SFU SIMON FRASER UNIVERSITY
ENGAGING THE WORLD

CMPT 120: Introduction to Computing Science and Programming 1

Data Representation: Unsigned and Signed Integers

python™

Reminders

One-Stop Access To Course Information

- **Course website: One-stop access** to all course information.

  http://www2.cs.sfu.ca/CourseCentral/120/liaqata/WebSite/index.html

- Course Outline
- Exam Schedule
- Python Info
- CourSys/Canvas link
- Learning Outcomes
- Office Hours
- Textbook links
- and more…
- Grading Scheme
- Lab/Tutorial Info
- Assignments

- **Canvas**: Discussions forum - https://canvas.sfu.ca/courses/39187
- **CourSys**: Assignments submission, grades - www.coursys.sfu.ca

How to Learn in This Course?

A **Attend** Lectures & Labs

R **Read** / review Textbook/Slides/Notes

R **Reflect** and ask Questions

O **Organize** – your learning activities on weekly basis,

and finally…

W **Write** Code, Write Code, and Write Code.

Deliverables

1. Deliverables are due by the given date and time.
2. For the course, we are using IDLE to write and run our Python code.
3. You can use the CSIL lab computers outside your lab hours.
4. Plan ahead your assignments and other deliverables. Computer crash, network problems etc. are not acceptable excuses for delays in deliverables.
5. You may use online Python interpreters for running and testing your codes, such as:

   https://repl.it/languages/Python3

Labs

1. Each lab has an assigned TA.
2. Attend your assigned lab and show your work to your TA for the participation marks.
3. Class enrolments and lab swaps are closed now.

Course Topics

1. General introduction
2. Algorithms, flow charts and pseudocode
3. Procedural programming in Python
4. Data types and Control Structures
5. Fundamental algorithms
6. Binary encodings
7. Basics of computability and complexity
8. Basics of Recursion
9. Subject to time availability:
   - Basics of Data File management

Today's Topics

- Data Representation (Binary Encoding)
  1. Unsigned Integer
  2. Signed Integer
  3. Binary Addition
  4. 1's Compliment Representation
  5. 2's Compliment Representation

1

Data Representation

Data Representation

- You type '**Li**' from a keyboard. How will computer store it in RAM as an ASCII code?.

  0 1 0 0 1 1 0 0 0 1 1 0 1 0 0 1

- How '**24**' will be stored in ASCII?

  0 0 1 1 0 0 1 0 0 0 1 1 0 1 0 0

- How about '**+ 524**'?

  0 0 1 0 1 0 1 1 0 0 1 1 0 1 0 1 0 0 1 1 0 0 1 0 0 0 1 1 0 1 0 0

- Numbers are represented as distinct ASCII codes, not as a single numeric value.
- Data stored in ASCII codes is **not good** for arithmetic operations (addition, subtraction etc.).
- Then, what if the numbers we type are integer data that we would like use in arithmetic operation? For example, when you type in Python: **marks = 12**
- Use a different representation structure for numbers i.e., store them differently (not ASCII).

Unsigned Integer Data Representation: Binary

- So, how data is stored inside computer when you write a statement: **marks = 12**
- The value 12 is an unsigned integer.
- One way is to store 12 in its binary equivalent form.
- 12 in binary is 0000 1100.
- If computer uses 8 bits to store an unsigned integer, then 12 would be stored as:

0	0	0	0	1	1	0	0

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

For a positive integer represented by **N** binary digits the possible values are **0 <= value <= 2^N -1.**

Signed Integer Data Representation: Binary

- A **signed integer**: For a positive integer represented by N binary digits the possible values are $-2^{N-1}-1 <= value <= 2^{N-1}-1$.

Sign bit	N -1 Binary Digits

	1	1	1	1	1	1	1
+/-	2^6	2^5	2^4	2^3	2^2	2^1	2^0
+/- 127	64	32	16	8	4	2	1
+12	0	0	0	0	1	1	0	0
-12	1	0	0	0	1	1	0	0

Signed Integer Data Representation: Problems

- Which value should we use to denote the positive or negative sign: 1 or 0?
- More than one value of 0:
  - **+0**
  - **-0**

0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0

- Creates difficulty in defining addition and subtraction on typical computer hardware for either choice.

Adapted from: Janice Regan, 2013.

Signed Integer Data Representation: Problems Example

- Let's see a simple addition problem.

	1 1 0 1	(-5)
	0 0 1 0	(2)
ADD or OR:	1 1 1 1	(-7)
	1 0 0 1	(-1)
	0 0 1 1	(3)
AND	1 1 0 0	(-4)

Adapted from: Janice Regan, 2013.

Signed Integer Data Representation: One's Complement

- Integer is represented by a string of **binary** digits.
  - But, is represented in 1's compliment form.

Sign bit	N -1 Binary Digits: 1's Compliment

- **How a number is converted to its 1's Compliment form:**
  1. If a number is positive, simply convert the number to its binary equivalent.
     - For example, if the number is: **6** **0 0 0 0 0 1 1 0**
  2. If a number is negative, **convert** the number to its binary equivalent and **flip** the bits.
     - For example , if the number is: -6 **0 0 0 0 0 1 1 0**
     - Flip the bits: **1 1 1 1 1 0 0 1**

Signed Integer Data Representation: One's Complement

- Suppose an 8-bit 1's pattern is shown as: **1 0 1 1 0 0 0 1**

- **What number this pattern represents?**
  - If first bit 0, then it is an unsigned/positive number, as shown (simply convert it to its decimal equivalent).
  - If first bit is 1, then:
    1. Flip all the bits. So, **1011 0001** becomes **0100 1110**
    2. Convert to decimal: 01001110 $= 2^6 + 2^3 + 2^2 + 2^1 = 64 + 8 + 4 + 2 = 78$
    3. Add a minus sign. So **10110001** represents **-78** in one's Complement form.

Adapted from: Janice Regan, 2013.

One's Complement Advantage and Disadvantage

- Advantage:
  - Addition is now more efficient.

- Disadvantage:
  - There are still two representations of 0
    - **+0 0000 0000**
    - **- 0 1111 1111**

Adapted from: Janice Regan, 2013.

Examples: One's complement

-84

1	0	1	0	1	0	1	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	1	0	1	0	1	0	0
-	64	0	16	0	4	0	0

35

0	0	1	0	0	0	1	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	0	1	0	0	0	1	1
	0	32	0	0	0	2	1

Remember if first digit is1 flip bits.

Adapted from: Janice Regan, 2013.

Examples: One's complement

-50

1	1	0	0	1	1	0	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	0	1	1	0	0	1	0
-	0	32	16	0	0	2	0

79

0	1	0	0	1	1	1	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	1	0	0	1	1	1	1
	64	0	0	8	4	2	1

Remember if first digit is 1 flip bits.

Adapted from: Janice Regan, 2013.

Decimal to 1s complement

- -49 (number < 0)
  - Express 49 in 8 bit binary
    - 32+16+1
    - 00110001
  - Flip the bits
    - 11001110

Adapted from: Janice Regan, 2013.

Decimal to 1s Complement

- 111 (number > 0)
  - Express 111 in 8 bit binary
    - 64 + 32+ 8+ 4 + 21
    - 0110 1111

Your turn

- 1010 1010
- 120
- 1101 1010
- -59

Adapted from: Janice Regan, 2013.

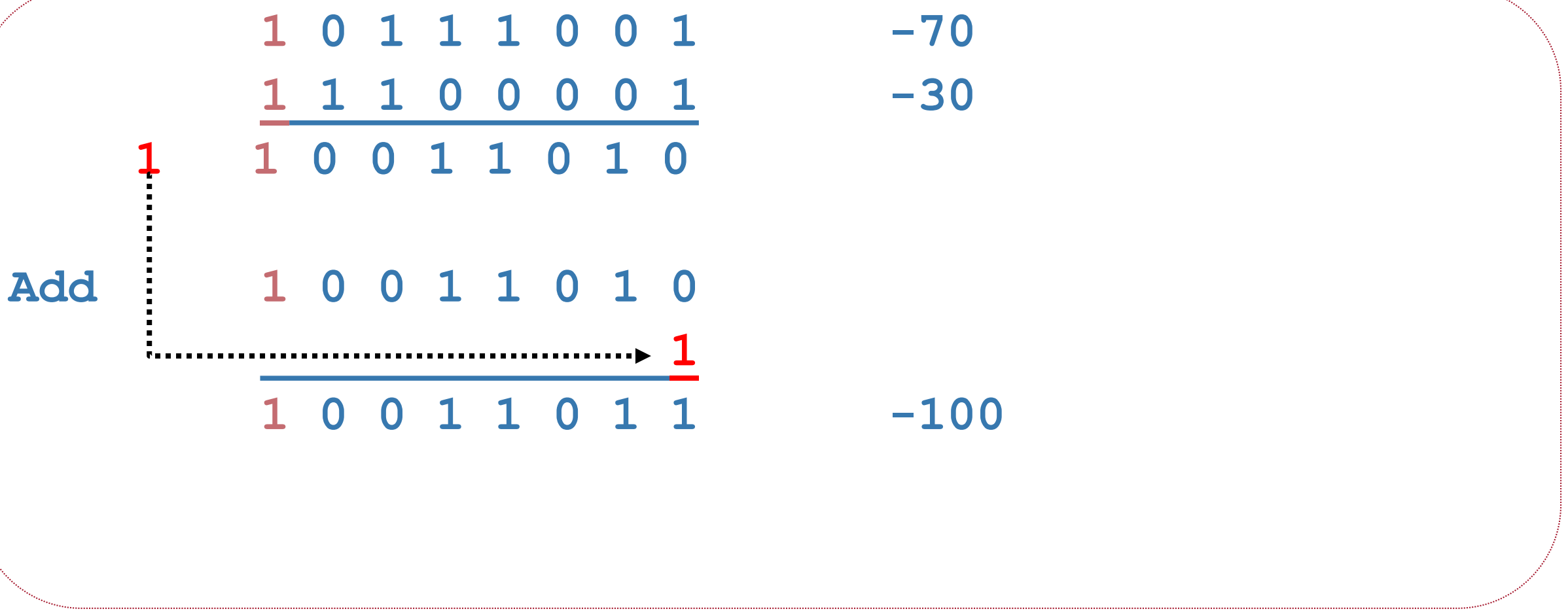
One's Complement Addition

```
 0  0  1  1   1  0  0  1        57
 0  1  0  0   0  0  0  1        65
-----------------------------------
 0  1  1  1   1  0  1  0       122

 0  0  1  1   1  0  1  1        59
 1  0  0  0   0  1  1  1      -120
-----------------------------------
 1  1  0  0   0  0  1  0       -61
```

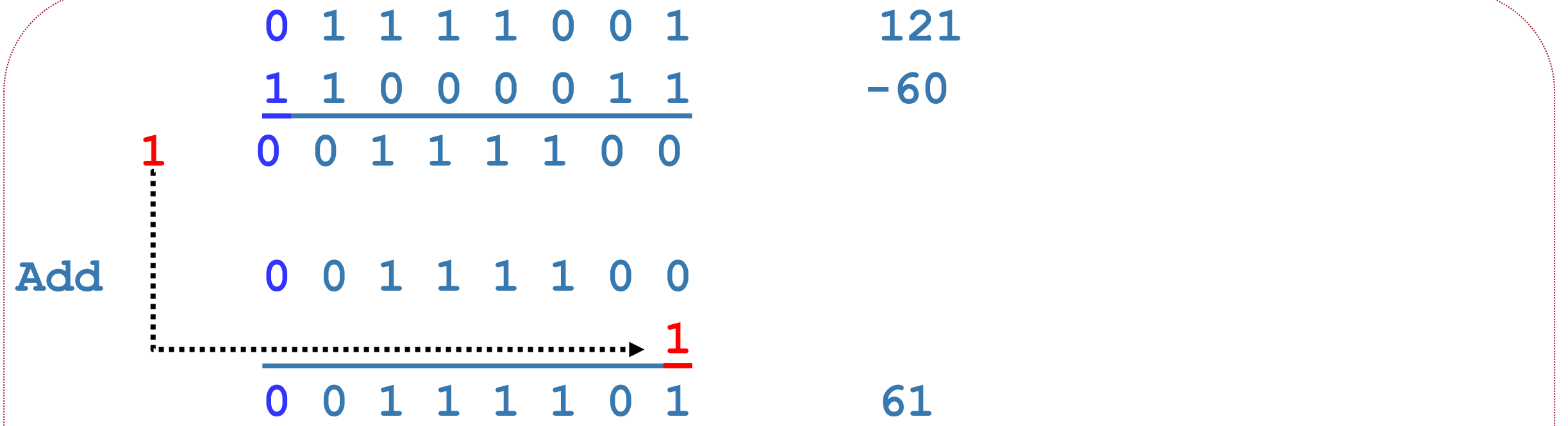
Adapted from: Janice Regan, 2013.

Ones complement addition



Adapted from: Janice Regan, 2013.

Ones complement addition



Adapted from: Janice Regan, 2013.

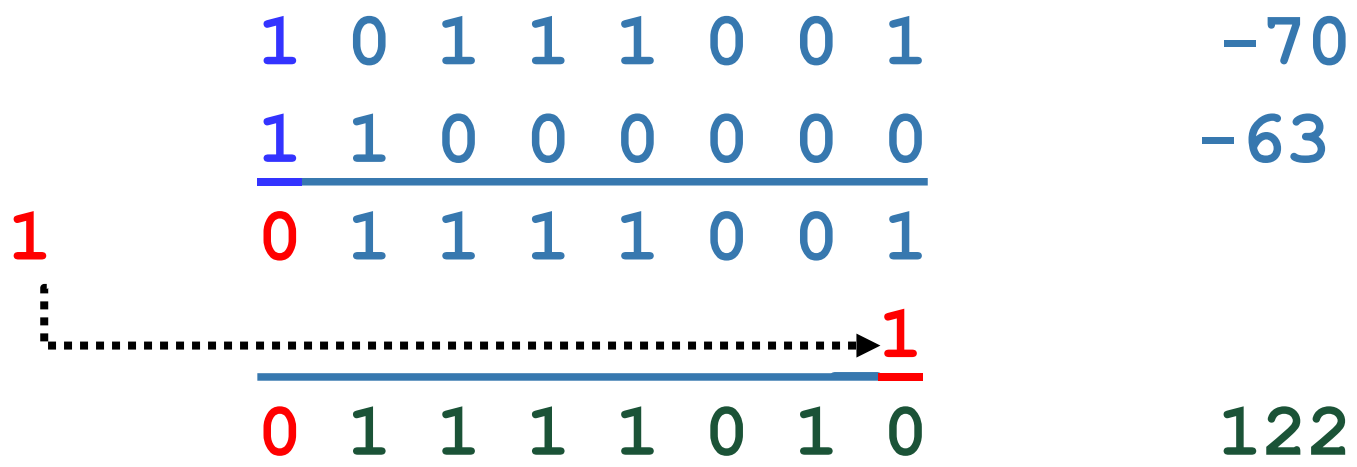
Problems? overflow

```
  0  1  1  1  1  0  0  1        121
  0  1  0  0  0  0  0  0         64
 ------------------------
  1  0  1  1  1  1  0  0        -67      ????
```

- 121 + 64 = 185
- Largest integer that can be represented is 127

Adapted from: Janice Regan, 2013.

Problems? overflow



- -70 + -63 = -133
- Smallest integer that can be represented is -127

Adapted from: Janice Regan, 2013.

Your Turn

- Compute 1's compliment binary addition and post your solution on the Canvas by tonight 11:59pm.

  -59 + 12

Adapted from: Janice Regan, 2013.

Questions?