UNIVERSITÄT DUISBURG ESSEN

NES Networked Embedded Systems

Programming in C/C++

Exercise 1
!!! Deadline: Tue, Oct 29rd, 23:55 !!!

Task 01:
Write a program that prints the sizes of all simple data types in the following order: char, short, int, long, long long, float, double, long double, pointer. First print the name, then its size.
Beginning of example output:

```
char 1
short 2
...
```

Hint: This can be achieved using the printf()command:

Include the output in your report, along with name and version of the compiler and your platform (Windows/Linux/Mac, 32/64 bit). If possible, run with different compilers and on different platforms. Report any differences you observe.

Submission: Your executable should be called **printsizes**
Points: (code 15 pts, report 5 pts)

Task 02:
In this exercise you will learn about the internal organisation of multidimensional arrays.
Start with defining the following local variable in the `main()` function of your program as such:

```
int cube1[3][2][4] = { {{1,2,3,4}}, {{2,3}, {4,6,8,10}}, {{3,4,5,6}, {6,8,10}} };
```

Now, add some printf() statements printing out the addresses of some cube elements so that you are able to determine the internal storage structure of the cube.
Hint: The addresses are given in bytes – use what you learn about size of an int in Task 01.

Make a report on what you discover about the cube internal storage structure. (But do not include these printf() statements in your final program!)

For the final program, define an additional local variable in `main()`:

```
int cube2[2][3][3] = { {{9,8,7}, {6}, {5,4}}, {{3,2}, {1,2,3}, {4}} };
```

The cubes can be interpreted as multiple two-dimensional matrices.
Write a function "`sum2nd`" that returns the sum of all numbers in the 2nd columns of these matrices. For example, for cube1 it shall return the sum of 2,3,6,...
The function shall accept a pointer to an int as first parameter and the dimensions of the cube as 2nd to 4th parameter, i.e. the function can be called as:

```
sum2nd( &cube1[0][0][0], 3, 2, 4)
```

You can assume that there is always a 2nd column in the cube.
In `main()`, just print out the return value of this function, invoked with appropriate parameters for cube1 and cube2. The output shall only contain two integers (one per line).

Submission: Your executable should be called **cubes**
Points: (code 30 pts, report 20 pts)

Task 03:

Write a program that prints out the length of a string without using the function `strlen()` from the standard library. Implement the functionality yourself.

The program accepts a single string as command line parameter. If more or less parameters are given it shall only print "wrong input", otherwise it prints out the length. Implement a solution using an integer index and a solution using pointer arithmetic only, i.e. no additional length counter or index, and no `sizeof()` operator! When the program is started, call both implementations of the solution and print out the results in two separate lines.
Example: When the program is called with *helloworld* as parameter, it shall print:

```
10
10
```

Submission: Your executable should be called **strlength**
Points: (*code 30 pts, no report necessary)*