Programming in C and C++
Jacobs University Bremen
Dr. Kinga Lipskoch

Course: CH-230-A
October 1st, 2019

Assignment 5 - Arrays, Dynamic Memory Allocation, Recursion

- The problems of this assignment must be solved in C or C++ (instruction in each problem).
- The TAs are grading solutions to the problems according to the following criteria: https://grader.eecs.jacobs-university.de/courses/ch_230_a/2019_2/Grading-Criteria-C-C++.pdf

Problem 5.1 *Triangle chars* (1 point)

Presence assignment, due by 11:00 AM today **Graded automatically with testcases only**
Language: C

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 5.1: input

```
4
$
```

Testcase 5.1: output

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

Problem 5.2 *Divide I* (1 point)

Presence assignment, due by 11:00 AM today **Graded automatically with testcases only**
Language: C

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 5.2: input

Testcase 5.2: 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 5.3 *Determine lowercase characters* (1 point)

Due by Monday, October 7th, 23:00 **Graded manually**
Language: C

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 5.4 *Divide II* (1 point)

Due by Monday, October 7th, 23:00 **Graded manually**

Language: C

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 5.5 *Computing the scalar product of two vectors* (1 point)

Due by Monday, October 7th, 23:00 **Graded automatically with testcases only**

Language: C

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 5.5: input

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

Testcase 5.5: 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 5.6 *Pointer arithmetic* (2 points)

Due by Monday, October 7th, 23:00 **Graded manually**

Language: C

Write a program that counts the number of elements in an array until encountering the first negative value without the usage of any integer counter variables (except for the loop for reading the elements of the array), but with the usage of pointers and pointer arithmetic.
Your program should read an `int` for the length of the array and an array of floats (containing at least one negative value) from the standard input. The number of non-negative values before the first negative value should be printed on the standard output.
You can assume that the input will be valid and the array will always contain at least one negative value. To pass the testcases your output has to be identical with the provided ones.

Testcase 5.6: input

```
5
1.2
3.4
-5.86
4
5.45
```

Testcase 5.6: output

```
Before the first negative value: 2 elements
```

Problem 5.7 *Concatenating two strings* (2 points)

Due by Monday, October 7th, 23:00 **Graded automatically with testcases only**
Language: C

Write a program to concatenate two strings (with the length of at most 100 characters) putting the result in a dynamically allocated array of chars with exact size.
Your program should read from the standard input two strings which may be statically allocated. The dynamically allocated string containing the concatenation result of the two strings should be printed on the standard output.
You can assume that the input will be valid. To pass the testcases your output has to be identical with the provided ones.

Testcase 5.7: input

```
one
two
```

Testcase 5.7: output

```
Result of concatenation: onetwo
```

Problem 5.8 *Dynamically allocated matrix multiplication* (3 points)

Due by Monday, October 7th, 23:00 **Graded manually**
Language: C

Write a program that computes the multiplication of two dynamically allocated matrices.
Your program should dynamically allocate the memory for the three matrices (two input matrices and the result matrix). You should write functions for reading a matrix from the standard input, printing a matrix to the standard output, and finally a function for computing the multiplication of two matrices. At the end, do not forget to deallocate the memory used by the three matrices.
The product of two matrices A and B of dimensions $n \times m$ and $m \times p$ can be calculated as:

$$C_{ik} = \sum_{j=1}^{m} A_{ij} \cdot B_{jk}, \text{ with } i = 1, \ldots, n \text{ and } k = 1, \ldots, p.$$

Your program should read three integers (the dimensions `n`, `m` and `p`) and the elements of two integer matrices from the standard input. The result of the matrix multiplication should be printed on the standard output.
You can assume that the input will be valid. To pass the testcases your output has to be identical with the provided ones.

Testcase 5.8: input

```
2
2
2
1
2
3
4
1
0
0
1
```

Testcase 5.8: output

```
Matrix A:
1 2
3 4
Matrix B:
1 0
0 1
The multiplication result AxB:
1 2
3 4
```

Bonus Problem 5.9 *Printing dynamically allocated 3D-array sections* (4 points)

Due by Monday, October 7th, 23:00 **Graded automatically with testcases only**
Language: C

Write a program that dynamically allocates memory for a 3D-array of integers and prints the 2D-sections parallel to the "XOY axis" (considering the array dimensions row-dimension, column-dimension and depth-dimension similar to the geometrical X, Y and Z dimensions) of a 3D-array.

Your program should read three integer values corresponding to the dimensions of a 3D-array (in the order of rows, columns, depth) and should dynamically allocate the memory for this 3D-array. You should write functions for reading the elements of the 3D-array from standard input (first iterating through rows, then columns and then the depth) and finally a function for printing the 2D-sections of the 3D-array which are parallel to the "XOY axis". Do not forget about the allocation and deallocation of the memory.
You can assume that the input will be valid. To pass the testcases your output has to be identical with the provided ones.

Testcase 5.9: input

```
2
2
3
1
2
3
1
2
3
1
2
3
1
2
3
```

Testcase 5.9: output

```
Section 1:
1 1
1 1
Section 2:
2 2
2 2
Section 3:
3 3
3 3
```

Problem 5.10 *Print numbers counting down* (1 point)

Due by Monday, October 7th, 23:00 **Graded manually**
Language: C
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 5.11 *Determine if a number prime* (1 point)

Due by Monday, October 7th, 23:00 **Graded automatically with testcases only**
Language: C
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 5.11: input

```
26
```

Testcase 5.11: output

```
26 is not prime
```

How to submit your solutions

- Your source code should be properly indented and compile with `gcc` or `g++` depending on the problem without any errors or warnings (You can use `gcc -Wall -o program program.c` or `g++ -Wall -o program program.cpp`). 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:

```
/*
   CH-230-A
   a5_p1.[c or cpp or h]
   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
  `k.lipskoch@jacobs-university.de` **with a subject line that begins with CH-230-A**.
  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 Monday, October 7th, 23:00.