

Assignment 2 - Pointers and Dynamically Allocated Multi-dimensional Arrays

- The problems of this assignment must be solved in C.
- Your programs should have the input and output formatting according to the testcases listed after the problems.
- Your programs should consider the grading rules:
<https://grader.eecs.jacobs-university.de/courses/320112/2018.1gA/Grading-Criteria-C2.pdf>

Problem 2.1 *Pointer arithmetic*

(2 points)

Presence assignment, due by 18:30 h today

Write a program that counts the number of elements in an array until encountering the first value which has the fractional part 0 without the usage of any integer variables for counting.

Your program should read an int for the length of the array and an array of doubles (containing at least one value with fractional part 0) from the standard input. The number of “real” floating point values before the first value with fractional part 0 should be outputted on the standard output.

You can assume that the input will be valid and at least one element with fractional part 0 will be entered.

Testcase 2.1: input

```
5
1.2
-3.4
5.00
4
5.45
```

Testcase 2.1: output

```
Before the first integer: 2 elements
```

Problem 2.2 *Substring of a string*

(2 points)

Presence assignment, due by 18:30 h today

Write a program that extracts a substring (specifying two positions for from and until) of a string putting the result into a dynamically allocated array of chars.

Your program should read from the standard input one string (may be statically allocated with at most 100 elements) and two positions. The dynamically allocated string containing the result of the substring operation on the input string should be printed on the standard output. Please check if the positions are valid or not (you can see the concrete “error” messages which have to be printed on the screen by submitting your solution and observing the results of the comparisons against the specified testcases). You are not allowed to use the `strstr()` function from `string.h`. You may use pointers and the `strcpy()` and `strncpy()` functions.

You can assume that the input will be syntactically valid.

Testcase 2.2: input

```
onestringtwo
3
8
```

Testcase 2.2: output

```
Result of substring(3, 8): string
```

Problem 2.3 *Dynamically allocated matrix comparison*

(3 points)

Write a program that compares two dynamically allocated matrices based on their elements on the main and secondary diagonals.

Your program should dynamically allocate the memory for the two input matrices. You should write functions for reading a matrix from the standard input, printing a matrix to the standard output and finally a function for comparing the main and secondary diagonals of the two matrices. Do not forget about the allocation and deallocation of the memory.

The main diagonal of a square matrix is the diagonal from the left to the right and the secondary diagonal is the one from the right to the left.

Your program should read one integer value (the dimension n) and the elements of two integer matrices from the standard input (first iterating through the rows and then through the columns). The result of the matrix comparison should be printed on the standard output. In case the result of the comparison is that they are not identical then the message "The comparison result: NOT identical" should be printed on the screen.

You can assume that the input will be valid.

Testcase 2.3: input

```
3
1
2
3
4
5
6
7
8
9
1
0
3
0
5
0
7
0
0
9
```

Testcase 2.3: output

```
Matrix A:
1 2 3
4 5 6
7 8 9
Matrix B:
1 0 3
0 5 0
7 0 9
The comparison result: identical
```

Problem 2.4 *Printing dynamically allocated 3D array sections*

(4 points)

Write a program that dynamically allocates memory for a 3D array of integers and prints the 2D sections parallel to the "y0z axis" (considering the array dimensions column-dimension, row-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 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 "y0z axis". Do not forget about the allocation and deallocation of the memory.

You can assume that the input will be valid.

Testcase 2.4: input

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

Testcase 2.4: output

```
Section 1:
1 2 3
1 2 3
Section 2:
4 4 4
4 4 4
```

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. Otherwise you might have problems with the inclusion of header files. Each program **must** include a comment on the top like the following:

```
/*
    JTSK-320112
    a2.p1.c
    Firstname Lastname
    myemail@jacobs-university.de
*/
```

- You have to submit your solutions via *Grader* at **`https://cantaloupe.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 JTSK-320112**.
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 Wednesday, February 14th, 10:00 h.