Datentypen

Speicherplatz der Datentypen

Daten brauchen Platz im Arbeitsspeicher. Jedes ASCII-Zeichen verwendet 1 Byte.

Zähle ab, wie viele Bytes durch die folgenden Strings belegt werden.

```
string antonVonDerSeite = "||||||\n|  ||||\n @  |||\nL    ||\n ~    |\n";
```

```
string antonVonHinten = "|||||||\n|||||||\n|||||||\n| ||| |\n|     |";
```

```
string antonImWind = "\\\\\\\\\\\\\n  \\\\\\\\\\\n| 0 0 |\n|  L  |\n|  ~  |";
```

```
string text  = "Anton sagte: \"Ich lerne jetzt C.\" "
```

Der Backslash leitet ein Sonderzeichen ein. Um ihn selbst als Zeichen anzugeben, verwendet man \\.
Wenn innerhalb eines Strings ein " erscheinen soll, so darf es auch nur als Zeichen und nicht als Ende des Strings verstanden werden: Also \".
Beachte \n und \\ und \“ sind nur 1 Zeichen.

Computer rechnen nicht mit Unendlich

Und wie viele Bytes benutzen die Datentypen bool, int, long und double?
bool ist der Datentyp, der nur ja/nein bzw. wahr/falsch unterscheidet: true und false .

byte	0 bis 255	unsigned 8-bit integer (1 Byte)
short	-32768 bis 32767	signed 16-bit integer (2 Byte)
int	-2147483648 bis 2147483647	signed 32-bit integer (4 Byte)
long	0x0 bis 0x7FFFFFFFFFFFFFFF	die positiven 64-bit Zahlen
	0x8000000000000000 bis 0xFFFFFFFFFFFFFFFF	die negativen 64-bit Zahlen (8 Byte)

Kommazahlen

Typ	Bereich	Genauigkeit
float	±1.5e−45 to ±3.4e38	7 Ziffern (4 Byte)
double	±5.0e−324 to ±1.7e308	15-16 Ziffern (8 Byte)

Man kann auch die **Funktion sizeof( )** benutzen**.** Man übergibt ihr einen Datentyp und erhält die Anzahl von Bytes, die für ihn reserviert werden.

Z.B. liefert sizeof(bool) die Zahl 1. Das bedeutet, dass jede Variable dieses Typs im Arbeitsspeicher 1 Byte für sich reserviert.

Erstelle ein Programm, das für bool, int, long und double den Rückgabewert von sizeof() ausgibt.

Über die Grenze rechnen

Und was passiert, wenn man einen Datentyp überfordert, d. h. über die Grenze hinaus zu rechnen versucht?
C-Compiler machen keine Überlaufprüfungen und das führt zu dem Effekt, dass der Wert in den negativen Zahlbereich "kippt".

Schreibe ein Programm das die größtmögliche Zahl des Datentyps "int" um 1 erhöht und gib das Ergebnis aus.

Berechne auch die Summe der "double" Zahlen 0.1 + 0.999999999999999. (15 Stellen hinter dem Komma)
Bei wie viel Stellen hinter dem Komma wird kein exaktes Ergebnis mehr geliefert?

Technische Hinweise:

Die Darstellung von Hexadezimalen Zahlen beginnt mit "0x". Berechne den Wert von 0x7FFFFFFF + 1
Die Darstellung von Gleikommazahlen (double) in der Exponentialdarstellung:
5.0E9 ist 5 Milliarden und 1 Milliardstel ist 1.0E-9

Testdaten

0.4e-40 und 0.3e-20 sollen addiert werden. Das ergibt 3.0E-21 und ist damit dasselbe wie 0.3e-20.
Ist das richtig?

MinMaxWerte

Die kleinsten und größten Werte eines Zahl-Datentyps sind als Konstante abrufbar.

```
#include <limits.h>

cout << (INT_MAX) << endl;
cout << (INT_MIN) << endl;
```

Computer-Super-GAU

Viele Rechnersysteme haben für ihren Sekundenzähler 32 Bit Speicherplatz. Wenn dieser für die Anzahl der seit dem 1.1.1970 0:00:00 verstrichenen Sekunden nicht mehr ausreicht, stellt sich das System wieder auf die Zeit 0 zurück. Berechne, in welchem Jahr dieser Computer-SuperGAU voraussichtlich stattfinden wird. Du wirst es ziemlich sicher noch erleben.

Technische Hinweise:

Nimm für ein Jahr 365,25 Tage
Verwende als Datentyp unsigned int.

Operatoren

Man beachte den Unterschied: `a = 1;` (a bekommt den Wert 1 zugewiesen) und `a == 1;` (der Inhalt von a wird mit 1 verglichen). Eine Wertzuweisung kann auch mit einer Rechenoperation verbunden werden, deren Ergebniswert der Variablen dann zugewiesen wird. Nur rechts vom „=“-Zeichen darf ein Rechenausdruck stehen.

```
bool b = ( a = = b );
// falls a und b gleich sind, erhält b den Wert “true”, sonst “false”
```

Operatoren-Übersicht

Zuweisen	= += -= *= /= %= &= |= ^= <<= >>= ??
Vergleichen	== != < > <= >=
Berechnungen (Arithmetic)	+ - * / %
Bit Verknüpfungen	& | ^ ! ~ AND OR XOR NOT Komplement
Logische Verknüpfungen	! && || true false NOT AND OR
String-Verknüpfung	+
Increment, decrement	++ --
Bitweises Verschieben (Shift)	<< >>
Indexing (Arrays)	[ ]
Cast	( )
Wenn-Operator	?:
Objekt-Erzeugung	new
Member eines Objekts	.
Typ-Information	as is sizeof typeof
Overflow Exception	checked unchecked
Indirection , Adresse	* -> [ ] &

Inkrement

```
i = 0;
i = i + 5; // ist dasselbe wie i += 5;
a = i++ * 3; // es wird zuerst mit 3 multipliziert, dann i um 1 erhöht
a = ++i * 3; // es wird zuerst um 1 erhöht, dann mit 3 multipliziert.
```

Dekrement

```
a = i-- * 3; // es wird zuerst mit 3 multipliziert, dann i um 1 vermindert
a = --i * 3; // es wird i zuerst um 1 vermindert, dann mit 3 multipliziert.
```

Der 3-stellige Wenn-Operator ?:

Er prüft eine Bedingung und gibt bei „true“ den Wert vor dem Doppelpunkt, bei „false“ den Wert nach dem Doppelpunkt aus.

Komplement-Operator

Anton's Rechentrick: Anton kann nicht mit "Minus" rechnen. Er macht es ebenso wie der Prozessor, der auch nur addieren kann:

"A minus B" ist "A plus Komplement von B plus 1"

A - B ist A + ~B + 1

Prüfe in einem Programm, ob beide Rechenarten gleich sind.
Funktioniert das auch bei Gleitkommazahlen?

Testdaten

12345 - 6789
0 - 999
10000 - 10001
0x7FFFFFFF - 0x7FFFFFFE

Modulo-Operator

Der Modulo-Operator "%" kann nur bei ganzen Zahlen verwendet werden und gibt den Teilerrest der Division zurück:
23 % 5 = 3

Berechne 10001 % 47