Programming in C I, Group A
Jacobs University Bremen
Dr. Kinga Lipskoch

Course: JTSK-320111
September 22^{nd}, 2017

Assignment 4 - Functions and Strings

- 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/320111/2017_2gA/Grading-Criteria-C.pdf

Problem 4.1 *Wrong position* (1 point)

Presence assignment, due by 18:30 h today

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 4.2 *Centimeters to kilometers* (1 point)

Presence assignment, due by 18:30 h today

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.

Problem 4.3 *Kilograms and grams to pounds* (1 point)

Presence assignment, due by 18:30 h today

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.

Problem 4.4 *Printing a form* (1 point)

Write a program which reads two integers `n`, `m` and a character `c` from the keyboard. This program 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 4.4: input

```
4
3
$
```

Testcase 4.4: output

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

You can safely assume that the input will be valid.

Problem 4.5 *Computing sum and average* (1 point)

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

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 elements of the array and the sum on the screen.

Problem 4.7 *Return changes by reference* (1 point)

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 4.8 *Working with strings* (2 points)

Write a program where you can enter two strings called `one` and `two` from the keyboard. 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.

Bonus Problem 4.9 *Detect all occurrences of a character* (1 point)

Modify your solution for **Problem 4.8** such that all occurrences of the character c can be detected in two. All positions of occurrence should be printed to the screen.

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
     a4_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 k.lipskoch@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.
- 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, September 27th, 10:00 h.