

# Forslag til løsninger til

## Skriftlig eksamen i Programmering

KVL, 10. januar 1998

NB: Disse forslag til løsninger er ikke blevet tjekket på maskine. Løsningerne kan altså indeholde såvel syntaksfejl som andre fejl. Peter Sestoft 1999-12-13.

### Opgave 1

#### Opgave 1.1

Programmet udskriver tallet 3. Det er længden af den længste ubrudte gentagelse i  $a[0..(antal-1)]$ .

#### Opgave 1.2

Følgende rettelser skal foretages:

- En ny variabel erklæres efter `int tmp = ...:`

```
int tal = tmp;
```

- Den betingede ordre `if (sek > maxsek) ...` rettes til:

```
if (sek > maxsek) {
    maxsek = sek;
    tal = tmp;
}
```

- Ordren `return maxsek` rettes til:

```
return tal;
```

#### Opgave 1.3

Metoden `main` udskiftes med følgende:

```
public static void main(String[] args) {
    Reader inp = new FileReader("talfil.txt");
    StreamTokenizer tstream = new StreamTokenizer(inp);
    tstream.parseNumbers();
    int[] a = new int[1000];
    int n = 0;
    tstream.nextToken();
    while (tstream.ttype != StreamTokenizer.TT_EOF) {
        a[n] = (int)(tstream.nval);
        n = n + 1;
    }
    System.out.println(maxsekvens(a, n));
}
```

Opgaveteksten siger ikke noget om hvor lange talsekvenser programmet skal kunne håndtere. I denne løsning går jeg ud fra at 1000 tal er nok.

Man kan håndtere vilkårlig lange talsekvenser (forudsat de kan være i maskinens lager) ved at fordoble størrelsen på tabellen `a` hver gang den viser sig at være for lille. [Den løsning vil selvfølgelig give ekstra point, men det var ikke den opgavestilleren tænkte på].

## Opgave 2

### Opgave 2.1

Her vises indholdet af tabellen `arr` til at begynde med og efter hver ombytning:

0	1	2	3	4	5	Kommentar
23	25	18	60	50	44	
23	18	25	60	50	44	Ombyttede (1, 2)
23	18	25	50	60	44	Ombyttede (3, 4)
23	18	25	50	44	60	Ombyttede (4, 5)
18	23	25	50	44	60	Ombyttede (0, 1)
18	23	25	44	50	60	Ombyttede (3, 4)

### Opgave 2.2

Når man sorterer en tabel med  $n = 6$  tal, så udføres den ydre løkke med  $i$  lig med 5, 4, 3, 2, 1. For hvert gennemløb af den ydre løkke udføres den indre løkke krop  $i$  gange. Det samlede antal udførelser af den indre løkke krop er derfor:

$$5 + 4 + 3 + 2 + 1 = 15$$

Generelt, når man sorterer en tabel med  $n$  tal, så udføres den indre løkke krop

$$(n - 1) + (n - 2) + \dots + 2 + 1 = \frac{n \cdot (n - 1)}{2}$$

gange.

Antal udførelser af den indre løkke krop bestemmer programmets køretid. Da den indre løkke krop altid udføres lige mange gange, er bedste og værste køretid ens, og størrelsesordenen er  $n^2$ .

### Opgave 2.3

Boblesortering ligner udvalgsortering, idet begge har bedste og værste køretid af størrelsesorden  $n^2$  for sortering af  $n$  elementer.

## Opgave 3

### Opgave 3.1

Programmet udskriver:

```
8
3
6.28
```

### Opgave 3.2

```
class Trug extends Beholder {
    double bredde, laengde;

    Trug(double bredde, double laengde) {
        this.bredde = bredde;
        this.laengde = laengde;
    }

    public double rumfang() {
        return laengde * 3.14 * bredde * bredde * 2;
    }
}
```

### Opgave 3.3

```
static double totalrumfang(Beholder[] b) {
    double rumf = 0;
    for (int i=0; i < b.length; i++)
        rumf += b.rumfang();
    return rumf;
}
```

### Opgave 3.4

I den abstrakte superklasse Beholder erklærer man:

```
public void fyldpaa(double maengde) {
    double res = indhold + maengde;
    indhold = Math.min(res, rumfang());
}
```