UiT Norges arktiske universitet

Kort hjemmeeksamen i:	INF-1100
Dato:	2022-03-01
Tidspunkt:	Kl 09:00 - 13:00 + 15 minutter til innlevering
Kursansvarlig:	Edvard Pedersen Telefon: 40458598
Antall sider:	3
Support:	Du kan ringe 776 20 880 for support på eksamensdagen.
Vekting av spørsmål, eller annen informasjon:	Vekting er spesifisert per oppgave.
Viktig informasjon om sitering og plagiering:	1. Dette er en individuell eksamen som skal besvares uten samarbeid med andre. 2. Alle hjelpemidler er tillatt (egne notater, pdfer fra forelesningene, lærebok, internett etc). 3. Alle eksamener som leveres i WISEflow blir automatisk sjekket for plagiat. Det er ikke tillatt å kopiere medstudenter, nettressurser, kilder, eller litteratur uten referanser.

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 trivialiserer 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. Der du blir spurt om å implementere noe, så kan du skrive i C-lignende pseudokode, skrivefeil eller syntaxfeil trekker ikke ned. 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 funksjon:

```
char *ukjent(char *a, int b, int c)
{
    char *d = (char *)calloc((c - b + 1), sizeof(char));
    strncpy(d, a + b, c-b);
    return d;
}
```

a) Hvilken verdi vil funksjonen *ukjent* ovenfor returnere ved følgende kall:

   (a) *ukjent("Du store verden", 2, 7)*

   (b) *ukjent("Hallo", 4, 5)*

b) Beskriv forskjellen mellom variabelen *a* og det funksjonen returnerer (*d*).

c) Hva betyr tegnet * på de forskjellige plassene det er i brukt i koden over?

Oppgave 3 - 25%

Gitt et array *A* av *n* heltall (les: *int *tall*).

a) Skriv en funksjon *sum* som regner ut summen av alle tallene.

```
int sum(int *A, int n) // A er tallene, n er antall tall
```

b) Skriv en funksjon *overmiddels* som henter posisjonen til alle tall som er over gjennomsnittet, og returnerer disse i ett nytt array.

```
int *overmiddels(int *A, int n)
```

Oppgave 4 - 25%

I denne oppgaven skal vi håndtere lister med ukjent antall elementer. Det eksisterer en funksjon sammenlign som tar to elementer som argument og returnerer en verdi som indikerer forholdet mellom dem:

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

Funksjonen returnerer verdien -1 dersom e1 er mindre enn e2, 0 dersom e1 er lik e2, og 1 dersom e1 er større enn e2.

a) Gitt to lister a og b, begge sortert og med minste element først. Lag funksjonen merge som tar a og b som argument og returnerer en ny liste med elementene fra a og b sortert i rekkefølge, fra minst til størst:

```
list_t *merge(list_t *a, list_t *b);
```

b) Gitt to lister a og b, begge sortert og med minste element først i listen, lag en funksjon mergenum som tar a og b som argument og returnerer antall unike elementer.

```
int mergenum(list_t *a, list_t *b);
```

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 sist i listen
void list_addlast(list_t *list, void *item);

// 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);
```