Exercise 2 – Expressions & Integers

Informatik I für Mathematiker und Physiker (HS 2015)

Yeara Kozlov

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

inf | Informatik
Computer Science

Agenda

- Judge system
- Integer division and modulo
- Expressions
- Binary representation
- Two's Complement Representation *
- HW #2

Agenda

- **Judge system**
- Integer division and modulo
- Expressions
- Binary representation
- Two's Complement Representation *
- HW #2

Judge

- Missing usernames - are you registered?
- Compiler errors
- Questions about last week's homework?

Agenda

- Judge system
- **Integer division and modulo**
- Expressions
- Binary representation
- Two's Complement Representation *
- HW #2

Integer types

- Integers store whole numbers
- Memory size may vary, because they depend on the implementation and Hardware.

Data-type	Function	
int	Stores integer numbers	
unsigned int (uint)	Stores non-negative integer numbers	

Integer Division /

- Dividing between two int or unsigned int variables discards the remainder (decimal digits)
- The resulting type of such a division is always an integer
- Rational parts of the division will be truncated
- Examples:

```
5 / 2 → 2    //The .5 is truncated
1 / 2 → 0    //Same reason here
```

(Integer) Modulo %

- The remainder of a division operation can be computed using the modulo operator
- a % b returns the remainder of a divided by b.
- The modulo operation is defined only for integers.
- Examples:

```
5 % 2 → 1
4 % 4 → 0
2 % 3 → 2
```

Integer Division and Modulo

- Use modulo to find if a number is even or odd.
  - how?
- Or if it's a leap year
  - how?
- The original number can be expressed in terms of these operators:

```
(15 % 4)*4 + 15 % 4 → 15
```

Division and Modulo - Example

- Schreibe ein Programm, welches zwei ganze Zahlen a und b einliesst, dann den Quotienten a/b als gemischten Ausdruck berechnet und zurückgibt. Zum Beispiel gibt es für a = 17 und b = 6 den gemischten Ausdruck 2 5/6 zurück.

- Write a program which reads in two integers a and b, then calculates the quotient as a / b mixed expression and outputs it. For example, if a = 17 and b = 6, the output should be 2 5/6.

Division and Modulo - Example

```
// Program: division.cpp

#include <iostream>

int main()
{
    // input




    // computation


    // output


    return 0;
}
```

Agenda

- Judge system
- Integer division and modulo
- **Expressions**
- Binary representation
- Two's Complement Representation *
- HW #2

Expressions and arithmetic operations

- There are 5 types of different arithmetic operators: +, -,* , / and %
- Order of evaluation:
  - Brackets before operators
  - Division, multiplication and modulo before minus and plus
- If two operators have the same precedence level, expression is usually evaluated *left to right*.
- Use brackets!
- http://en.cppreference.com/w/cpp/language/operator_precedence

Arithmetic operations

- Examples

```
5 * 4u + 1;          //21
5 * (4u + 1);        //25
2 * 3 + 3 * 4 / 2;   //12
9 % 4 + 1;           //2
9 % (4 + 1);         //4
5 * 3 % 4 * 2;       //6
5 * (3 % 4) * 2;     //30
```

Order of evaluation - Exceptions

- What happens here?

```
b = ++a + a++;
```

- The order of some expressions is not guaranteed!
- When in doubt be explicit.

Expressions - lvalue and rvalue

- An expression is a program statement that yields a value.
- All expressions are either *rvalues* or *lvalues*.
- *lvalues* persist behind a single operation.
- *rvalues* persist only for a single operation - no allocated storage.

- The operator = assigns the rvalue to the lvalue.
  - lvalue = rvalue

Expressions - lvalue and rvalue

- Arithmetic operations can be used to combine rvalues:
  - [rvalue ×] rvalue → rvalue
  - lvalue = [[rvalue ×] rvalue]
- lvalues can be used on the right hand side, but rvalues cannot be placed to the left of an assignment.

Class Exercise

- Are the following expressions correct?
- If so, what are the types (l-value, r-value)
- What are their values?

```
1*(2*3)              //R-value, 6
(a=1)                //L-value, 1
(1                   //Parenthesis not closed
(a*3)=(b*5)          //Left of '=' must be an L-value
127%7*67%17%1        //R-value, 0 (since x % 1 always 0)
```

Class Exercise

```
127 % 7 * 67 % 17 % k;
(((127 % 7) * 67) % 17) % k;
((1 * 67) % 17) % k;
(67 % 17) % k;
16 % k;

k = 1: final answer is 0;
```

Agenda

- Judge system
- Integer division and modulo
- Expressions
- **Binary representation**
- Two's Complement Representation *
- HW #2

Binary Representation

- The memory is binary:

5 -> 101 4 -> 100 12 -> 1100

- Larger Numbers need many bits:

32,767 -> 01111111 11111111

16 bits = 2 bytes

Binary Representation

- Conversion into binary:

61	2 * 30 + 1
30	2 * 15 + 0
15	2 * 7 + 1
7	2 * 3 + 1
3	2 * 1 + 1
1	2 * 0 + 1
	111101

- And vice versa:

1	1	1	1	0	1	
32	16	8	4	2	1	61

Binary Representation

- Tables for 4bit integers:

bin	uint	int
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7

bin	uint	int
1000	8	-8
1001	9	-7
1010	10	-6
1011	11	-5
1100	12	-4
1101	13	-3
1110	14	-2
1111	15	-1

Hexadecimal Representation

- … and how to count them.

hexadecimal:	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**a**	**b**	**c**	**d**	**e**	**f**	**10**	**11**	**12**	**13**
decimal:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

hexadecimal:	**14**	**15**	**16**	**17**	**18**	**19**	**1a**	**1b**	**1c**	**1d**	**1e**	**1f**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**…**
decimal:	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	...

Binary Representation Example

- Schreibe ein Programm threebin.cpp, welches eine Dezimalzahl n!= 0 vom Standard-Input einliest und die letzten drei bits von ns Binärdarstellung ausgibt. Wenn die Binärdarstellung weniger als drei Bits umfasst, fülle vorne mit Nullen auf.
- Write a program threebin.cpp that reads a (decimal) number n≠ 0 from standard input and outputs the last three bits of n's binary representation. Fill up with leading zeros in case the binary representation has less than three bits.

Binary Representation Example

- Getting the last three digits:

61	2 * 30 + 1	n % 2
30	2 * 15 + 0	n / 2 % 2
15	2 * 7 + 1	n / 4 % 2

- Let's program this…

Binary Representation Example

```
// Program: threebin.cpp

#include <iostream>

int main()
{
    // input
    std::cout << "Last three binary digits of n =? ";
    unsigned int n;
    std::cin >> n;

    // computation and output
    std::cout << "The digits are "
              << n / 2 / 2 % 2 << n / 2 % 2 << n % 2 << ".\n";

    return 0;
}
```

Agenda

- Judge system
- Integer division and modulo
- Expressions
- Binary representation
- **Two's Complement Representation ***
- HW #2

Two's Complement Representation

- Motivation: 1 + (-1) = 0
  - in 4 bits:

$$\begin{array}{r} 1111 \\ +0001 \\ \hline (1)0000 \end{array}$$

  - The (1) gets deleted - overflow
  - called 2's complement!
- Conversion to negative values:

$$+1 \xrightarrow{1.)} 0001 \xrightarrow{2.)} 1110 \xrightarrow{3.)} 1111$$

Two's Complement Representation

- 4 bits allow to store:
  - unsigned int 0 -> 15
  - int -8 -> 7

bin	uint	bin	uint
0000	0	1000	8
0001	1	1001	9
0010	2	1010	10
0011	3	1011	11
0100	4	1100	12
0101	5	1101	13
0110	6	1110	14
0111	7	1111	15

Two's Complement Representation

- How do you represent 9 with a 4 bit int?

bin	uint	int	bin	uint	int
0000	0	0	1000	8	-8
0001	1	1	1001	9	-7
0010	2	2	1010	10	-6
0011	3	3	1011	11	-5
0100	4	4	1100	12	-4
0101	5	5	1101	13	-3
0110	6	6	1110	14	-2
0111	7	7	1111	15	-1

unsigned int range overflow

For `unsigned int (4 bit)`

```
1111
1111+1 = 10000 %(2^4) -> 0000(0 dec)
10000+1 = 10001 %(2^4) -> 0001 (1 dec)
10001+1 = 10002 %(2^4) -> 0010 (2 dec)
```

Agenda

- Judge system
- Integer division and modulo
- Expressions
- Binary representation
- Two's Complement Representation *
- **HW #2**

Homework #2 Submission

- Exercise 2
- Submit answers to Q 1-2 to:
  - yeara.kozlov@inf.ethz.ch
  - Subject: [IFMP15] Ex 2
  - (Automatic filter!)
- Homework cycles - one week for submission, one week for grading.