

Exercise Session

Week 02

Adel Gavranović

agavranovic@student.ethz.ch

Today's topics

▶ polybox for session material

▶ Mail to TA

Intro

Admin

Integer Division and Modulo

Binary Representation

Expressions

Outro

Safety First!

Masks must be worn properly at all times

- Over mouth and nose
- At all times
- I can and will kick you out of the session, if need be
- Please refrain from drinking during the session

Safety First!

Masken müssen jederzeit ordnungsgemäss getragen werden

- Über Mund und Nase
- Jederzeit
- Sollte es nötig sein, kann und werde ich Leute des Zimmers verweisen
- Bitte unterlasst das Trinken während der Sessions

Safety First!

In general

- If you feel unwell: **stay at home**
- You can e-mail me any day, at any time to ask questions

Im Allgemeinen

- Falls du dich krank fühlst: **bleib zuhause**
- Du kannst mir an jedem Tag, zu jeder Zeit eine E-mail mit fragen schicken

About your TA

- Started ETH with no prior coding experience

About your TA

- Started ETH with no prior coding experience
- Had a great mark in Informatik on the BPB-I

About your TA

- Started ETH with no prior coding experience
- Had a great mark in Informatik on the BPB-I
- Is going to make sure you're not going to fail the BPB-I because of Computer Science

About your TA

- Started ETH with no prior coding experience
- Had a great mark in Informatik on the BPB-I
- Is going to make sure you're not going to fail the BPB-I because of Computer Science
- Studies RW/CSE just like all of you

About your TA

- Started ETH with no prior coding experience
- Had a great mark in Informatik on the BPB-I
- Is going to make sure you're not going to fail the BPB-I because of Computer Science
- Studies RW/CSE just like all of you
- Likes coffee



About (all of) you

- What's your name?
- Where are you from?
- Why RW/CSE @ ETH?
- Do you have any prior coding experience?

Administrative Matters

:: open much better looking slides now ::

Questions?

About this session

- 45/10/45 (usually)

About this session

- 45/10/45 (usually)
- We'll discuss questions on last exercises at the beginning of the sessions

About this session

- 45/10/45 (usually)
- We'll discuss questions on last exercises at the beginning of the sessions
- You're allowed to **not** come to the sessions, even if you **do** feel fine

About this session

- 45/10/45 (usually)
- We'll discuss questions on last exercises at the beginning of the sessions
- You're allowed to **not** come to the sessions, even if you **do** feel fine
- I'll try to upload the slides to the polybox (link via e-mail) in advance and the annotated slides at the end of the day

About this session

- 45/10/45 (usually)
- We'll discuss questions on last exercises at the beginning of the sessions
- You're allowed to **not** come to the sessions, even if you **do** feel fine
- I'll try to upload the slides to the polybox (link via e-mail) in advance and the annotated slides at the end of the day
- **ASK QUESTIONS**
I might not know the answer, but I'll be able to point you to the solution (usually)

About this session

- 45/10/45 (usually)
- We'll discuss questions on last exercises at the beginning of the sessions
- You're allowed to **not** come to the sessions, even if you **do** feel fine
- I'll try to upload the slides to the polybox (link via e-mail) in advance and the annotated slides at the end of the day
- **ASK QUESTIONS**
I might not know the answer, but I'll be able to point you to the solution (usually)
- Feel free to put questions and comments in your code (with `//`) but keep in mind, that I'll only see them **after** submission/deadline

Questions?

About this course

- Primary goal: learn C++ and think more like a programmer

About this course

- Primary goal: learn C++ and think more like a programmer
- If you're going to fail BPB-I,
it probably won't be because of Informatik

About this course

- Primary goal: learn C++ and think more like a programmer
- If you're going to fail BPB-I,
it probably won't be because of Informatik
- **But** this is probably the most important course you'll take
in your first year, so study it well
(don't forget to study Analysis: you wouldn't be the first
ones to fail because of it)

About this course

- Primary goal: learn C++ and think more like a programmer
- If you're going to fail BPB-I,
it probably won't be because of Informatik
- **But** this is probably the most important course you'll take
in your first year, so study it well
(don't forget to study Analysis: you wouldn't be the first
ones to fail because of it)
- Exams will be in winter, but:
Study. During. The. Semester.

About this course

- Primary goal: learn C++ and think more like a programmer
- If you're going to fail BPB-I,
it probably won't be because of Informatik
- **But** this is probably the most important course you'll take
in your first year, so study it well
(don't forget to study Analysis: you wouldn't be the first
ones to fail because of it)
- Exams will be in winter, but:
Study. During. The. Semester.
- Try not to lose track, but if you do:
ask for help, asap!

About this course

- Primary goal: learn C++ and think more like a programmer
- If you're going to fail BPB-I, it probably won't be because of Informatik
- **But** this is probably the most important course you'll take in your first year, so study it well (don't forget to study Analysis: you wouldn't be the first ones to fail because of it)
- Exams will be in winter, but:
Study. During. The. Semester.
- Try not to lose track, but if you do:
ask for help, asap!
- Make use of the summaries on the course page

About this course

- Primary goal: learn C++ and think more like a programmer
- If you're going to fail BPB-I, it probably won't be because of Informatik
- **But** this is probably the most important course you'll take in your first year, so study it well (don't forget to study Analysis: you wouldn't be the first ones to fail because of it)
- Exams will be in winter, but:
Study. During. The. Semester.
- Try not to lose track, but if you do:
ask for help, asap!
- Make use of the summaries on the course page
- Do the weekly exercises

Intro
○○○○○○

Admin
○○○○●○○

Integer Division and Modulo
○○○○○

Binary Representation
○○○○○○○○

Expressions
○○○

Outro
○○

Questions?

How to [code]expert

open `expert.ethz.ch` on your device of choice

- [code]expert can be a little picky...
- Follow the instruction in the exercises as closely as possible (i.e. avoid unnecessary text)
- The autograder will do most of the grading

How to [code] expert

open `expert.ethz.ch` on your device of choice

- [code] expert can be a little picky...
- Follow the instruction in the exercises as closely as possible (i.e. avoid unnecessary text)
- The autograder will do most of the grading
- I won't ever give you more points than the autograder has given you

How to [code]expert

open `expert.ethz.ch` on your device of choice

- [code]expert can be a little picky...
- Follow the instruction in the exercises as closely as possible (i.e. avoid unnecessary text)
- The autograder will do most of the grading
- I won't ever give you more points than the autograder has given you
- I might have to deduct points, if the exercise is solved in a wrong way

How to [code]expert

open `expert.ethz.ch` on your device of choice

- [code]expert can be a little picky...
- Follow the instruction in the exercises as closely as possible (i.e. avoid unnecessary text)
- The autograder will do most of the grading
- I won't ever give you more points than the autograder has given you
- I might have to deduct points, if the exercise is solved in a wrong way
- For more information on [code]expert, check out the "Organizational Sheet" on the [polybox](#)

Questions?

The difference between = and ==

Assignment Operator =

Used for assigning values to variables

The difference between = and ==

Assignment Operator =

Used for assigning values to variables

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

The difference between = and ==

Assignment Operator =

Used for assigning values to variables

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

Equality operator ==

is used for checking equality between variables

The difference between = and ==

Assignment Operator =

Used for assigning values to variables

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

Equality operator ==

is used for checking equality between variables

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

Integer Division & Modulo

Integer Division & Modulo

Integer Division

The compiler ignores decimal places when dividing an (unsigned) int by another (unsigned) int

Integer Division & Modulo

Integer Division

The compiler ignores decimal places when dividing an (unsigned) int by another (unsigned) int

Modulo

Division with rest, but ignore the rest

Integer Division & Modulo

Integer Division

The compiler ignores decimal places when dividing an (unsigned) int by another (unsigned) int

Modulo

Division with rest, but ignore the rest

$7/3 ==$

Integer Division & Modulo

Integer Division

The compiler ignores decimal places when dividing an (unsigned) int by another (unsigned) int

Modulo

Division with rest, but ignore the rest

$$7/3 == 2$$

$$15/4 ==$$

Integer Division & Modulo

Integer Division

The compiler ignores decimal places when dividing an (unsigned) int by another (unsigned) int

Modulo

Division with rest, but ignore the rest

$$7/3 == 2$$

$$15/4 == 3$$

$$16/4 ==$$

Integer Division & Modulo

Integer Division

The compiler ignores decimal places when dividing an (unsigned) int by another (unsigned) int

Modulo

Division with rest, but ignore the rest

$$7/3 == 2$$

$$15/4 == 3$$

$$16/4 == 4$$

$$7\%3 ==$$

Integer Division & Modulo

Integer Division

The compiler ignores decimal places when dividing an (unsigned) int by another (unsigned) int

Modulo

Division with rest, but ignore the rest

$$7/3 == 2$$

$$15/4 == 3$$

$$16/4 == 4$$

$$7\%3 == 1$$

$$15\%4 ==$$

Integer Division & Modulo

Integer Division

The compiler ignores decimal places when dividing an (unsigned) int by another (unsigned) int

Modulo

Division with rest, but ignore the rest

$$7/3 == 2$$

$$15/4 == 3$$

$$16/4 == 4$$

$$7\%3 == 1$$

$$15\%4 == 3$$

$$16\%4 ==$$

Integer Division & Modulo

Integer Division

The compiler ignores decimal places when dividing an (unsigned) int by another (unsigned) int

Modulo

Division with rest, but ignore the rest

$$7/3 == 2$$

$$15/4 == 3$$

$$16/4 == 4$$

$$7\%3 == 1$$

$$15\%4 == 3$$

$$16\%4 == 0$$

Integer Division & Modulo

Integer Division

The compiler ignores decimal places when dividing an (unsigned) int by another (unsigned) int

Modulo

Division with rest, but ignore the rest

$$7/3 == 2$$

$$15/4 == 3$$

$$16/4 == 4$$

$$7\%3 == 1$$

$$15\%4 == 3$$

$$16\%4 == 0$$

%-Identity

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

Questions?

Let's see what you've learned!

- Go to [code]expert (expert.ethz.ch)
- Log into your account
- Go to "Code Examples"
- Under "Lecture 2", open ▶ "Last Three Digits"
- Give it a try! On paper or on your device (10 minutes)
- We'll discuss your approaches later

division op: $/$
 modulo op: $\%$

Task

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

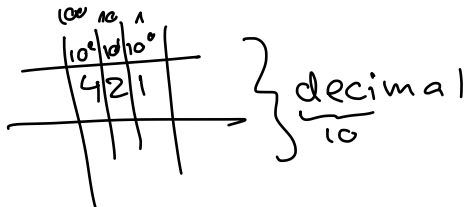
For example, if $a = 14325$, the output should be 3 2 5.

Questions?

Binary Representation

$$N_{16} = 0, \dots, 9, a \dots f; N$$

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

 N_2
 N_{10}


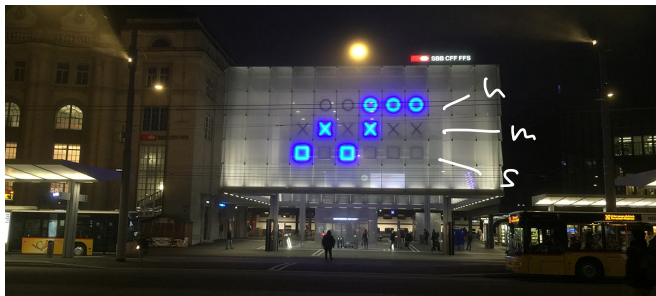
$$1 \cdot 1 + 2 \cdot 10 + 4 \cdot 100 = \underline{\underline{421}}$$

Binary Representation

0	1	0	0	0	1	1	0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
x128	x64	x32	x16	x8	x4	x2	x1
	64	+			4	+	2
<hr/>							
70							

Binary Representation

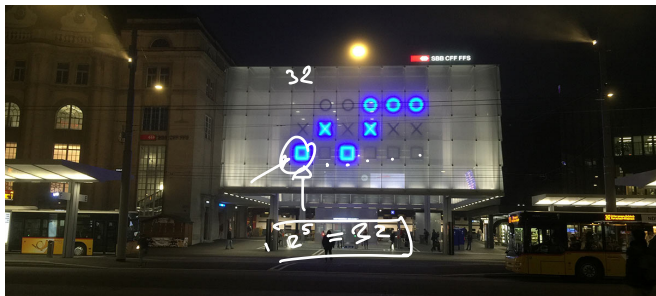
What time is it?



Source: bahnhofsuhrsg.ch

Binary Representation

What time is it?



Source: bahnhofsuhrsg.ch

07:20:40

Binary Representation

Task

What would be an algorithm that derives the binary representation of a decimal number?

For example, how would you convert 61 into binary?

$$N_1 \rightarrow N_2$$

Binary Representation

Task

What would be an algorithm that derives the binary representation of a decimal number?

For example, how would you convert 61 into binary?

Solution

Divide the decimal number by two and keep the rest (just like a modulo division). Divide the remaining number again, and so on and so forth until you reach 0.

Binary Representation

$$61 = 2 * 30 + 1$$

$$30 = 2 * 15 + 0$$

$$15 = 2 * 7 + 1$$

$$7 = 2 * 3 + 1$$

$$3 = 2 * 1 + 1$$

$$1 = 2 * 0 + 1$$

Binary Representation

$$61 = 2 * 30 + 1$$

$$30 = 2 * 15 + 0$$

$$15 = 2 * 7 + 1$$

$$7 = 2 * 3 + 1$$

$$3 = 2 * 1 + 1$$

$$1 = 2 * 0 + 1$$

Then read the last column from bottom to top and you're done!

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

Questions?

Binary Representation of Negative Integers

-6 $B_2 \rightarrow$ (010) $-$ $(2^4 \dots 2^0)$ 1010011
 $-8 + 0 + 2 + 0 = -6$

$$\begin{array}{r} -2^4 \quad -2^3 \quad -2^2 \quad -2^1 \quad -2^0 \\ 0110 \\ + 1010 \\ \hline 110000 \end{array}$$

Task

Come up with a way to store negative integers.

Hint: $a + (-a) = 0$.

1st Apr.

$\frac{1}{2} \dots$
 $+/-$
 $10000 \quad 2 \quad +0$
 $00000 \quad 2 \quad -0$

2nd Apr

111
 \vdots
 011
 010
 001
 $000 \rightarrow$ most neg number
 $0 = 100$

3rd Apr.

111	-1
110	-2
101	-3
100	-4
011	3
010	2
001	1
000	0

Binary Representation of Negative Integers

Task

Come up with a way to store negative integers.

Hint: $a + (-a) = 0$.

Solution

Just treat the front-most digit as negative his "usual" value.

▶ [excellent video on the topic](#)

Binary Representation of Negative Integers

Task

How to get the signed binary representation of any $x < 0$?

Binary Representation of Negative Integers

Task

How to get the signed binary representation of any $x < 0$?

Solution

1. Convert the absolute value of x to binary
2. Flip bits
3. Add 1

Handwritten solution for $x = -10$:

1. $10_{10} = 01010_2$ (with weights 16, 8, 4, 2, 1 above)
2. Flip bits to 10100_2 (with weights 16, 8, 4, 2, 1 below). This is equal to $-16 + 4 + 1 = -11$.
3. Add 1: $10100_2 + 1 = 10101_2$. This is equal to $-16 + 6 = -10$.

Questions?

Expressions

- C++-math is a lot like regular math, but not quite...

Expressions

- C++-math is a lot like regular math, but not quite...
- check `▸ cppreference` for precedence

Expressions

- C++-math is a lot like regular math, but not quite...
- check `▸ cppreference` for precedence
- evaluation from left to right

Expressions

- C++-math is a lot like regular math, but not quite...
- check `▸ cppreference` for precedence
- evaluation from left to right
- except then using (parenthesis), just like in math

Expressions

- C++-math is a lot like regular math, but not quite...
- check `▸ cppreference` for precedence
- evaluation from left to right
- except then using (parenthesis), just like in math
- expressions can be invalid

Expressions

- C++-math is a lot like regular math, but not quite...
- check `> cppreference` for precedence
- evaluation from left to right
- except then using (parenthesis), just like in math
- expressions can be invalid

l-value

- is on left site of =
- has an address in memory

r-value

- is on right site of =
- **no** address in memory

Expressions

Task

For the following expression, decide if they are

- valid
- an r - or l -value
- and what they evaluate to.

1. $1*(2*3)$

2. $(1=a)$

3. $(1$

4. $(a*3)=(b*5)$

Expressions

Task

For the following expression, decide if they are

- valid
- an r- or l-value
- and what they evaluate to.

1. $1*(2*3)$

2. $(1=a)$

3. $(1$

4. $(a*3)=(b*5)$

Solutions

1. valid, r-value, = 6

Expressions

Task

For the following expression, decide if they are

- valid
- an r- or l-value
- and what they evaluate to.

1. $1*(2*3)$
2. $(1=a)$
3. $(1$
4. $(a*3)=(b*5)$

Solutions

1. valid, r-value, = 6
2. valid, l-value, = a = 1

Expressions

Task

For the following expression, decide if they are

- valid
- an r- or l-value
- and what they evaluate to.

1. $1*(2*3)$
2. $(1=a)$
3. $(1$
4. $(a*3)=(b*5)$

Solutions

1. valid, r-value, = 6
2. valid, l-value, = a = 1
3. invalid, no closing parenthesis

Expressions

Task

For the following expression, decide if they are

- valid
- an r- or l-value
- and what they evaluate to.

1. $1*(2*3)$
2. $(1=a)$
3. $(1$
4. $(a*3)=(b*5)$

Solutions

1. valid, r-value, = 6
2. valid, l-value, = a = 1
3. invalid, no closing parenthesis
4. invalid, because of r-value on left side of =

Questions?

Outro

- Do the C++-tutorial if you haven't done it yet
- Do your weekly exercises
- Stay healthy

Final Questions