

**Algoritme 1.7** Program for oppgave 1.4–1 n, m : hvor langt løkkene skal gå

```

1 void oppgave1(int n, int m) {
2   for (int i = 1; i < n; ++i) {
3     for (int j = 0; j < m; ++j) {
4       ...
5     }
6   }
7 }

```

```

1 void oppgave1(int n, int m) {
2   for (int i = 1; i < n; ++i) {
3     for (int j = 0; j < m; ++j) {
4       ...
5     }
6   }
7 }

```

**Algoritme 1.8** Program for oppgave 1.4–2 t : tabell n : tabellengde/2 x : sum av tabellelementene

```

1 void oppgave2(int n, int *t) {
2   int i, x = 0;
3   for (i = 0; i < 2*n; ++i) x += t[i];
4   for (i = n; i > 1; --i) t[i] -= x;
5 }

```

```

1 void oppgave2(int n, int []t) {
2   int x = 0, i;
3   for (i = 0; i < 2*n; ++i) x += t[i];
4   for (i = n; i > 1; --i) t[i] -= x;
5 }

```

Engelske uttrykk**asymptotisk notasjon** asymptotic notation**kompleksitet** complexity**løkke** loop**Oppgaver til kapittel 1**

Vi har ei tallrekke som angir forandring i kurs for en aksje fra dag til dag. Vi ønsker å finne hvordan vi kunne fått best fortjeneste, dvs. maksimal positiv differanse mellom kjøpspris og salgspris. Kjøp må selvsagt skje før salg.

Se tabell 1.1 på motstående side for et eksempel på en slik tallrekke.

Det meste vi kan tjene på ett kjøp fulgt av salg, er 5. Det får vi til ved å kjøpe etter kursfallet på dag 3 og selge igjen etter oppgangen på dag 7.

1–1 Lag og implementer en algoritme som finner hvilket kjøps- og salgstidspunkt som lønner seg best. Sjekk at algoritmen virker.

**Algoritme 1.9** Program for oppgave 1.4–3

```

1 void oppgave3(int n, int *t) {
2   oppgave1(n, n);
3   for (int i = 0; i < n; ++i) {
4     oppgave2(i, t);
5   }
6 }

```

```

1 void oppgave3(int n, int []t) {
2   oppgave1(n, n);
3   for (int i = 0; i < n; ++i) {
4     oppgave2(i, t);
5   }
6 }

```

Tabell 1.1: Eksempel på kursforandringer

Dag nr.	1	2	3	4	5	6	7	8	9
Kursforandring	-1	+3	-9	+2	+2	-1	+2	-1	-5

1–2 Analyser algoritmen. Finn kompleksiteten for tidsforbruket. Bruk Θ -notasjon om mulig, eller i hvert fall O -notasjon.

1–3 Legg inn tidsmåling i programmet, og prøvekjør med tallrekker av ulik lengde. Se om tidsmålingene stemmer med analysen i oppgave 1–2.

Vær oppmerksom på at klokka i datamaskinen sannsynligvis har for lav oppløsning til å ta tiden på behandling av korte tallrekker. Dette kan avhjelpest enten ved å bruke lengre tallrekker eller ved å behandle mange små tallrekker og dele den totale tiden med antall rekker. Vær også oppmerksom på at for lange tallrekker gir problemer med tidsmålinger, de kan fylle opp minnet slik at operativsystemet skriver ut data til disk. Det tar lang tid og fører til målinger som ikke er representative for algoritmen.

