

Programming Exercise 4

Due date and time: 6. April 2009, 23:59

You should work out the exercises independantly. Group cooperation is not allowed. Submit the sensibly documented program files. (Files for Bsp. 5: `bsp_5.cpp`, `bsp_5_fkt.cpp`, `bsp_5_fkt.hpp`) Copy these to your directory in the global submission folder.

10. Compute the Taylor series expansion $\ln x$, $x \in]0, 2]$ up to the n th term

$$\begin{aligned}\ln x &\approx \sum_{k=1}^n (-1)^{k+1} \frac{(x-1)^k}{k} & (1) \\ &= \frac{x-1}{1} - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3} - \dots + (-1)^{n+1} \frac{(x-1)^n}{n}\end{aligned}$$

for the given values x and n . Return the computed value and compare it with (1 Pkt.) the exact value.

Add: Try to do this without using the `pow` function.

Test data: (x, n) : (1.9, 16), (0.01, 1000), (0.01, 10)

11. Compute the Taylor series approximation of equation (1) until you have a relative error of ε with respect to the exact value.

Print the approximated value, the difference wrt. the exact value and the last index n that contributed to the sum.

You are **not allowed** to use the commands `break`, `continue`, or `goto` in this problem (nor may you do so within the scope of the course).

Hint: A formula for the relative error $\frac{|v_{\text{exact}} - v_{\text{approx}}|}{|v_{\text{exact}}|}$ provided $v_{\text{exact}} \neq 0$. (1 Pkt.)

Test data: (x, ε) : (1.9, 1e-5), (0.01, $3.2 \cdot 10^{-12}$)

12. Write a program that reads in a C++-**string**¹ until the given string agrees with a self-specified password string.

With each failed attempt, triple a penalty value, which, for example, could be the waiting time required before a new attempted entry may be tried. (1 Pkt.)

Permit a maximum of five attempts only.

Hint: Begin with the first portion of the exercise and continue your previous solution in the next portion(s) of the exercise(s).

You are **not allowed** to use the commands `break`, `continue`, or `goto` in this problem (nor may you do so within the scope of the course).

choose your own entry data.

¹<http://www.uni-graz.at/~haasegu/Lectures/Kurs-C/Script/html/node23.html>

13. **Double loop:** Print a table with a natural number $k \in \mathcal{I}$, its square, cubic number and so on to the p th power ($p \geq 3$) per row: (2 Pkt.)

number	square	cubic	...	number ^{p}
4	16	64	...	calculate 4^p
\vdots	\vdots	\vdots	...	\vdots
k	k^2	k^3	...	k^p
\vdots	\vdots	\vdots	...	\vdots
n	n^2	n^3	...	n^p
$\sum_{k \in \mathcal{I}} k$	$\sum_{k \in \mathcal{I}} k^2$	$\sum_{k \in \mathcal{I}} k^3$		

Only the sums of numbers, squares and cubics should be calculated.

Choose $k \in \mathcal{I}$ as all those natural numbers that can be divided by 4 and $k \leq n$.

Test data (n, p) : (33, 6), (431, 3)

Hinweis 1: Test first with all natural numbers $k = 1, \dots, 10$.

Hinweis 2: The formatted output for the table is described in §10 of script² (rechtsbündige Ausgabe; Platzhalter), don't forget `#include <iomanip>` .

Hinweis 3: What happens for data $n = 432$ and $p = 3$?

general hint:

Show the following C++-functionality and use it in the given case: `for`, `while`, `do`, `fabs`³

You are **not allowed** to use the commands `break`, `continue`, or `goto` .

Submission of solutions:

The submission of the solutions (as *.cpp-Files, *.hpp, ..., **no exe-Files**) may be carried out exclusively by copying these files in your directory (initially created as) `Lastname_Firstname` into the global submission folder `y: \\pers.ad.uni-graz.at\fs\ou\621\stud_haase>Lastname_Firstname` .

Instructions on how to do this can be found at: LV-Homepage⁴ .

The file names **must** follow the format `bsp_nummer`, with the corresponding file extension. For example, ist in Beispiel 1 (example 1) the file `bsp_1.cpp` would be submitted. Files with names which deviate from this formula will not be considered as submitted files.

²<http://www.uni-graz.at/~haasegu/Lectures/Kurs-C/Script/html/node93.html>

³<http://www.cplusplus.com/reference/clibrary/cmath/fabs.html>

⁴<http://www.uni-graz.at/~haasegu/Lectures/Kurs-C/SS08/index.html>