

Norges teknisk-naturvitenskapelige
universitet
Institutt for datateknikk og
informasjonsvitenskap

TDT4110 IT Grunnkurs
Høst 2012

Auditorieøving 2

Navn:

Linje:

Brukernavn:

Oppgavesettet inneholder 5 oppgaver. For flervalgsoppgavene er det kun ett riktig svaralternativ. Skriv svarene rett på oppgavearket.

1 Teori

a) Informasjon som beskriver informasjon kalles _____

1. ekstra-informasjon
2. meta-data
3. spesial-data
4. formatering

b) RGB-verdien for helt blå er (0 = av, 1 = på) _____

1. 0000 0000 0000 0000 0000 0000
2. 1111 1111 0000 0000 0000 0000
3. 0000 0000 1111 1111 0000 0000
4. 0000 0000 0000 0000 1111 1111

c) Analog informasjon er _____

1. diskret
2. kontinuerlig
3. tilfeldig
4. digital

d) En digital-til-analog-konverterer (DAC) _____

1. endrer digital informasjon til analoge bølger
2. konverterer kontinuerlig lyd til digital lyd
3. konverterer lyd til et elektrisk signal
4. setter tilnærmede verdier

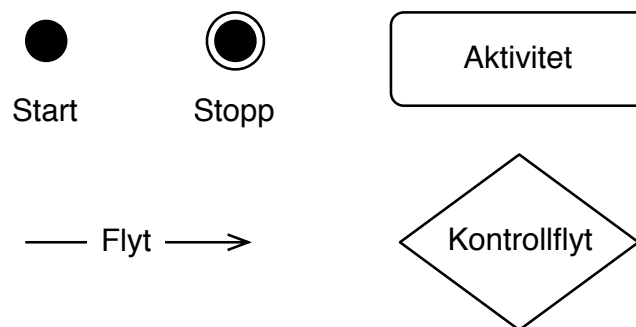
e) Hvor mange forskjellige bitmønster kan man lagre i et register av lengde K bits?

1. K^2
2. 2^k
3. K
4. 2

Deloppgave	a	b	c	d	e
Svar					

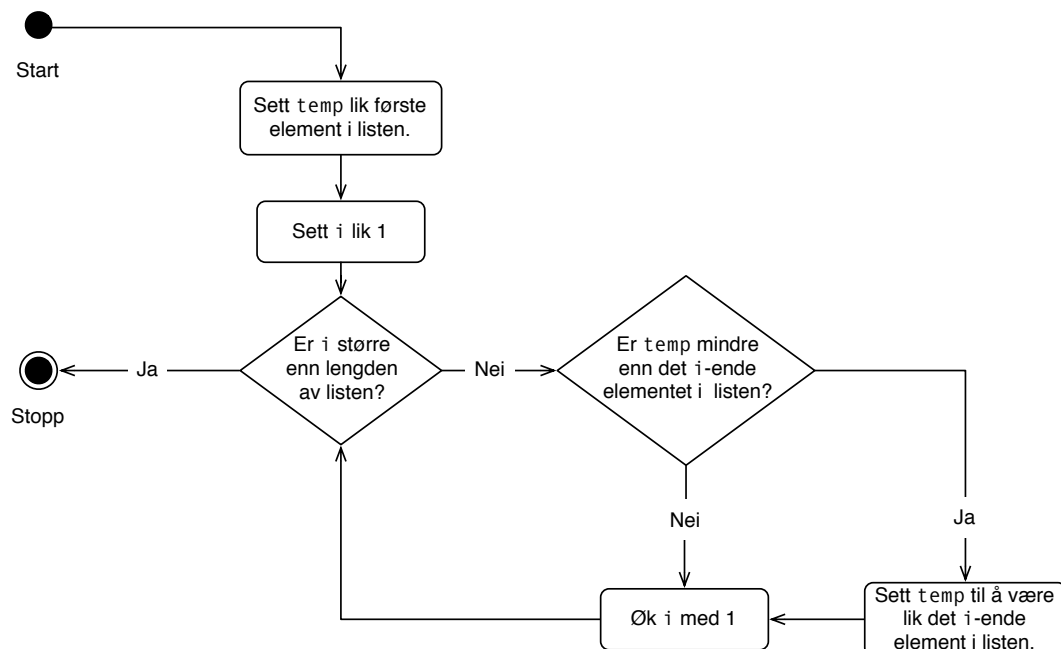
2 Flytskjema

Figur 1 beskriver syntaksen for flytskjemaer brukt i denne oppgaven.



Figur 1: Template for flytskjema

Gitt følgende flytskjema hvor `i` og `temp` er variabler.



Figur 2: Mystisk flytskjema

Tips: Funksjonen tar inn en liste som parameter.

a) Hva gjør funksjonen?

b) Skriv funksjonen.

c) Følgende kode definerer funksjonen `filter`.

Kodesnutt 1

```
def filter(li, func):  
    result = []  
    i = 0  
    while i < len(li):  
        if func(li[i]):  
            result.append(li[i])  
            i += 1  
  
    return result
```

Tegn et flytskjema for funksjonen.

3 Kattelister



Tenk deg at du ønsker å flytte et antall katter ved hjelp av en vannbøtte. Problemer er at du eier veldig mange katter og enda flere vannbøtter, men kan likevel bare bruke én bøtte for å unngå at noen katter begynner å fryse.

- a) Du har en rekke bøtter av forskjellig størrelser, samt et antall katter. Enheten til størrelsen er antall katter som får plass i bøtten. Lag en funksjon `first_index` som tar inn disse størrelsene i form av en liste, samt et tall som sier hvor mange katter du har. Funksjonen skal returnere indeksen til den første bøtten som er stor nok til å romme antallet katter. Du kan anta at det gitte antallet katter alltid vil få plass i minst én av bøttene.

Eksempel 1

```
>>> first_index([3, 6, 1, 5, 99, 4, 2, 3, 7, 9], 4)
1
```

- b) Du ønsker nå å returnere indeksen til den minste bøtta som er stor nok til å romme alle kattene dine. Skriv en funksjon `min_index` som gjør dette. Bruk gjerne funksjonen fra deloppgave a) om nødvendig.

Eksempel 2

```
>>> min_index([3, 6, 1, 5, 99, 4, 2, 3, 7, 9], 4)
5
```

4 Kodeforståelse

- a) Hva skrives ut når vi kjører koden under?

Kodesnutt 2

```
def mystic_function(input):
    for i in range(len(input)):
        tall = i
        for j in range(i + 1, len(input)):
            if input[j] < input[tall]:
                tall = j
        input[i], input[tall] = input[tall], input[i]

a = [12, 5, 4, 7, 8, 18, 2, -9, 17, 6]
mystic_function(a)
print(a)
```

b) Hva skrives ut når vi kjører koden under?

Kodesnutt 3

```
def a(b, c, d, e):  
    if d < e:  
        return a(c, b + c, d + 1, e)  
    else:  
        return c  
  
print(a(0, 1, 0, 3))
```

5 Kjøretid

For å representere en veistrekning, heretter kalt *strekning*, fra ett sted til et annet kan vi bruke en liste som inneholder to lister. Den første “interne” listen inneholder lengdene av delstrekningene i kilometer, den andre listen inneholder fartsgrensen for hver delstrekning.

En vei som starter med 3 km 60-sone og deretter 24 km 80-sone, osv. kan representeres slik:

```
lengder = [3, 24, 5, 10, 6, 31, 12, 7, 8, 21]  
fartsgrenser = [60, 80, 60, 80, 90, 70, 60, 30, 50, 60]  
veistrekning = [lengder, fartsgrenser]
```

- a) Lag funksjonen `is_valid` som tar en veistrekning `path` som parameter. Den skal gå gjennom `path` og sjekke at det ikke er noen “ugyldige” delstrekninger – dvs. delstrekninger med lengde eller fartsgrense mindre eller lik null. Hvis den finner en slik strekning skal funksjonen returnere **False**, ellers skal den returnere **True**.

Eksempel 3

```
>>> is_valid([[103, 45, 10, 24], [30, 60, 80, 90]])  
True  
>>> is_valid([[5, 11, 45, -13], [30, 0, 80, 90]])  
False
```

- b) Lag funksjonen `drive_time` som tar inn en veistrekning `path` som parameter. Funksjonen skal regne ut hvor lang kjøretiden for strekningen er. Hvis strekningen er ugyldig skal funksjonen returnere `False`.

Tips: Kjøretiden for en delstrekning kan regnes ut ved å dele lengden på fartsgrensen.

Eksempel 4

```
>>> drive_time([[103, 45, 10, 24], [30, 60, 80, 90]])  
4.575
```

- c) Lag funksjonen `shortest_valid` som tar inn to veistrekninger som parameter, `p1` og `p2`. Funksjonen skal returnere kjøretiden til den strekningen med kortest kjøretid som også er gyldig. Hvis ingen av rutene er gyldige skal funksjonen returnere `False`.

Tips: Bruk funksjonen fra deloppgave b).