Grading Criteria C-Lab 2, Group B, Spring 2018

Your TAs:

- **Alexandru Sasu**
- **Taha Zia**

Code formatting

- Indentation is a way to organize the document, make sure you indent your code. This will make your work easier to read/understand/modify/maintain/debug by anyone who sees your code or even by you after a long time.

Deduction for NOT following the criterion	What do we expect from you
-5%	There are many styles of indenting your code (in C the most common ones are K&R and Allman). It is your choice which one to use, however we ask that all your programs will be indented only in the one chosen way
-5%	Name the variables according to their meaning (e.g., `numberOfCars` and not `x`); you can use `i`, `j`, `k` for iterator variables
-5%	Use at most 80 characters per line to improve readability of your code in JGrader
-5%	Use only the English alphabet (i.e., stick to ASCII characters)

Comments

Deduction for NOT following the criterion	What do we look for:
-5%	Comment out the important parts of your code, be as precise as possible and avoid commenting every line
	Comments should describe your code: no questions, suggestions, or complaints should be included
	Put in the beginning of your program a block of comments (look for example below)

Example of the header block for your program:

```
/*
   320112
   problem_1.1.c
   Student Name
   j.email@jacobs-university.de
*/
```

Good example:

```
int cars[100], length = 0;

    for (int i = 0; i < length - 1; i++)
        for (int j = 0; j < length; j++)
            if (cars[i] > cars[j])
                // sort array of cars in ascending order
                swap(cars[i], cars[j]);
    return 0;
}
```

Bad example:

```
// array of cars of size 100, variable length of 0 value
int cars[100], length = 0;
// for loop to go from 0 to length - 1
for (int i = 0; i < length - 1; i++)
    // another for loop which goes from 0 to length
    for (int j = 0; j < length; j++)
        // compare cars[i] and cars[j]
        if (cars[i] > cars[j])
            // swap cars[i] and cars[j]
            swap(cars[i], cars[j]);
return 0;
```

Compiling

- The code should compile with no errors or warning messages. **Please note that there will be test cases provided in Grader that your program needs to pass.** Otherwise the TA's will use the following table for point deduction:

Max points subtracted	Criterion
-10%	Program compiles with warnings
-50%	Program does not compile
-30%	Segmentation fault
-20%	Program doesn't pass provided test cases

Correctness and completeness

Deduction for NOT meeting the expectation	What do we expect from you
-5%	Check for null pointers
-20%	Allocate and deallocate memory correctly
-35%	Do not make logical mistakes in functions and be sure your functions behave as they are meant to
-35%	Follow the guidelines for the given problem and make sure you meet all requirements
-20%	Check whether your program works for all cases (in addition to the test cases provided, think about possible edge cases)

We would like to see your own work and assess your own effort in completing the tasks for the course. If we find two completely identical or very similar codes we will penalized students with 100 % deduction for the task.