

Eksamen INF-1100

Innføring i programmering

Vår 2009

Eksamenssettet består av 5 oppgaver.

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

Oppgave 1 - 20%

De fleste av dagens datamaskiner er strukturert i henhold til en modell foreslått av John Von Neumann i 1946. Beskriv denne modellen. Beskrivelsen bør vektlegge hvordan programmer utføres.

Oppgave 2 - 20%

Skriv en funksjon *triangel*:

```
void triangel(int n)
```

Funksjonen skal skrive ut et triangelmønster bestående av n linjer. For eksempel, med $n = 3$ skal funksjonen skrive ut følgende mønster:

```
  *
 ***
*****
```

Du kan anta at argumentet til funksjonen er et tall større enn 0. Merk at første linje starter med 2 mellomrom (spaces) og andre linje starter med 1 mellomrom.

Løsningsforslag 2:

```
void triangel(int n)
{
    int i;
    int lineno;

    for(lineno = 0; lineno < n; lineno++) {
        // Mellomrom før stjernene
        for(i = 0; i < n-lineno-1; i++)
            printf(" ");

        // Stjernene
        for(i = 0; i < 1 + 2*lineno; i++)
            printf("*");

        printf("\n");
    }
}
```

Oppgave 3 - 30%

Skriv en funksjon *søk*:

```
int søk(int *a, int len, int key)
```

Funksjonen tar som argumenter en peker til et array av heltall (*a*), en angivelse av lengden på *a* (*len*), samt et heltall *key*. Funksjonen skal returnere verdien 1 dersom heltallet *key* forekommer i *a* og 0 hvis ikke.

Løsningsforslag 3:

```
int søk(int *a, int len, int key)
{
    int i;

    for (i = 0; i < len; i++) {
        if (key == a[i])
            return 1;
    }

    return 0;
}
```

Oppgave 4 - 20%

Skriv en funksjon *ersortert*:

```
int ersortert(int *a, int len)
```

Funksjonen tar som argumenter en peker til et array av heltall (*a*) og en angivelse av lengden på *a* (*len*). Funksjonen skal returnere 1 dersom tallene i *a* er i sortert rekkefølge, og 0 hvis ikke. Merk at det kan forekomme flere heltall med samme verdi.

Løsningsforslag 4:

```
int ersortert(int *a, int len)
{
    int i;

    for (i = 1; i < len; i++) {
        if (a[i] < a[i - 1])
            return 0;
    }

    return 1;
}
```

Oppgave 5 - 20%

Et *palindrom* er et ord eller en setning som er det samme enten det leses fra høyre eller venstre. Eksempler inkluderer (dersom man overser mellomrom):

radar
renner
rolf are vurderer om arons ni drag i gardinsnora morer edru vera flor

Skriv en funksjon *erpalindrom*:

```
int erpalindrom(char *s)
```

Funksjonen tar som argument en peker til en null-terminert tekststreng *s*. Funksjonen skal returnere 1 dersom *s* er et palindrom, og 0 hvis ikke.

Løsningsforslag 5:

```
int erpalindrom(char *s)
{
    int v, h;

    // Finn null-terminering
    for(h=0; s[h] != '\0'; h++) ;

    // Siste bokstav er på plassen før null-terminering
    h--;

    // Gå gjennom halve arrayet
    for(v=0; v < h; v++, h--) {
        // Flytt venstre indeks til neste bokstav
        while(s[v] == ' ')
            v++;

        // Flytt høyre indeks til neste bokstav
        while(s[h] == ' ')
            h--;

        // Hvis de to bokstavene ikke er like er det ikke et palindrom
        if(s[v] != s[h])
            return 0;
    }

    // Hvis vi er kommet hit har vi et palindrom
    return 1;
}
```