



Übungsstunde W02

Computer Science (CSE) – AS 23

① siehe revidiertes Handout
für mehr Info zu "Expressions"

Heutiges Programm

Kennenlernen

Organisatorisches

Integer Division & Modulo

Binärrepräsentation

Tipps zu [code]expert

Outro



scan me

`n.ethz.ch/~agavranovic`

Herzlich Willkommen!



Ziele für heute

- Uns kennenlernen
- Organisatorisches besprechen
- [code]expert einrichten
- Unklarheiten klären
- Int-Division, Operatoren und Modulo verstehen
- Zahlen in andere Basen umrechnen können

1. Kennenlernen

Adel

- Hatte vor der ETH praktisch keine Erfahrungen im Programmieren

Adel

- Hatte vor der ETH praktisch keine Erfahrungen im Programmieren
- Hatte gute Note in Informatik am BPB-I

Adel

- Hatte vor der ETH praktisch keine Erfahrungen im Programmieren
- Hatte gute Note in Informatik am BPB-I
- Wird dafür sorgen, dass Ihr den BPB-I nicht wegen Informatik verkackt

Adel

- Hatte vor der ETH praktisch keine Erfahrungen im Programmieren
- Hatte gute Note in Informatik am BPB-I
- Wird dafür sorgen, dass Ihr den BPB-I nicht wegen Informatik verkackt
- Hier, um euch zu helfen

Euer Assistent

Adel

- Hatte vor der ETH praktisch keine Erfahrungen im Programmieren
- Hatte gute Note in Informatik am BPB-I
- Wird dafür sorgen, dass Ihr den BPB-I nicht wegen Informatik verkackt
- Hier, um euch zu helfen
- Trinkt gern guten Kaffee
- Fährt gern Fahrrad
- Aus Züri



Eure Kommilitoninnen und Kommilitonen

Stell dich vor!

Eure Kommilitoninnen und Kommilitonen

Stell dich vor!

- Wie heisst du?
- Woher kommst du?
- Wieso RW an der ETH?
- Hast du Erfahrung im Programmieren (insb. C++)?
- Hast du coole Hobbies oder aussergewöhnliche Interessen?

Fragen/Unklarheiten?

2. Organisatorisches



Informatik

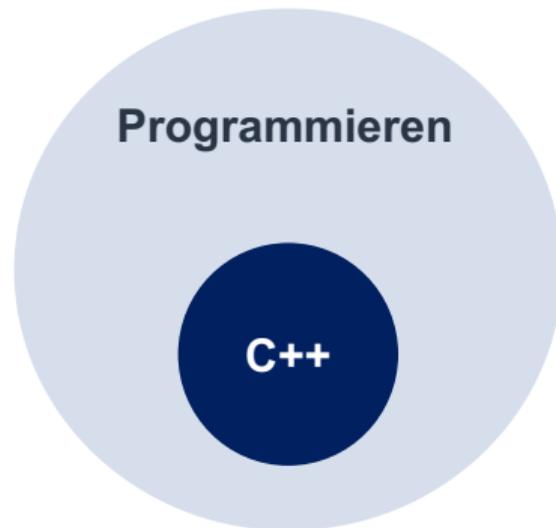
Informationen zur Organisation

Kennenlernen

- Name?
- Welcher Teilbereich deines Studiums interessiert dich am meisten?

Ziel des Kurses

- Vorlesung
- Übungsstunde
- Wöchentliche Übungen
- Bonusaufgaben
- Study Center



Wöchentlicher Plan

Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Samstag	Sonntag	Montag
Vorlesung CSE	Vorlesung FMP	Tutorial CSE					
	US CSE Übungsstunden <i>here</i>						
		Study Center					

Hausaufgaben auf CodeExpert lösen
(Ausgabe am Montag 6:00, Abgabe am folgenden Montag 18:00)

Wöchentliche Übungen und Bonusaufgaben

- Alle Übungen findet ihr auf [code]expert (<https://expert.ethz.ch>).
- Ihr müsst euch zuerst mithilfe dem euch gesandten Link in die Übungsgruppe einschreiben.

- **Wöchentliche Übungen:**
 - Zweck: Neuen Stoff anwenden.
 - Veröffentlichung: Montag um 6:00.
 - Deadline: Eine Woche später (Montag 18:00).
 - Erlaubt euch Erfahrungspunkte (XP) zu sammeln.

- **Bonusaufgaben** (benötigt ca. 2/3 der Erfahrungspunkte zur Freischaltung):
 - Zweck: Wissen zu verschiedenen Themen kombinieren.
 - Erlaubt euch max. +0.25 Bonus für die Endnote zu erreichen (mit 2/3 der Bonuspunkte).

Übungsstunden

- Zweck: Vorbereitung für das Lösen von zukünftigen und vergangenen Übungen.
- Ansatz: Vor allem interaktiver Unterricht und konstruktive Diskussionen.
- Wir erwarten, dass ihr:
 - Aktiv am Unterricht teilnehmt
 - Fragen stellt, wenn ihr den Inhalt nicht versteht, oder warum wir ein bestimmtes Thema behandeln
- Es ist völlig normal, wenn ihr Fehler macht, wir sind im Lernprozess. Bitte vermeidet es, Sachen zu tun, welche andere stören könnten. Wenn eine Aufgabe zu einfach ist, helft anderen.

Study Center

- Zweck: Eine Möglichkeit, um nach individueller Hilfe zum Kurs zu fragen

IFMP:

- Zeit: Mittwoch 16:15-18:15, beginnend ab dem 27. September
- Ort: Mensa Polyterrasse
- Link: <https://studycenter.ethz.ch/>

CSE (mit MAVT geteilt):

- Zeit: Mittwoch 18:15-20:00, beginnend ab dem 4. Oktober
- Ort: HG F3

Informationen & Kontakt

- Mehr Informationen sind auf dem Informationsblatt zur Organisation zu finden.
- Für Fragen betreffend der *Vorlesungsinhalte* könnt ihr während der Vorlesung fragen.
- Für Fragen betreffend der *Übungen* könnt ihr euren TA fragen.
- Für *administrative* Fragen kontaktiert bitte den Head-TA (siehe Website für die E-Mail-Adresse).

Fragen/Unklarheiten?

Zur Übungsstunde

- Alles wird auf der ~~meiner~~ Website hochgeladen (meistens noch am gleichen Tag)

Zur Übungsstunde

- Alles wird auf der meienr Website hochgeladen (meistens noch am gleichen Tag)
- Wird nicht aufgezeichnet

Zur Übungsstunde

- Alles wird auf der meienr Website hochgeladen (meistens noch am gleichen Tag)
- Wird nicht aufgezeichnet
- Bitte stellt Fragen bei Unklarheiten
- Bitte beteiligt euch am Unterricht
- Bitte korrigiert mich, sobald ich Fehler mache

Zur Übungsstunde

- Alles wird auf der meiner Website hochgeladen (meistens noch am gleichen Tag)
- Wird nicht aufgezeichnet
- Bitte stellt Fragen bei Unklarheiten
- Bitte beteiligt euch am Unterricht
- Bitte korrigiert mich, sobald ich Fehler mache
- Ihr könnt Fragen auch direkt in euren Code bei `[code]expert` schreiben, aber das sehe ich erst *nach* der Abgabe

```
int a = 42; // Eure Kommentare und Fragen hierhin
```

- Fragen via Mail sind immer willkommen

Zum Kurs

- Ist nicht der schwerste Kurs...

Zum Kurs

- Ist nicht der schwerste Kurs...
- ...aber einer der wichtigsten

Zum Kurs

- Ist nicht der schwerste Kurs...
- ...aber einer der wichtigsten
- Falls ihr den BPB-I nicht besteht, wird es wahrscheinlich nicht an Informatik liegen

Zum Kurs

- Ist nicht der schwerste Kurs...
- ...aber einer der wichtigsten
- Falls ihr den BPB-I nicht besteht, wird es wahrscheinlich nicht an Informatik liegen
- Prüfungen werden erst im Januar stattfinden, aber...

Zum Kurs

- Ist nicht der schwerste Kurs...
- ...aber einer der wichtigsten
- Falls ihr den BPB-I nicht besteht, wird es wahrscheinlich nicht an Informatik liegen
- Prüfungen werden erst im Januar stattfinden, aber...
- **...ÜBT WÄHREND DES SEMESTERS UND BLEIBT DRAN**

Zum Kurs

- Ist nicht der schwerste Kurs...
- ...aber einer der wichtigsten
- Falls ihr den BPB-I nicht besteht, wird es wahrscheinlich nicht an Informatik liegen
- Prüfungen werden erst im Januar stattfinden, aber...
- **...ÜBT WÄHREND DES SEMESTERS UND BLEIBT DRAN**
- ...und falls ihr doch mal abhängt: besorgt euch Hilfe, z.B. bei
 - mir
 - dem Study Center
 - euren Mitstudierenden
 - der Kanzlei

How to `[code]`expert

(Öffnet `expert.ethz.ch` auf dem Gerät eurer Wahl)

- `[code]`expert kann ein wenig pingelig sein

How to `[code]`expert

(Öffnet `expert.ethz.ch` auf dem Gerät eurer Wahl)

- `[code]`expert kann ein wenig pingelig sein
- deshalb folgt den Anweisungen sehr genau (vermeidet unnötigen Text)

How to [code]expert

(Öffnet `expert.ethz.ch` auf dem Gerät eurer Wahl)

- [code]expert kann ein wenig pingelig sein
- deshalb folgt den Anweisungen sehr genau (vermeidet unnötigen Text)
- der Autograder wird den Grossteil der Korrekturen übernehmen
- von mir bekommt ihr die letzten paar Punkte für Stil, Dokumentation, Herangehensweise, etc.

How to [code]expert

(Öffnet `expert.ethz.ch` auf dem Gerät eurer Wahl)

- [code]expert kann ein wenig pingelig sein
- deshalb folgt den Anweisungen sehr genau (vermeidet unnötigen Text)
- der Autograder wird den Grossteil der Korrekturen übernehmen
- von mir bekommt ihr die letzten paar Punkte für Stil, Dokumentation, Herangehensweise, etc.
- Textaufgaben werden vollständig manuell korrigiert

How to [code]expert

(Öffnet `expert.ethz.ch` auf dem Gerät eurer Wahl)

- [code]expert kann ein wenig pingelig sein
- deshalb folgt den Anweisungen sehr genau (vermeidet unnötigen Text)
- der Autograder wird den Grossteil der Korrekturen übernehmen
- von mir bekommt ihr die letzten paar Punkte für Stil, Dokumentation, Herangehensweise, etc.
- Textaufgaben werden vollständig manuell korrigiert
- Feedback zu den Aufgaben könnt ihr innerhalb einer Woche (von mir) erwarten (falls dringend, einfach eine Mail an mich)

Fragen/Unklarheiten?

3. Integer Division & Modulo

Der Unterschied zwischen = und ==

Assignment Operator =

gebraucht, um Variablen Werte zuzuweisen

Der Unterschied zwischen = und ==

Assignment Operator =

gebraucht, um Variablen Werte zuzuweisen

```
int a = 42; // assigns the value 42 to the variable a  
int b = 18; // assigns the value 18 to the variable b
```

Equality Operator ==

gebraucht, um Gleichheit zwischen Variablen zu überprüfen

Der Unterschied zwischen = und ==

Assignment Operator =

gebraucht, um Variablen Werte zuzuweisen

```
int a = 42; // assigns the value 42 to the variable a
int b = 18; // assigns the value 18 to the variable b
```

if(a == b) { }

Equality Operator ==

gebraucht, um Gleichheit zwischen Variablen zu überprüfen

```
(a == b) // this "expression" will equal 1 (true)
         // or 0 (false) ("boolean")
```

Integer Division & Modulo

Integer Division & Modulo

Integer Division

Der Compiler “ignoriert” Dezimalstellen, wenn er `(unsigned) int` durch `(unsigned) int` teilt.

Integer Division & Modulo

Integer Division

Der Compiler “ignoriert” Dezimalstellen, wenn er `(unsigned) int` durch `(unsigned) int` teilt.

Modulo

Division mit Rest, nur ~~ohne~~ Rest

Integer Division & Modulo

Integer Division

Der Compiler “ignoriert” Dezimalstellen, wenn er `(unsigned) int` durch `(unsigned) int` teilt.

Modulo `%`

Division mit Rest, nur ~~ohne~~ Rest

$7/3 ==$

Integer Division & Modulo

Integer Division

Der Compiler “ignoriert” Dezimalstellen, wenn er `(unsigned) int` durch `(unsigned) int` teilt.

Modulo

Division mit Rest, nur ~~ohne~~ Rest

$$7/3 == 2$$

$$15/4 ==$$

Integer Division & Modulo

Integer Division

Der Compiler "ignoriert" Dezimalstellen, wenn er `(unsigned) int` durch `(unsigned) int` teilt.

Modulo

Division mit Rest, nur ~~ohne~~ Rest

$$7/3 == 2$$

$$15/4 == 3$$

$$16/4 ==$$

Integer Division & Modulo

Integer Division

Der Compiler "ignoriert" Dezimalstellen, wenn er `(unsigned) int` durch `(unsigned) int` teilt.

Modulo

Division mit Rest, nur ~~ohne~~ Rest

$$7/3 == 2$$

$$15/4 == 3$$

$$16/4 == 4$$

$$7\%3 ==$$

Integer Division & Modulo

Integer Division

Der Compiler "ignoriert" Dezimalstellen, wenn er `(unsigned) int` durch `(unsigned) int` teilt.

Modulo

Division mit Rest, nur ~~der~~ Rest ~~?~~ ^{der} ~~?~~

$$7/3 == 2$$

$$15/4 == 3$$

$$16/4 == 4$$

$$7\%3 == 1$$

$$15\%4 ==$$

Integer Division & Modulo

Integer Division

Der Compiler "ignoriert" Dezimalstellen, wenn er `(unsigned) int` durch `(unsigned) int` teilt.

Modulo

Division mit Rest, nur ~~ohne~~ Rest

$$7/3 == 2$$

$$15/4 == 3$$

$$16/4 == 4$$

$$7\%3 == 1$$

$$15\%4 == 3$$

$$16\%4 ==$$

Integer Division & Modulo

Integer Division

Der Compiler "ignoriert" Dezimalstellen, wenn er `(unsigned) int` durch `(unsigned) int` teilt.

Modulo

Division mit Rest, nur ~~ohne~~ Rest

$$7/3 == 2$$

$$15/4 == 3$$

$$16/4 == 4$$

$$7\%3 == 1$$

$$15\%4 == 3$$

$$16\%4 == 0$$

Integer Division & Modulo

Integer Division

Der Compiler "ignoriert" Dezimalstellen, wenn er (**unsigned**) **int** durch (**unsigned**) **int** teilt.

Modulo

Division mit Rest, nur ~~ohne~~ Rest

$$7/3 == 2$$

$$15/4 == 3$$

$$16/4 == 4$$

$$7\%3 == 1$$

$$15\%4 == 3$$

$$16\%4 == 0$$

wichtige Identität

$$(a / b) * b + a \% b == a$$

always true

Fragen/Unklarheiten?

Mal schauen, was ihr gelernt habt

- Geht auf `expert.ethz.ch`
- Logt euch ein
- Geht zu “Code Examples”
- Unter “Lecture 2: Exercise Session”, öffnet “Last Three Digits”

Mal schauen, was ihr gelernt habt

- Geht auf `expert.ethz.ch`
- Logt euch ein
- Geht zu “Code Examples”
- Unter “Lecture 2: Exercise Session”, öffnet “Last Three Digits”
- Versucht die Aufgabe zu lösen (10 Minuten)
- Wir ~~schauen uns~~ eure Ansätze später ~~an~~
besprechen

Aufgabe

Task “Last Three Digits”

Write a program which reads in an integer **a** larger than 1000 and outputs its last three digits with a space between them.

For example, if **a** = 14325 , the output should be 3 2 5 .



Handwritten red annotations on the example output "3 2 5". A bracket is drawn above the digits "3 2 5". Below the digits, there are two double quotes "" and a small 'v' mark. A red arrow points from the word "spaces!" written below to the spaces between the digits.

4. Binärrepräsentation

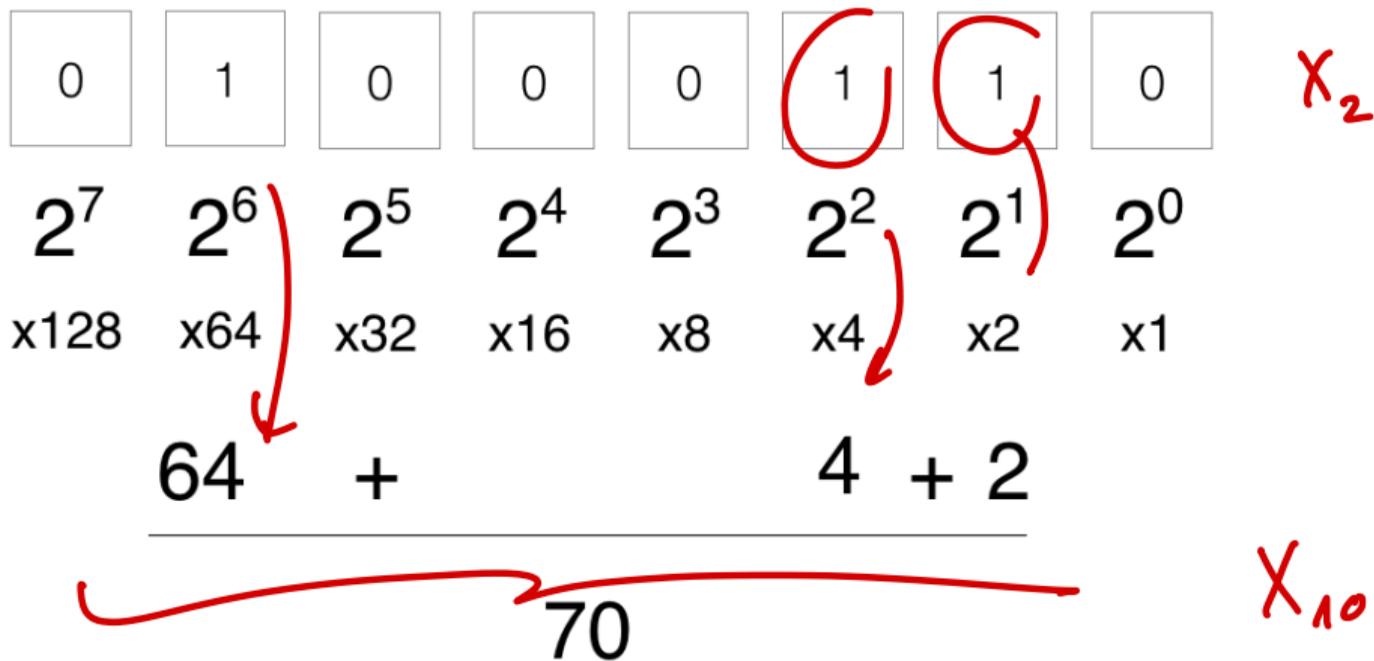
Binärrepräsentation

Binärrepräsentation

01000110

0	1	0	0	0	1	1	0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	$2^0 = 1$
x128	x64	x32	x16	x8	x4	x2	x1

Binärrepräsentation



Bahnhofsuhr

Bahnhofsuhr

Wie spät ist es?



Bahnhofsuhr

Wie spät ist es?



Source: bahnhofsuhrsg.ch

Bahnhofsuhr

Wie spät ist es?

0 1 2 3 ... 9 A -- F
| " --
16 15



X₁₆

Source: bahnhofsuhrsg.ch

07:20:40

Binärrepräsentation

Aufgabe

Wie sieht ein Algorithmus aus, der die Binärrepräsentation einer Dezimalzahl herausfindet?

Wie würde man beispielsweise bei der Dezimalzahl 61_{10} vorgehen, um sie in Binärrepräsentation zu bringen?

$61 / 2$	rest	
	1	1
$30 / 2$	0	0
$15 / 2$	1	1
\vdots	\vdots	\vdots
$1 / 2$	1	1

The diagram illustrates the division process for converting the decimal number 61 to binary. It shows a series of divisions by 2, with the remainder (rest) recorded on the right. The remainders are 1, 0, 1, and 1, which are then grouped by a bracket to show they form the binary digits 1011. The final remainder 1 is also grouped by a bracket to show it forms the binary digit 1.

Binärrepräsentation

Aufgabe

Wie sieht ein Algorithmus aus, der die Binärrepräsentation einer Dezimalzahl herausfindet?

Wie würde man beispielsweise bei der Dezimalzahl 61_{10} vorgehen, um sie in Binärrepräsentation zu bringen?

Lösung

Teile die Dezimaldarstellung durch 2 und behalte den Rest (wie eine Modulodivision). Teile dann die übrige Nummer nochmal und so weiter, bis man 0 erreicht.

Binärrepräsentation

$$0026 = 26$$

$$61 = 2 * 30 + 1$$

$$30 = 2 * 15 + 0$$

$$15 = 2 * 7 + 1$$

$$7 = 2 * 3 + 1$$

$$3 = 2 * 1 + 1$$

$$1 = 2 * 0 + 1$$

$$0 = \begin{matrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{matrix}$$

Binärrepräsentation

$$61 = 2 * 30 + 1$$

$$30 = 2 * 15 + 0$$

$$15 = 2 * 7 + 1$$

$$7 = 2 * 3 + 1$$

$$3 = 2 * 1 + 1$$

$$1 = 2 * 0 + 1$$



Dann die letzte Spalte von unten nach oben ablesen und fertig!

$$61_{10} = 111101_2$$

Binärrepräsentation von negativen Integer

Aufgabe

Finde einen Weg, um negative Integer abzuspeichern.

Tipp:

$$a + (-a) = 0$$

Binärrepräsentation von negativen Integer

Aufgabe

Finde einen Weg, um negative Integer abzuspeichern.

Tipp:

$$a + (-a) = 0$$

Lösung

Behandle die vorderste Ziffer einfach als das negative ihres eigentlichen Werts

Sehr gutes Video dazu

$$1111 = \frac{2^4 \cdot 1111 - 1111}{1111} = 1111 - 1$$

$$\begin{array}{r|l} 2^3 & 2^2 \ 2^1 \ 2^0 \\ 1 & 1 \ 0 \ 1 \\ \hline -8 & 4 \ 2 \ 1 \end{array}$$

siehe →



$$-8 + 4 + 1 = -5$$

int

siehe →

$$7 \rightarrow 2 + 1 = 011$$

unsigned int



Binärrepräsentation von negativen Integern

Eight-bit integers

Bits ↕	Unsigned value ↕	Signed value (Two's complement) ↕
0000 0000	0	0
0000 0001	1	1
0000 0010	2	2
0111 1110	126	126
0111 1111	127	127
1000 0000	128	-128
1000 0001	129	-127
1000 0010	130	-126
1111 1110	254	-2
1111 1111	255	-1

Die Anzahl darstellbarer Zahlen bleibt gleich, aber die Zahlen die sich darstellen lassen, hängt von der Wahl ab, ob es int oder unsigned int ist

Ranges: $[0, 255]$ unsigned int
 $[-128, 127]$ int

Binärrepräsentation von negativen Integern

-7 → int

Aufgabe

Wie erhält man die **signed int** Repräsentation einer beliebigen, negativen Ganzzahl $x < 0$?

Fragen/Unklarheiten?

signed int representation

nächste pos. $100 \rightarrow -4$
signed int $011 = 3$
zahl
für 3 ziffern

3 ziffern : $[-4, 3]$
↓ ↓
100 011

R vs. L Werte

a, b, const c
L

int a = 5;
L-Wert R-Wert
 ↓ ↓
 d 5

 a

5, 2...
R

std... a = 5
 5

5. Tipps zu `[code]`expert

Tipps für [code] expert

- Früh genug anfangen
- Zusammen daran arbeiten (aber nicht abschreiben!)

6. Outro

Allgemeine Fragen?

Bis zum nächsten Mal

Schöne Woche!