

Assignment 6 - Arrays, Recursion and Files

- The problems of this assignment must be solved in C.
- The TAs are grading solutions to the problems according to the following criteria:
<http://minds.jacobs-university.de/sites/default/files/uploads/teaching/CProgrammingFall17/Grading-Criteria-c.pdf>

Problem 6.1 *Rectangle chars* (1 point)

Presence assignment, due by 18:30 h today

Write a function that takes three arguments: a character `ch` and two integers `r` and `c`. The function should print the character `ch` `r` x `c` times in the form of `r` rows and `c` columns.

Write a simple program that reads the appropriate variables and prints the result on the screen by calling the function.

You can safely assume that the input will be valid.

Problem 6.2 *Triple I* (1 point)

Presence assignment, due by 18:30 h today

Write a function `void triple(int arr[], int size)` that multiplies all elements of an initialized array by three. Your program should print the elements of the array before and after the multiplication. Test your program with an array that contains the hard coded values 3, 2, 9, 3, 7, 12.

Problem 6.3 *Determine uppercase characters* (1 point)

Write a function `int count_upper(char* str)` that counts and returns the number of uppercase characters within a string. Then write a program where you repeatedly read a string from the keyboard and determine and print the number of uppercase characters in that string. If you provide an empty string (the string will just contain `'\n'`), then the program should stop its execution. You must use a pointer to walk through the string.

You can assume that the string will be not longer than 100 characters.

Problem 6.4 *Triple II* (1 point)

Modify your program for **Problem 6.2** such that you first read a number `n`, and then the elements of an array with `n` components. Therefore you will need to dynamically allocate your array. Then triple the elements using your previous `triple()` function and print the results on the screen.

Do not forget to release the allocated memory when not needed anymore.

You can safely assume that the input will be valid.

Problem 6.5 *Computing a vector's norm* (1 point)

Write a program that reads from the keyboard a number `n`, and then a vector `v` of double values with `n` components (you are free to do this in any way you like, for example, read `n` lines, with one number on each). Write a function that computes the norm of this vector. The norm is defined as:

$$|v| = \sqrt{\sum_{i=1}^n v_i^2}$$

Use the function to compute the norm of the vector you read. From the `main` function print the value of the norm on the screen. Additionally, determine and print on the screen the smallest and largest components of the vector, and the positions in the vector where they occur.

You can safely assume that the input will be valid.

Problem 6.6 *Sum of natural numbers between two limits* (1 point)

Write a program which reads two positive integers n and m from the keyboard. Also write a recursive function for computing the sum of the natural numbers between n and m (including both of them). Print the resulting value on the screen within the `main` function.

You can safely assume that the input will be valid.

Problem 6.7 *Greatest common divisor of two numbers* (1 point)

Write a program which reads two positive integers a and b from the keyboard. Also write a recursive function for determining the gcd (greatest common divisor) of a and b using Euclid's algorithm. According to this algorithm if one of the numbers is divisible by the other then the second one is the gcd. If this is not the case then the gcd of the second number and the remainder of a/b has to be determined.

You can safely assume that the input will be valid.

Problem 6.8 *Read a char and write an int* (1 point)

Write a program which reads the first character from the file called `char.txt` and writes its ASCII code value as number into the file called `code.txt`. Use an editor to create the input file `char.txt`. Your program is responsible to create the output file `code.txt`.

You can safely assume that the content of the input file will be valid.

Problem 6.9 *Read and write integers* (1 point)

Write a program which reads from the keyboard the names of two files containing two integer numbers. Your program should read these two values from the two files, compute their sum and write the result into a file `sum.txt`.

You can safely assume that the two input files are correct if existing.

Bonus Problem 6.10 *Copy file* (1 point)

Write a program which reads the content of the file called `text.txt` line by line and writes them into another file called `text_copy.txt`.

You can safely assume that the content of the input file will be valid if existing.

How to submit your solutions

- Your source code should be properly indented and compile with `gcc` without any warnings (You can use `gcc -Wall -o program program.c`). Insert suitable comments (not on every line ...) to explain what your program does.
- Please name the programs according to the suggested filenames (they should match the description of the problem) in Grader.

Each program **must** include a comment on the top like the following:

```
/*
    JTSK-320111
    a6_p1.c
    Firstname Lastname
    myemail@jacobs-university.de
*/
```

- You have to submit your solutions via Grader at <https://grader.eecs.jacobs-university.de>.
If there are problems (but **only** then) you can submit the programs by sending mail to x.he@jacobs-university.de **with a subject line that begins with JTSK-320111**.
It is important that you do begin your subject with the coursenummer, otherwise I might have problems to identify your submission.
- Please note, that after the deadline it will not be possible to submit any solutions. It is useless to send late solutions by mail, because they will not be accepted.

This assignment is due by Friday, October 6th, 10:00 h.