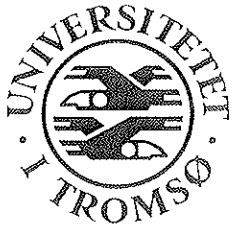

EKSAMENSOPPGAVE I INF-1100

Eksamen i	: INF-1100 Innføring i programmering
Dato	: 29. Februar 2008
Tid	: 09:00 — 13:00
Sted	: Aud.max.
Tillatte hjelpemidler	: Ingen

Oppgavesettet er på fire (4) sider inkl. forside

Kontaktperson
Åge Kvalnes, 48011111

DET MATEMATISK-NATURVITENSKAPELIGE FAKULTET

Universitetet i Tromsø, N-9037 Tromsø, Telefon 77 64 40 01, Telefaks 77 64 47 65

Eksamen INF-1100
Innføring i programmering
Vår 2008

Eksamenssettet består av 3 oppgaver.

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

Oppgave 1 - 30%

a) Forklar kort hva som ligger i begrepet instruksjonssettarkitektur (*instruction set architecture (ISA)*).

b) Beskriv kort fasene i en instruksjonssyklus ('instruction cycle') i en datamaskin strukturert i henhold til Von Neumann modellen.

c) Forklar hvordan I/O utføres i en Von Neumann basert datamaskin.

Oppgave 2 - 35%

a) To sirkler overlapper dersom sirkelbuene skjærer hverandre eller dersom den ene sirkelen omslutter den andre totalt. Gitt en datastruktur som spesifiserer midtpunkt og radius til en sirkel:

```
typedef struct sirkel sirkel_t;
struct sirkel {
    float x, y;
    float radius;
};
```

Skriv en funksjon *SirkelOverlapp* som avgjør om to sirkler a og b overlapper hverandre:

```
int SirkelOverlapp(sirkel_t *a, sirkel_t *b)
```

Funksjonen skal returnere 1 dersom sirklene overlapper og 0 dersom sirklene ikke overlapper. Hint: Sjekk avstanden mellom sirklenes midtpunkter. For å beregne avstanden mellom to punkter x1,y1 og x2,y2 kan du benytte Pythagoras' formel:

$$avstand = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$$

b) Gitt en datastruktur som spesifiserer nedre venstre hjørne og høyde og bredde til et rektangel:

```
typedef struct rektangel rektangel_t;
struct rektangel {
    float x, y;
    float høyde, bredde;
};
```

Skriv en funksjon *RektangelOverlapp* som avgjør om to rektangel a og b overlapper hverandre:

```
int RektangelOverlapp(rektangel_t *a, rektangel_t *b)
```

Funksjonen skal returnere 1 dersom rektanglene overlapper og 0 dersom rektanglene ikke overlapper. Hint: Dersom to rektangel ikke overlapper vil det ene rektangelet i sin helhet enten ligge til høyre, venstre, over, eller under det andre rektangelet.

Oppgave 3 - 35%

a) Skriv en funksjon *zenosum*:

```
double zenosum(int n)
```

Funksjonen skal beregne og returnere summen opp til n ledd av rekken:

$$\frac{1}{2^1} + \frac{1}{2^2} + \frac{1}{2^3} + \ldots + \frac{1}{2^n}$$

b) Skriv en funksjon *uniketall* som tar som inn-parametre en peker til et integer array og en angivelse av størrelsen på arrayet:

```
int uniketall(int *array, int arraysize)
```

Funksjonen skal returnere antall unike tall i *array*. Funksjonen kan ikke modifisere tallene i *array*. Denne oppgaven kan løses på flere måter. Dersom du velger å benytte lister kan du anta at følgende listefunksjoner er tilgjengelige:

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

// Frigi liste. Elementer i listen blir ikke frigitt
void list_destroy(list_t *list);

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

// Lag en ny listeiterator
list_iterator_t *list_createiterator(list_t *list);

// Frigi listeiterator
void list_destroyiterator(list_iterator_t *iter);

// Returner element som pekes på av iterator og la iterator peke på neste element
void *list_next(list_iterator_t *iter);

// La en iterator peke på første element i listen
void list_resetiterator(list_iterator_t *iter);
```