Programming in C I, Group B
Jacobs University Bremen
Dr. Kinga Lipskoch
Xu He

Course: JTSK-320111
September 28th, 2018

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: https://grader.eecs.jacobs-university.de/courses/320111/2018_2gB/Grading-Criteria-C.pdf

Problem 6.1 *Triangle chars* (1 point)

Presence assignment, due by 18:30 h today **Graded automatically with testcases only**

Write a function that takes two arguments: an integer `n` and a character `ch`. The function should print the character `ch` in a triangle form as below.
Write a simple program that reads the appropriate variables and prints the result to screen by calling the function.
You can safely assume that the input will be valid.
Your solution has to satisfy the requirements from the problem description and has to pass the following testcase and potentially other testcases which are uploaded. All characters are relevant for passing testcases including newlines and spaces.

Testcase 6.1: input

```
4
$
```

Testcase 6.1: output

```
$$$$
$$$
$$
$
```

Problem 6.2 *Divide I* (1 point)

Presence assignment, due by 18:30 h today **Graded automatically with testcases only**

Write a function `void divby5(float arr[], int size)` that divides by 5 all elements of an array. Your program should print in the `main()` function the elements of the array before and after the division. Test your program with an array that contains the following values:
$5.5, 6.5, 7.75, 8.0, 9.6, 10.36$.
Your solution has to satisfy the requirements from the problem description and has to pass the following testcase and potentially other testcases which are uploaded. All characters are relevant for passing testcases including newlines and spaces.

Testcase 6.2: input

Testcase 6.2: output

```
Before:
5.500 6.500 7.750 8.000 9.600 10.360
After:
1.100 1.300 1.550 1.600 1.920 2.072
```

Problem 6.3 *Determine lowercase characters* (1 point)

Due by Wednesday, October 3rd, 10:00 h **Graded manually**

Write a function `int count_lower(char* str)` that counts the number of lowercase characters within in a string. Then write a program where you repeatedly read a string and determine and print the number of lowercase 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 50 characters and will be valid.

Problem 6.4 *Divide II* (1 point)

Due by Wednesday, October 3rd, 10:00 h **Graded manually**

Modify your solution for *Divide I* such that you first read an integer `n`, and then elements of an array with `n` components. Therefore you will need to dynamically allocate your array. Then divide by `5` the elements using the `divby5()` function and print the result from the `main()` function. 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 the scalar product of two vectors* (1 point)

Due by Wednesday, October 3rd, 10:00 h **Graded automatically with testcases only**

Write a program that reads a number `n`, and then two vectors `v` and `w` of real numbers (of type `double`) with `n` components. Write a function that computes the scalar product of these two vectors. The scalar product is defined as:

$$v \cdot w = \sum_{i=1}^{n} v_i \cdot w_i$$

Use the function to compute the scalar product of the two vectors you read. From the `main()` function print the value of the scalar product on the screen. Additionally write functions for determining and printing on the screen the smallest and largest components of the vector `v`, and the position in the vector where they occur.
You can safely assume that the input will be valid.
Your solution has to satisfy the requirements from the problem description and has to pass the following testcase and potentially other testcases which are uploaded. All characters are relevant for passing testcases including newlines and spaces.

Testcase 6.5: input

```
3
1.1
2.5
3.0
1.0
1.0
1.0
```

Testcase 6.5: output

```
Scalar product=6.600000
The smallest = 1.100000
Position of smallest = 0
The largest = 3.000000
Position of largest = 2
The smallest = 1.000000
Position of smallest = 0
The largest = 1.000000
Position of largest = 0
```

Problem 6.6 *Print numbers counting down* (1 point)

Due by Wednesday, October 3rd, 10:00 h **Graded manually**

Write a program which reads a positive integer `n` from the keyboard. Then write and call a recursive function for printing the numbers $n, n-1, \ldots, 1$.
You can safely assume that the input will be valid.

Problem 6.7 *Determine if a number prime* (1 point)

Due by Wednesday, October 3rd, 10:00 h **Graded automatically with testcases only**

Write a program which reads a positive integer `x`. Then write a recursive function for determining if `x` is a prime number or not. The function should return `1` if the number is prime and `0` if not. Print a corresponding message from the `main()` function.
You can safely assume that the input will be valid.
Your solution has to satisfy the requirements from the problem description and has to pass the following testcase and potentially other testcases which are uploaded. All characters are relevant for passing testcases including newlines and spaces.

Testcase 6.7: input

```
26
```

Testcase 6.7: output

```
26 is not prime
```

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

Due by Wednesday, October 3rd, 10:00 h **Graded manually**

Write a program which reads the first two characters from the file "`chars.txt`" and writes the sum of their ASCII code values as a number into "`codesum.txt`". Use an editor to create the input file "`chars.txt`". Your program is responsible to create the output file "`codesum.txt`".
You can safely assume that the content of the input file will be valid.

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

Due by Wednesday, October 3rd, 10:00 h **Graded manually**

Write a program which reads from the keyboard the names of two files containing two double numbers. Your program should read these two values from the two files, compute their sum, difference, product and division, and write the results on separate lines into the file "`results.txt`".
You can safely assume that the input is valid, the two input files exist and each contains one valid double value.

Bonus Problem 6.10 *Merge two files* (1 point)

Due by Wednesday, October 3rd, 10:00 h **Graded manually**

Write a program which reads the content of two files "`text1.txt`" and "`text2.txt`" line by line and merges them into another file called "`merge12.txt`".
You can safely assume that the input is valid.

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.
- 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 coursenumber, otherwise I might have problems to identify your submission.
- 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 Wednesday, October 3rd, 10:00 h.