

**Eksamen INF-1100**  
**Innføring i programmering og**  
**datamaskiners virkemåte**  
**Vår 2012**

*Eksamenssettet består av 4 oppgaver.*

Der oppgaven ber om at du skriver en funksjon kan du bruke C lignende pseudo-kode. Husk også at du kan referere tilbake til funksjoner du tidligere har definert.

**Oppgave 1 - 20%**

(a) Oversett følgende tall fra desimal til binær representasjon:

- 34
- 17

(b) Adder sammen følgende binære tall og oversett resultatet til desimal representasjon.

- $00001010 + 00100101$
- $00101010 + 00000011$

(c) Skisser i pseudo-kode hvordan du oversetter et tall fra desimal til binær representasjon.

(d) Skisser i pseudo-kode hvordan du oversetter et tall fra binær til desimal representasjon.

## Oppgave 2 - 25%

De fleste av dagens datamaskiner er strukturert i henhold til en modell foreslått av John von Neumann i 1946. Beskriv komponentene i denne modellen og kommunikasjonen mellom disse (maksimum 2 sider).

### Oppgave 3 - 30%

Gitt et array  $A$  som inneholder  $n$  heltall.

- (a) Skisser i pseudo-kode en funksjon som finner tallet i  $A$  med størst verdi.
- (b) Skisser i pseudo-kode en funksjon som beregner *medianen* av tallene i  $A$ . En algoritme du kan bruke er å først sortere alle tallene i  $A$ . Deretter kan medianen beregnes som følger: dersom  $n$  er et oddetall er medianen det midterste elementet. Dersom  $n$  er et partall, er medianen gjennomsnittet av de to midterste tallene. Du kan anta at det eksisterer en funksjon for å sortere tallene i  $A$ .
- (c) Anta at tallene i  $A$  representerer høyden til personer i en bestemt aldersgruppe. Skisser i pseudo-kode en funksjon som finner et intervall som omfavner 95% av tallene i  $A$  (95% av alle tallene i  $A$  må ha en verdi som er innenfor dette intervallet). Hint: sorter tallene i  $A$  først.

## Oppgave 4 - 25%

Begge oppgavene nedenfor involverer lister med et ukjent antall elementer. Elementene har en assosiert nøkkel og du kan anta at det eksisterer en funksjon *sammenlign* som tar to elementer som argument, sammenligner elementenes nøkler og returnerer en verdi som indikerer forholdet mellom nøklene:

```
int sammenlign(void *e1, void *e2)
```

*sammenlign* vil returnere  $-1$  dersom nøkkel til  $e1$  har mindre verdi enn nøkkel til  $e2$ ,  $0$  dersom nøkkel til  $e1$  og nøkkel til  $e2$  har lik verdi og  $1$  dersom nøkkel til  $e1$  har større verdi enn nøkkel til  $e2$ .

- (a) Gitt en liste  $a$ . Skisser i pseudo-kode en funksjon som finner ut om det er to elementer i  $a$  med nøkler som har lik verdi.
- (b) Gitt to lister  $a$  og  $b$ . I begge listene har et element i posisjon  $i$  en nøkkel med mindre eller lik verdi sammenlignet med elementet i posisjon  $i + 1$ . Elementene i listene er med andre ord i sortert rekkefølge (lav til høy). Skisser i pseudo-kode en funksjon som tar  $a$  og  $b$  som argument og returnerer en ny liste med elementene fra  $a$  og  $b$  i sortert rekkefølge (lav til høy).

Du kan anta at følgende listefunksjoner er tilgjengelige:

```
// Lag en ny liste
list_t *list_create(void);

// Fjern og returner første element i en liste
void *list_removefirst(list_t *list);

// Sett inn et element først i en liste
int list_addfirst(list_t *list, void *item);

// Lag en ny listeiterator som peker på første element i listen
list_iterator_t *list_createiterator(list_t *list);

// Returner element som pekes på av iterator og
// la iterator peke på neste element.  NULL
// returneres når en når slutten på listen.
void *list_next(list_iterator_t *iter);

// Frigi iterator
void list_destroyiterator(list_iterator_t *iter);
```