestående av 6 tekststrenger på 30 tegn, og parameteren `line` er et heltall mellom 1 og 6. Funksjonen skal vise fram innholdet fra lista `content` på storskjermen, der teksten på linje `line` skal roteres mot venstre (scrolle) en helt til teksten på denne linja er tilbake der den startet (som vist på figurene nederst). Oppdatering av storskjermen skal skje hvert tiendedels sekund (0,1 sek). Teksten på linje `line` vil altså forflytte seg 30 ganger mot venstre før funksjonen avslutter. Du kan anta at funksjonen kalles med riktige argumenter (`content` inneholder 6 strenger på 30 tegn og `line` er heltall mellom 1 og 6). Tidsforsinkelse gjøres ved å bruke funksjonen `sleep(s)` fra biblioteket `time`, der `s` spesifiserer antall sekunder tidsforsinkelse (kan også bruke desimaltall for `s`).

Løsning:
```
def scroll_display(content, line):
    import time
    text = content[line-1] # Pick out the string to be scrolled
    for i in range(len(text)):
        text = text[1:]+text[0] # Text from index 1->end + text from index 0
        content[line-1]=text # insert scrolled text into context
        show_display(content) # show text on large display
        time.sleep(0.1) # wait for 0.1 seconds
```

Oppgave 3f (10%)
Skriv funksjonen `display_from_file` som har en input-parameter `filename`. Funksjonen skal lese inn fra tekstfila `filename`, og vise fram innholdet i tekstfila på storskjermen seks linjer av gangen. Funksjonen returnerer ingen ting. Hvis teksten på ei linje i fila er over 30 tegn, skal resterende tekst kuttes. Hvis teksten på ei linje i fila er under 30 tegn, skal teksten midtstilles og fylles ut med mellomrom (space). Funksjonen skal ha 10 sekunders pause mellom hver gang nytt innhold vises på skjermen. Man kan anta at fila har et antall linjer som går opp i seks-gangen.

Løsning:

```
def display_from_file(filename):
    import time
    f = open(filename)
    x = 0
    content = [] # Storing data for large display
    for line in f:
        line = adjust_string(line.strip(),30)
        content.append(line)
        x+=1
        if x==6:
            show_display(content)
            time.sleep(10)
            x=0
            content=[]
    f.close()
```

Oppgave 4 Kodeforståelse (20%)

Oppgave 4a) 5%
Svar: `'God middag!!'`
Svar: Kombinerer tegn fra tre tekststrenger

Oppgave 4b) 5%
Svar:

```
m=[[0,0,0,0,0],
    [0,3,4,5,0],
    [0,4,5,6,0],
    [0,5,6,7,0],
    [0,0,0,0,0]]
```

Svar: Legger 0 rundt kanten på tabellen

Oppgave 4c) 5%
Svar: `'The Matrix'`
Svar: Reverserer strengen og tar med annethvert tegn.

Oppgave 4d) 5%
Svar: 256
Svar: Multipliserer argument 1 med seg selv rekursivt antall ganger fra argument 2 til argument 3 (ikke til og med).

Tablå riktig svar flervalgsoppgave

Kandidatnummer: _____ Program: _____

Fagkode: _____ Dato: _____

Antall sider: _____ Side: _____

Oppgavenr	***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	■	■		■

Tablå feil svar flervalgsoppgave

Kandidatnummer: _____ Program: _____

Fagkode: _____ Dato: _____

Antall sider: _____ Side: _____

Oppgavenr	***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			■	