Assignment 3 - Loops, Arrays, Functions and Strings

- 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: Grading Criteria

Problem 3.1 *Writing numbers* (1 point)
Presence assignment, due by 11:00 AM today   **Graded manually**
Language: C
Write a program where you first enter a float x and then an integer n from the keyboard. Print the float x n times to the screen. Make sure that n will have a valid integer value. In the invalid case repeat entering n until a valid value will be entered.

Testcase 3.1: input	Testcase 3.1: output
1.25	Input is invalid, reenter value
-8	Input is invalid, reenter value
0	1.250000
4	1.250000
	1.250000
	1.250000

Problem 3.2 *Writing characters* (1 point)
Due by Monday, September 23rd, 23:00   **Graded manually**
Language: C
Write a program where you first enter a lowercase character ch and then an integer n from the keyboard. Print the characters ch, ch−1, ..., ch−n on the screen.
You can safely assume that the input is valid and you do not have to do any checks.

Problem 3.3 *Centimeters to kilometers* (1 point)
Due by Monday, September 23rd, 23:00   **Graded manually**
Language: C
Write a program that converts an integer length that is entered from the keyboard in cm to km. Write and use a function `float convert(int cm)` that does the actual conversion.
You can safely assume that the input will be valid.

Testcase 3.3: input	Testcase 3.3: output
12	Result of conversion: 0.000120

Problem 3.4 *Wrong position* (1 point)
Due by Monday, September 23rd, 23:00   **Graded manually**
Language: C
The program below should print the position of the first occurrence of the character 'g' within a string. You can safely assume that 'g' will be contained in the string. Why does it always print the position 0? Fix the program such that it prints the correct position.

```
#include <stdio.h>

int position(char str[], char c)
{
    int idx;
    /* Loop until end of string */
    for (idx = 0; str[idx] != c && str[idx] != '\0'; ++idx)
        /* do nothing */
    return idx;
}

int main() {
    char line[80];
    while (1) {
        printf("Enter string:\n");
        fgets(line, sizeof(line), stdin);
        printf("The first occurrence of 'g' is: %d\n", position(line, 'g'));
    }
}
```

Problem 3.5 *Computing the average of temperature values*   (2 points)
Due by Monday, September 23rd, 23:00   **Graded manually**
Language: C

Write a program where you first enter a character c followed by an integer n and n double values
representing temperatures in Celsius. Use an array for storing the temperatures. You can assume
that not more than 100 temperature values would be entered. Your program should compute
and/or print the following: if c is 's' the sum of the temperatures, if c is 'p' the list of all tem-
peratures, if c it 't' the list of all temperatures in Fahrenheit and if another character was entered
then the arithmetic mean (or average) of all temperatures. The formula for converting Celsius to
Fahrenheit is the following:

$$F = \frac{9}{5} \cdot C + 32.$$

Use a `switch` *instruction in your solution.*
You can safely assume that the input will be valid.

Problem 3.6 *Kilograms and grams to pounds*   (1 point)
Due by Monday, September 23rd, 23:00   **Graded manually**
Language: C

Write a program that converts the units of mass kg and g to pounds. First read the weight of an
object expressed by values for kilograms and grams from the keyboard and convert the units of
mass using the function (written by you as well)

```
float to_pounds(int kg, int g);
```

that does the actual conversion. Note that:
1 kilogram = 2.2 pounds
You can safely assume that the input will be valid.

Testcase 3.6: input	Testcase 3.6: output
5 100	Result of conversion: 11.220000

Problem 3.7 *Printing a form*   (1 point)
Due by Monday, September 23rd, 23:00   **Graded manually**
Language: C

Write a program which reads two integers n, m and a character c from the keyboard. This pro-
gram should define and call a function with the prototype:
```
void print_form(int n, int m, char c);
```
which prints a trapezoid of height n, bases m and m+n-1 consisting of the character c as in the
following testcase.

Testcase 3.7: input	Testcase 3.7: output
4 3 $	$$$ $$$$ $$$$$ $$$$$$

You can safely assume that the input will be valid.

Problem 3.8 *Computing sum and average* (1 point)

Due by Monday, September 23rd, 23:00   **Graded manually**
Language: C

Write a program where you can enter from the keyboard up to 10 floats. If the number entered is equal to -99.0, stop reading numbers from the keyboard and compute the sum and average of all values (excluding -99.0) using two functions (one for the sum and one for the average). Print your results on the screen.
You can safely assume that the input will be valid.
Make sure you consider all the cases: less than 10 numbers might be entered. After all the numbers have been entered you need to make sure that the sum and average are computed.

Problem 3.9 *Determine sum of two values in array* (1 point)

Due by Monday, September 23rd, 23:00   **Graded manually**
Language: C

Write a program which reads from the keyboard an integer value n followed by n double values as elements of an array with not more than 20 elements. Write also a function with the prototype:
`double sum25(double v[], int n);`
which computes and returns the sum of the elements in v at position 2 and position 5. Check that positions 2 and 5 are valid within v, if not then print a corresponding message on the screen. In this case the function should return the value -111. Print the sum on the screen.

Testcase 3.9: input	Testcase 3.9: output
6 1.5 1.5 1.5 1.5 1.5 −1.5	sum=0.000000

Problem 3.10 *Return changes by reference* (1 point)

Due by Monday, September 23rd, 23:00   **Graded manually**
Language: C

Write a program which reads two float values from the keyboard. Then write three functions.
The first function should return the product of the two float values and should have the prototype:
`float product(float a, float b);`
The second function should return the product of the two float values by reference and should have the prototype:
`void productbyref(float a, float b, float *p);`
The third function should add 3 to the first float and 11 to the second float and return the change by reference. It should have the prototype:
`void modifybyref(float *a, float *b);`
Show that the calls of the first two functions have the same effect. Also show what is the effect of calling `modifybyref`.
You can safely assume that the input will be valid.

Problem 3.11 *Working with strings* (2 points)

Due by Monday, September 23rd, 23:00   **Graded manually**
Language: C

Write a program where you can enter two strings called `one` and `two` from the keyboard. The string should be able to contain spaces. The program should do the following:

1. Print on the screen the lengths of both strings,

2. Print on the screen the concatenation of `one` with `two`,

3. Declare a third string, copy correctly `two` into it and print the third string to the screen,

4. Compare the two strings `two` and `one` and print a corresponding message to the screen,

5. Read a character `c` from the keyboard and search for `c` in `two`. The position of the first occurrence of `c` within `two` should be printed to the screen. If the character is not contained in the string then print a corresponding message on the screen.

For solving this problem use the string functions from `string.h`.
Learn how to use them with the help of the man pages.

Testcase 3.11: input	Testcase 3.11: output
| ```
first string
hello world
l
``` | ```
length1=12
length2=11
concatenation=first stringhello world
copy=hello world
one is smaller than two
position=2
``` |

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 Teams.
  Each program **must** include a comment on the top like the following:
  ```
  /*
     CH-230-A
     a3_p1.[c or cpp or h]
     Firstname Lastname
     myemail@constructor.university
  */
  ```

- You have to submit your solutions via *Teams*.
  If there are problems (but **only** then) you can submit the programs by sending mail to
  `klipskoch@constructor.university` **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, September 23rd, 23:00.