

ASSIGNMENT (AFLEVERINGSOPGAVE) 5

GENERAL INFORMATION

This assignment is made public on Friday, March 5th, 1 PM. The assignment is due on

Friday, March 12th, 1PM.

Hand in your assignment to the teaching assistant running your lab session.

The first page of your (written) assignment has to contain at least the following information:

- the course name (Grundlæggende Programmering)
- name and student number of the fellow student(s) in your group (max two)
- assignment number

Please staple your assignment!

You will get back the graded assignment one week after submission deadline.

WARM-UP AND SUGGESTIONS FOR FURTHER EXERCISES

As a warm-up I suggest the review questions from Chapter 5 on Arrays (pp. 182–184 of my copy of the textbook).

If you want to do further programming exercises I suggest the following ones from the book (pp. 184–188):

- Exercises 1–5 are simple about simple manipulations of arrays and form a good warm-up.
- Exercises 6–9 are about character arrays and manipulations of those.
- Exercises 10–12 deal with character arrays and sorting them lexicographically.
- Exercises 14–16 deal with an implementation of polynomials.
- Exercises 17–20 are about sorting various types of arrays.
- Exercises 21 and 22 are about the well-known *birthday problem* and ask for a simple simulation of it: If there are more than 23 people in a room one can bet that at least two of them have their birthday on the same day of the year. Doing this one will (on average) win, because the probability of this event is just above 50% (assuming that the birthdays are uniformly distributed over the 365 different days of the year).

It is up to you whether or not you want to work on these exercises. Those exercises will not be marked.

ØVELSER

Arbejd i små grupper.

- Arrays: Opgaver 3, 4 og 5 fra kapitel 5.
- Arrays: Opgaver 10, 11, 12 fra kapitel 5.

ASSIGNMENT: SEARCHING AND SORTING

AFLEVERINGSOPGAVE: AT SØGE OG AT SORTERE

1. Brug metoden `binsearch()` til at finde key 28 i arrayet

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
17	20	28	33	35	39	39	51	52	57	62	69	74	84	84	86	88	95	97

Vis værdierne af `l`, `r`, `c` og `found` på linjerne `pp1`, `pp2` og `pp3` (koden findes også på hjemmesiden).

```
static boolean binsearch(int[] arr, int key){
    /* Binary search for key in an ordered array arr.
    */

    boolean found = false;
    int l = 0;           // left marker
    int r = arr.length-1; // right marker
    int c;              // centre

    // pp1

    while(!found && l <= r){
        c = (l+r)/2;
        if(key < arr[c])
            { // search in lower part
                r = c-1;
            }
        else if(arr[c] < key)
            { // search in upper part
                l = c+1;
            }
        else
            found = true;

        // line pp2

    }
    // pp3

    return found;
}
```

2. Forklar hvorfor metoden giver det korrekte svar hvis arrayet er tomt, dvs. hvis `arr.length` er nul.

3. Her findes koden af metoder `selSort()` og `swap`:

```
public static void selSort(int[] arr){
    int length = arr.length;
    for(int i = 0; i<length-1; i++){
        int index = i;
        for(int j = i+1; j < length; j++){
            if(arr[j]<arr[index])
                index = j;
        }
        // pp1
        swap(arr,i,index);
        // pp2
    }
}

public static void swap(int[] arr, int i, int j){
    int tmp = arr[i];
    arr[i] = arr[j];
    arr[j] = tmp;
}
```

Forklar med dine egen ord algoritmen som benyttes til at sortere arrayet. Hvis du har problemer med at forstå algoritmen så skal du først læse afsnit 5.5. i tekstbogen.

4. Brug metoden `selSort()` til at sortere arrayet

0	1	2	3	4	5	6	7	8	9	10	11	12
95	17	39	20	52	28	86	35	39	51	88	57	84

manuelt.

Vis værdierne af `i` og `index` på linjerne `pp1`, og hvis det hele array på linjerne `pp2` (koden findes også på hjemmesiden).