

EKSAMENSOPPGAVE

Eksamen i: INF-1100 Innføring i programmering og datamaskiners
virkemåte

Dato: Mandag 23. februar 2015

Tid: Kl 09:00 – 13:00

Sted: Administrasjonsbygget, B154

Tillatte hjelpemidler: Ingen

Oppgavesettet er på 4 sider inklusiv forside

Kontaktperson under eksamen: Steffen Viken Valvåg

Telefon: 98117749

NB! Det er ikke tillatt å levere inn kladd sammen med besvarelsen

Eksamenssettet består av 4 oppgaver.

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

Oppgave 1 – 25%

Gi en kort beskrivelse av von Neumann 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 – 20%

Gitt følgende program:

```
#include <stdio.h>

int main(int argc, char **argv)
{
    int x = 42;
    char *y = "42";
    int z[] = { 4, 2 };

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

- 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.
- Hvilket tall vil skrives ut på skjermen når programmet kjøres? Forklar hvordan du kommer frem til svaret ditt.

Oppgave 3 – 30%

- a) Oversett følgende tall fra desimal til binær representasjon:
- 29
 - 50
- b) Adder sammen følgende binære tall og oversett resultatet til desimal representasjon:
- $00001010 + 00100101$
 - $00101010 + 00000011$
- c) Skriv et program som «dekomponerer» et tall og skriver det ut som en sum av 2-er potenser. Tallet som skal dekomponeres skal oppgis som et ekstra argument på kommandolinja. Bruk funksjonen `atoi` for å konvertere fra tekststrenger til heltall:

```
int atoi(char *s);
```

(Om du ikke vet hvordan man bruker argumenter fra kommandolinja kan du komme videre ved å anta at du allerede har tallet lagret i en variabel.)

Eksempel: dersom programmet invokeres fra kommandolinja slik:

```
./dekomponer 429
```

Skal følgende utskrift produseres:

$$1 + 4 + 8 + 32 + 128 + 256 = 429$$

Oppgave 4 – 25%

I et terningspill er en sekvens av n terningkast representert som et array av heltall, hvor hvert element i arrayet har en verdi mellom 1 og 6 som tilsvarer antall øyne på terningen i det aktuelle kastet.

a) Implementer en funksjon

```
int yatzee(int *A, int n);
```

som skal sjekke om alle de n terningkastene i A er like. Hvis alle er like, skal funksjonen returnere 1, og ellers skal den returnere 0.

b) Implementer en funksjon

```
void frekvens(int *A, int n, int *B);
```

som teller antall forekomster av de ulike verdiene i A og fyller inn resultatet i arrayet B slik at $B[1]$ er antall enere, $B[2]$ er antall toere, $B[3]$ er antall treere, osv. Anta at B i utgangspunktet inneholder verdien 0 for alle elementer.

c) Funksjonen `frekvens` er også brukt på noe kryptisk vis i denne funksjonen:

```
int ukjent(int *A, int n)
{
    int B[7] = { 0 };
    int *C = calloc(n+1, sizeof(int)); // Array av n+1 tall
    int x;

    frekvens(A, n, B);
    frekvens(B, 7, C);
    x = C[n];
    free(C);
    return x;
}
```

Hva er det funksjonen `ukjent` gjør (hvilket formål har den)? Forklar hvordan funksjonen virker.