

EKSAMENSOPPGAVE

Eksamen i:	INF-1100 Innføring i programmering og datamaskiners virkemåte
Dato:	06.12.2016
Klokkeslett:	09:00 – 13:00
Sted:	Teorifagb, hus 3, 3.203 og 3.218 og Adm.bygget, Aud.max og B.154
Tillatte hjelpemidler:	Ingen
Type innføringsark (rute/linje):	
Antall sider inkl. forside:	6
Kontaktperson under eksamen:	Robert Pettersen
Telefon/mobil:	47 24 75 52

NB! Det er ikke tillatt å levere inn kladdepapir som del av eksamensbesvarelsen. Hvis det likevel leveres inn, vil kladdepapiret bli holdt tilbake og ikke bli sendt til sensur.

Eksamen INF-1100
Innføring i programmering og
datamaskiners virkemåte
Høst 2016

Eksamenssettet består av 4 oppgaver.

Les oppgaveteksten grundig og disponer tiden slik at du får tid til å svare på alle oppgavene. I noen oppgaver kan det være nødvendig å tolke oppgaveteksten ved å gjøre noen antagelser - gjør i så fall rede for hvilke antagelser du har gjort, men pass på å ikke gjøre antagelser som trivialisere oppgaven.

Der du skal utvikle eller beskrive en algoritme anbefales det at du først beskriver algoritmen på et høyt abstraksjonsnivå, f.eks. med figurer, før du går videre med detaljer og eventuell pseudokode. Dersom det spørres etter en implementasjon i C kreves det ikke 100% syntaktisk korrekt kode. Dersom det spørres etter pseudokode kan du også skrive ren C-kode om du ønsker.

Husk også at du kan referere tilbake til funksjoner du tidligere har definert.

Oppgave 1 - 25%

Gi en kort beskrivelse av von Neumann modellen og instruksjonssyklusen. Beskrivelsen bør omfatte de ulike komponentene i modellen og hvordan disse interagerer med hverandre.

Oppgave 2 - 25%

Gitt følgende program:

```
#include <stdio.h>

int main()
{
    int x = 128;
    char *y = "128";
    int z[] = { 1, 2, 8 };

    if (y[0] == z[0]) {
        printf("%d\n", x + z[1]);
    }
    else {
        printf("%d\n", x / z[1]);
    }
    return 0;
}
```

- a) Forklar hva som er forskjellen på variablene x, y, og z, med spesielt fokus på hvilken type variablene har og hvordan verdiene deres er representert i datamaskinens minne.
- b) Hvilket tall vil skrives ut på skjermen når programmet kjøres? Forklar hvordan du kommer frem til svaret ditt.

Oppgave 3 - 25%

Gitt et array A som inneholder n flyttall (float eller double i C).

- a) Skriv en funksjon som beregner gjennomsnittet av tallene i A med følgende signatur:

```
double gjennomsnitt(double *A, int n);
```

Gjennomsnitt er summen av alle tallene delt på antall tall, og kan beregnes med følgende formel:

$$\frac{1}{n} \sum_{i=0}^{n-1} x_i$$

- x_i er tallet i posisjon i i array A .

For eksempel, gitt tallene 3 og 5 beregnes gjennomsnittet slik: $(3+5)/2 = 4$.

- b) Skriv en funksjon som beregner standardavviket for tallene i A , med følgende signatur:

```
double standardavvik(double *A, int n)
```

Standardavvik er summen av kvadratene til hvert av tallene minus gjennomsnittet delt på antall tall, og kan beregnes med følgende formel:

$$\sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (x_i - \bar{x})^2}$$

- x_i er tallet i posisjon i i array A .

- \bar{x} er gjennomsnittet av alle tallene i A .

- For å beregne kvadratroten kan du anta at det eksisterer en funksjon `double sqrt(double n)`.

For eksempel, gitt tallene 3 og 5 som har gjennomsnitt 4, beregnes standardavviket slik: $\frac{\sqrt{(3-4)^2 + (5-4)^2}}{2}$

- c) Skriv en funksjon som flytter rundt på tallene i A slik at de blir liggende i stigende rekkefølge. Funksjonen skal ha følgende signatur:

```
void sorter(float *A, int n);
```

Oppgave 4 - 25%

Denne oppgaven involverer bruk av lister og et angitt sett med listefunksjoner. Bruk de angitte listefunksjonene i besvarelsen. **Ikke** gjør antagelser om hvordan listene er implementert.

Gitt en datastruktur som spesifiserer et koordinat:

```
typedef struct koordinat koordinat_t;
struct koordinat {
    int x;
    int y;
};
```

Koordinater er organisert i lister som hver representerer en geometrisk figur (trekant, firkant, osv).

- a) Skriv en funksjon som tar en liste med koordinater og en enkelt koordinat som argument. Funksjonen skal addere den enkle koordinaten til alle koordinatene i listen.
- b) Skriv en funksjon som tar en liste med koordinater som argument, og tegner linjer mellom koordinatene i listen. For eksempel, hvis listen inneholder 3 koordinater skal funksjonen tegne en trekant, og hvis listen inneholder 4 koordinater skal funksjonen tegne en firkant.
- c) Skriv en funksjon som tar en liste med koordinater som argument og returnerer en ny liste med koordinater til det minste rektangelet (boundingbox) som kan omslutte figuren beskrevet i argument listen.

Du kan anta at følgende funksjoner er tilgjengelige:

```
// Tegn en linje mellom punktet (x1, y1) og (x2, y2)
void drawLine(int x1, int y1, int x2, int y2);

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

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

// Lag en ny liste-iterator som peker på første element i listen
list_iterator_t *list_createiterator(list_t *list);
```

```
// Returnerer elementet som pekes på av iteratoren og
// lar iteratoren peke på neste element.
// NULL returneres når en kommer til slutten av listen
void *list_next(list_iterator_t *iterator);

// Frigir iteratoren
void list_destroyiterator(list_iterator_t *iterator);
```

EKSAMENSOPPGÅVE

Eksamen i:	INF-1100 Innføring i programmering og datamaskiner sin virkemåte
Dato:	06.12.2016
Klokkeslett:	09:00 – 13:00
Stad:	Teorifagb, hus 3, 3.203 og 3.218 og Adm.bygget, Aud.max og B.154
Lovlege hjelpemiddel:	Ingen
Type innføringsark (rute/linje):	
Antall sider inkl. forside:	6
Kontaktperson under eksamen:	Robert Pettersen
Telefon/mobil:	47 24 75 52

NB! Det er ikkje lov å levere inn kladd saman med svaret. Om det likevel leverast inn, vil kladden bli heldt tilbake og ikkje sendt til sensur.

Eksamen INF-1100
Innføring i programmering og
datamaskiner sin virkemåte
Haut 2016

Eksamenssettet består av 4 oppgåver.

Les oppgåveteksten grundig og disponer tida slik at du får tid til å svara på alle oppgåvene. I nokre oppgåver kan det vera naudsynt å tolka oppgåveteksten ved å gjera nokre føresetnader (antakelser) - gjer i så fall greie for kva slags føresetnader du har gjort, men pass på å ikkje gjera føresetnader som trivialiserer oppgåva.

Der du skal utvikla eller skildra ein algoritme tilrådest det at du først skildrar algoritmen på eit høgt abstraksjonsnivå, t.d. med figurar, før du går vidare med detaljar og eventuell pseudokode. Dersom det vert spurt etter ein implementasjon i C krev det ikkje 100% syntaktisk korrekt kode. Dersom det vert spurt etter pseudokode kan du òg skriva rein C-kode om du ynskjer.

Husk òg at du kan referera attende til funksjonar du tidlegare har definert.

Oppgåve 1 - 25%

Gje ei kort skildring av von Neumann modellen og instruksjonssyklusen. Skildringa bør omfatta dei ulike komponentane i modellen og korleis desse interagerer med kvarandre.

Oppgåve 2 - 25%

Gjeve følgjande program:

```
#include <stdio.h>

int main()
{
    int x = 128;
    char *y = "128";
    int z[] = { 1, 2, 8 };

    if (y[0] == z[0]) {
        printf("%d\n", x + z[1]);
    }
    else {
        printf("%d\n", x / z[1]);
    }
    return 0;
}
```

- a) Forklar kva som er skilnaden på variablane x, y, og z, med spesielt fokus på kva for ein type variablane har og korleis verdiane deira er representert i minnet til datamaskina.
- b) Kva for eit tal vil skrivast ut på skjermen når programmet vert køyrt? Forklar korleis du kjem fram til svaret ditt.

Oppgåve 3 - 25%

Gjeve eit array A som inneheld n flyttal (float eller double i C).

- a) Skriv ein funksjon som bereknar gjennomsnittet av tala i A med følgjande signatur:

```
double gjennomsnitt(double *A, int n);
```

Gjennomsnitt er summen av alle tala delt på mengd tal, og kan bereknast med følgjande formel:

$$\frac{1}{n} \sum_{i=0}^{n-1} x_i$$

- x_i er talet i posisjon i i array A .

Til dømes, gjeve tala 3 og 5 vert gjennomsnittet berekna slik: $(3+5)/2 = 4$.

- b) Skriv ein funksjon som bereknar standardavviket for tala i A , med følgjande signatur:

```
double standardavvik(double *A, int n)
```

Standardavvik er summen av kvadrata til kvart av tala minus gjennomsnittet delt på mengd tal, og kan bereknast med følgjande formel:

$$\sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (x_i - \bar{x})^2}$$

- x_i er talet i posisjon i i array A .

- \bar{x} er gjennomsnittet av alle tala i A .

- For å berekna kvadratroten kan du anta at det eksisterer ein funksjon `double sqrt(double n)`.

Til dømes, gjeve tala 3 og 5 som har gjennomsnitt 4, vert standardavviket berekna slik: $\frac{\sqrt{(3-4)^2 + (5-4)^2}}{2}$

- c) Skriv ein funksjon som flyttar rundt på tala i A slik at dei vert liggjande i stigande rekkjefølgje. Funksjonen skal ha følgjande signatur:

```
void sorter(float *A, int n);
```

Oppgåve 4 - 25%

Denne oppgåva involverer bruk av lister og eit angeve sett med listefunksjonar. Bruk dei angjevne listefunksjonane i besvarelsen. **Ikkje** gjer antagelser om korleis listene er implementert.

Gjeve ein datastruktur som spesifiserer ein koordinat:

```
typedef struct koordinat koordinat_t;
struct koordinat {
    int x;
    int y;
};
```

Koordinatar er organisert i lister som kvar representerer ein geometrisk figur (trekant, firkant, osv).

- a) Skriv ein funksjon som tek ei liste med koordinatar og ein enkelt koordinat som argument. Funksjonen skal addera den enkle koordinaten til alle koordinatane i lista.
- b) Skriv ein funksjon som tek ei liste med koordinatar som argument, og teiknar linjer mellom koordinatane. Til dømes, viss lista inneheld 3 koordinatar skal funksjonen teikna ein trekant, og viss lista inneheld 4 koordinatar skal funksjonen teikna ein firkant.
- c) Skriv ein funksjon som tek ei liste med koordinatar som argument og returnerer ei ny liste med koordinatar til det minste rektangelet (boundingbox) som kan omgje figuren skildra i argument lista.

Du kan anta at følgjande funksjonar er tilgjengelege:

```
// Teikn ei linje mellom punktet (x1, y1) og (x2, y2)
void drawLine(int x1, int y1, int x2, int y2);

// Lag ei ny liste
list_t *list_create(void);

// Set inn eitt element først i ei liste
int list_addfirst(list_t *list, void *element);

// Lag ei ny liste-iterator som peikar på første element i lista
list_iterator_t *list_createiterator(list_t *list);
```

```
// Returnerer elementet som vert peikt på av iteratoren og
// lèt iteratoren peika på neste element.
// NULL vert returnert når ein kjem til slutten av lista
void *list_next(list_iterator_t *iterator);

// Frigjev iteratoren
void list_destroyiterator(list_iterator_t *iterator);
```