

Aufgabenblatt 2

Abgabedatum: 15. Mai 2017

Die mit einem '*' markierten Aufgaben sind Pflicht. Sie können insgesamt 10 Punkte erreichen. Testen sie alle Funktionen innerhalb der *main*-Funktion.

Achtung: Jede Aufgabe die Sie abgeben muss ein eigenes lauffähiges Programm sein, daher eine *main*-Funktion enthalten und **ausführbar** sein. Benennen Sie ihre zugehörigen Sourcedateien nach dem Muster '*sheet1-exercise1.cpp*'. Packen Sie alle Aufgaben in einem **einzelnem ZIP-Archiv**, das sie nach dem Muster '*SheetX-Surname.zip*' benennen und laden sie dieses im Moodle hoch. Beispiel: '*Sheet1-Mustermann.zip*'. Abgaben die diese Modalitäten nicht einhalten werden automatisch mit **0 Punkten** bewertet.

AUFGABE 1 Wiederholung

Schreiben Sie ein Programm, das

- (a) alle ungeraden Zahlen von 1 bis zu einer vom Benutzer spezifizierten Zahl summiert und das Ergebnis auf der Konsole ausgibt.
- (b) alle geraden Zahlen zwischen 1 und 20 quadriert und auf der Konsole ausgibt.
- (c) eine Multiplikationstabelle für eine vom Benutzer spezifizierte Zahl ausgibt

AUFGABE 2 Primzahlen I (*)

Schreiben Sie eine Funktion `bool is_prime(int number)`. Die Funktion soll **true** zurückgeben, falls der übergebene Parameter eine Primzahl ist, ansonsten **false**. Beachten Sie, dass der Parameter negative Zahlen erlaubt.

AUFGABE 3 Primzahlen II

Schreiben Sie eine Funktion `void print_primes(int upto)`. Die Funktion soll alle Primzahlen bis inklusive *upto* in der Konsole ausgeben. Verwenden Sie hierzu Ihre Funktion aus Aufgabe 1. Die Ausgabe soll beispielsweise wie folgt aussehen:

```
>> Prime numbers from 1 to 10:  
>> 2 3 5 7 9
```

AUFGABE 4 Potenzen (*)

Schreiben Sie eine Funktion `int pow(int base, int exp)`. Diese Funktion soll zur Berechnung von Potenzen dienen:

$$a^n = \underbrace{a * a * a \dots a}_{n\text{-mal}}$$

Beachten Sie auch den oder die Randfälle!

AUFGABE 5 Von Strings zu Zahlen (*)

Schreiben Sie eine Funktion `int convert_number(std::string number)`. Diese Funktion soll einen String wie `"4582"` in eine Ganzzahl umwandeln. Sie dürfen davon ausgehen, dass die übergebenen Strings valide und nicht vorzeichenbehaftet sind.

Hinweise:

1. Natürliche Zahlen werden in der b -adischen Darstellung durch eine endliche Folge von Ziffern in der Form $a_n \dots a_2 a_1 a_0$ dargestellt. Dieser Ziffernfolge wird nun die Zahl

$$\sum_{i=0}^n a_i * b^i = a_0 + a_1 * b + a_2 * b^2 + \dots + a_n * b^n$$

zugeordnet, wobei a_i der der Ziffer a_i zugewiesene Ziffernwert ist.¹

2. Um einen `char` in seinen korrekten Ziffernwert umzuwandeln, müssen Sie von seinem eigentlichen Wert 48 abziehen.² Zum Beispiel:

```
1         char num = '9';
2
3         std::cout << (int)num << std::endl // 57
4             << num - 48 << std::endl; // 9
5
```

AUFGABE 6 Von Strings zu Zahlen in anderen Zahlensystemen

Schreiben Sie eine Funktion `int convert_number(std::string number, int base)`. Wie bereits in der vorherigen Aufgabe sollen Strings in Ganzzahlen umgewandelt werden. Allerdings soll es hier dem Benutzer zusätzlich ermöglicht werden die Basis der Konversion frei zu wählen, um so beispielsweise binäre Zahlen umzuwandeln. So soll der String `"1011"` zum Beispiel `11` ergeben. Wie zuvor dürfen Sie davon ausgehen, dass die übergebenen Strings valide und nicht vorzeichenbehaftet sind. Weiterhin können Sie davon ausgehen, dass $0 < base < 10$ gilt.

AUFGABE 7 Diamond-Pattern (*)

Schreiben Sie ein Programm, das auf der Konsole ein Diamantmuster ausgibt. Fragen sie den Benutzer nach der Breite des Musters. Falls der Benutzer eine gerade oder negative Breite angibt geben Sie einen Fehler aus, ansonsten das Muster in der festgelegten Breite. Die Ausgabe könnte wie folgt aussehen:

```
>> Please provide a width for the diamond pattern (must be odd): 4
>> Error: the width must be an odd positive number! Please try again!
>>
>> Please provide a width for the diamond pattern (must be odd): 5
>>
>>  *
>> ***
>> *****
>> ***
>>  *
```

¹<https://de.wikipedia.org/wiki/Stellenwertsystem>

²https://de.wikipedia.org/wiki/American_Standard_Code_for_Information_Interchange