



**Eksamen
Bokmål**

**INF-1100
INNFØRING I PROGRAMMERING**

Eksamen : Inf-1100, Innføring i programmering

Dato : Torsdag 23. februar 2010

Tid : 0900-1300

Sted : Administrasjonsbygget, rom B154

Denne oppgaveteksten er på 3 sider inklusiv forside

Kontakt: Anders Andersen, 951 80 675

Fakultet for naturvitenskap og teknologi

Universitetet i Tromsø, 9037 Tromsø, Tlf. 77 64 40 01, Fax 77 64 47 65

Les hele oppgaveteksten nøye før du starter. Disponer tiden slik at du rekker å svare på alle spørsmålene. Vær nøye på å svare på det som det blir spurt om, og bruk fullstendige setninger.

Oppgave 1 – 30%

- Forklar kort hva som ligger i begrepet instruksjonssettarkitektur (*instruction set architecture (ISA)*).
- Beskriv kort fasene i en instruksjonssyklus ('instruction cycle') i en datamaskin strukturert i henhold til Von Neumann modellen.
- Forklar hvordan I/O utføres i en Von Neumann basert datamaskin.

Oppgave 2 – 35%

To sirkler overlapper dersom sirkelbuene skjærer hverandre eller dersom den ene sirkelen omslutter den andre totalt. Gitt en datastruktur som spesifiserer midtpunkt x, y og diameter b til en sirkel (h skal vi ignorere nå, men den vil alltid ha verdien 0 for en sirkel):

```
typedef struct legeme legeme_t;
struct legeme {
    int x, y;
    int b, h;
};
```

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

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

Funksjonen skal returnere 1 dersom sirklene overlapper og 0 dersom sirklene ikke overlapper. Hint: Sjekk avstanden mellom sirklens midtpunkter. For å beregne avstanden a mellom to punkter x_1, y_1 og x_2, y_2 kan du benytte Pythagoras' formel:

$$a = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Datastrukturen `legeme_t` over kan også brukes for å spesifiserer nedre venstre hjørne x, y , og bredde b og høyde h til et rektangel.

- b) Skriv en funksjon `RektangelOverlapp` som avgjør om to rektangel a og b overlapper hverandre:

```
int RektangelOverlapp(legeme_t *a, legeme_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.

- c) Skriv funksjonen `Overlapp` som sjekker om to legemer overlapper hverandre:

```
int Overlapp(legeme_t *a, legeme_t *b);
```

Funksjonene skal returnere 1 dersom legmene overlapper og 0 dersom legemene ikke overlapper. Et legeme kan enten være en sirkel eller et rektangel. Funksjonen `Overlapp` skal implementeres ved å blant annet benytte (kalle) funksjonene `SirkelOverlapp` og `RektangelOverlapp`. Du må spesielt ta hensyn til tilfellet hvor et legeme er sirkel og det andre er rektangel.

Oppgave 3 – 35%

Lag et enkelt listebibliotek `liste.c`. Biblioteket skal kunne brukes til å lage nye (tomme) lister, legge elementer inn først og sist i en liste, ta ut første og siste element i en liste, søke og ta ut et spesifikt element i listen, og slette en liste og alle dets elementer. Du må også spesifisere de nødvendige datastrukturene (`liste_t` og `node_t`). Feilsituasjoner skal håndteres av biblioteket. For eksempel så skal feil indikeres hvis man forsøker å ta ut et element fra en tom liste. Dette kan gjøres ved å returnere en tom peker (`nil`). Her er `liste.h` uten de nødvendige datastrukturene:

```
liste_t *NyListe();
int LeggInnFørst(liste_t *liste, node_t *node);
int LeggInnSist(liste_t *liste, node_t *node);
node_t *TaUtFørst(liste_t *liste);
node_t *TaUtSist(liste_t *liste);
node_t *TaUtElement(liste_t *liste, node_t *node);
int SlettListe(liste_t *liste);
```