Eksamen
Bokmål

INF-1100
INNFØRING I PROGRAMMERING

Eksamen : Inf-1100, Innføring i programmering

Dato : Torsdag 17. desember 2009

Tid : 0900-1300

Sted : Åsgårdveien 9

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. Der oppgaven ber om at du skriver en funksjon kan du bruke C lignende pseudo-kode.

Oppgave 1 - 30%

De fleste av dagens datamaskiner er organisert etter Von Neumann modellen. Beskriv denne modellen. Beskrivelsen bør omfatte de ulike komponentene i modellen, hvordan disse interagerer med hverandre, og hva som skjer når instruksjonene i et program utføres.

Oppgave 2 - 50%

Du er ansatt i et spillutviklingsfirma og du skal være med å utvikle spillet *Stjernekamp*. I første utgave av spillet vil romskip være representert som et kvadrat, og astroider som en sirkel. Både romskip og astroider er representert ved denne datastrukturen:

```
typedef struct legeme legeme_t;
struct legeme {
  int posx, posy;
  int bredde;
  int liv;
};
```

Posisjon på skjermen angis med henholdsvis x- og y-koordinat `posx` og `posy`. Dette er sentrum av romskipet eller astroiden. Størrelse er angitt som `bredde`, og for et romskip vil bredden være det samme som høyden. For en astroide er `bredde` det samme som diameter.

a) Skriv funksjonene `KvadratKollisjon` og `SirkelKollisjon` som henholdvis sjekker om to romskip har kollidert (overlapper hverandre) eller om to astroider har kollidert (overlapper hvernadre):

```
int KvadratKollisjon(legeme_t *a, legeme_t *b);
int SirkelKollisjon(legeme_t *a, legeme_t *b);
```

Funksjonene skal returnere 1 hvis vi har kollisjon og 0 hvis vi ikke har kollisjon. Hint: For kvadrat så har vi overlapping hvis forskjellen mellom både x- og y-koordinat er mindre enn kvadratenes gjennomsnittlige bredde. For to sirkler kan du sjekke avstanden mellom sirkelens midtpunkter. For å beregne avstanden mellom punktet x_1, y_1 og punktet x_2, y_2 så kan du benytte Pythagoras formel (d er avstanden mellom punktene):

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

Astroider har evig liv, eller for å si det på en annen måte, de blir ikke skadet i en kollisjon. For astroider setter vi derfor telleren `liv` til -1, og vi endrer aldri denne. For romskip angir telleren antall liv vi har igjen. For hver kollisjon telles denne ned med 1. Når verdien blir 0 så skal romskipet slettes. Det betyr at vi kan bruke verdien `liv` for å sjekke om vi har et romskip eller en astroide. Vi skal ved hjelp av funksjonene `KvadratKollisjon` og `SirkelKollisjon` nå lage funksjonen `Kollisjon`, som sjekker om to vilkårlige legemer har kollidert (altså enten to romskip, eller to astroider, eller en astroide og et romskip).

b) Skriv funksjonen `Kollisjon` som sjekker om to legemer har kollidert (overlapper hverandre):

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

Funksjonene skal returnere 1 hvis vi har kollisjon og 0 hvis vi ikke har kollisjon. Vi tillater følgende forenkling: For å sjekke om et romskip og en astroide kolliderer så kan vi først late som om de begge er sirkler, og sjekke om de kolliderer da. Hvis det ikke er tilfelle så kan vi sjekke avstanden fra alle hjørnene av kvadratet til sirkelens midtpunkt. Hvis ingen av disse er mindre enn sirkelens radius så kan vi anta at vi *ikke* har kollisjon.

I *Stjernekamp* er alle legemer (romskip og astroider) samlet i en lenket liste. Du skal lage funksjonen `RenskOpp` som går gjennom denne listen og fjerner alle forekomster av romskip som har null liv igjen (telleren `liv` har verdien 0).

c) Definer de datastrukturene du trenger for den dynamiske lista med alle legemer (romskip og astroider).

d) Skriv funksjonen `RenskOpp` for å fjerne forekomster av romskip med null liv igjen. Funksjonen skal returnere antall romskip som er igjen etter at romskip som har null liv igjen er fjernet.

Oppgave 3 – 20%

Rekken av Fibonaccitall $(0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, \ldots)$ er definert slik (for $n > 0$):

$$\begin{aligned} F(1) &= 0 \\ F(2) &= 1 \\ F(n) &= F(n-2) + F(n-1) \end{aligned}$$

Det vil si at det første tallet i rekken er 0. Det andre er 1, og deretter er hvert tall definert som summen av de to foregående tallene.

Lag en rekursiv funksjon som returnerer det n'te Fibonaccitallet:

```
int fib(int n);
```