Outline

1. Computer Architecture
   - Hardware Components
   - Programming Environments

2. Getting Started with Python
   - Installing Python
   - Executing Python code

3. Number Systems
   - Decimal and Binary notations

4. Running Sage

5. Summary + Assignments

MCS 260 Lecture 2
Introduction to Computer Science
Jan Verschelde, 27 August 2008
A computer system consists of

1. **Hardware**: physical components of computer
   - computer: processor, memory, bus, ...
   - peripherals: printer, screen, keyboard, mouse, ...

2. **Software**: programs executed by computer
   - basic software like the Operating System (OS)
     either Unix (e.g.: Solaris, GNU-Linux, Mac OS X)
     or Windows (the OS of Microsoft)
   - application software such as IDLE, Sage, ...
     application software needs operating system to run
A computer system consists of

1. **Hardware**: physical components of computer
   - computer: processor, memory, bus, ...
   - peripherals: printer, screen, keyboard, mouse, ...

2. **Software**: programs executed by computer
   - basic software like the Operating System (OS)
     either Unix (e.g.: Solaris, GNU-Linux, Mac OS X)
     or Windows (the OS of Microsoft)
   - application software such as IDLE, Sage, ...
     application software needs operating system to run
Computer Architecture

1. Computer Architecture
   Hardware Components
   Programming Environments

2. Getting Started with Python
   Installing Python
   Executing Python code

3. Number Systems
   Decimal and Binary notations

4. Running Sage

5. Summary + Assignments
Hardware Components

**processor** (or CPU: Central Processing Unit) does the computing and coordinates data transfer

**memory** (or RAM: Random Access Memory) is used to store data and programs, of limited capacity and volatile (lost if power off)

**storage** persistently stores large quantities of data and programs, slower access to storage than to memory, but larger than RAM

**peripherals** are used to communicate with computer

**system bus** connects CPU, RAM, storage, and peripherals
Hardware Components

**processor** (or CPU: Central Processing Unit) does the computing and coordinates data transfer.

**memory** (or RAM: Random Access Memory) is used to store data and programs, of limited capacity and volatile (lost if power off).

**storage** persistently stores large quantities of data and programs, slower access to storage than to memory, but larger than RAM.

**peripherals** are used to communicate with computer.

**system bus** connects CPU, RAM, storage, and peripherals.
Hardware Components

**processor** (or CPU: Central Processing Unit) does the computing and coordinates data transfer

**memory** (or RAM: Random Access Memory) is used to store data and programs, of limited capacity and volatile (lost if power off)

**storage** persistently stores large quantities of data and programs, slower access to storage than to memory, but larger than RAM

**peripherals** are used to communicate with computer

**system bus** connects CPU, RAM, storage, and peripherals
Hardware Components

**processor** (or CPU: Central Processing Unit) does the computing and coordinates data transfer

**memory** (or RAM: Random Access Memory) is used to store data and programs, of limited capacity and volatile (lost if power off)

**storage** persistently stores large quantities of data and programs, slower access to storage than to memory, but larger than RAM

**peripherals** are used to communicate with computer

**system bus** connects CPU, RAM, storage, and peripherals
Hardware Components

**processor** (or CPU: Central Processing Unit) does the computing and coordinates data transfer.

**memory** (or RAM: Random Access Memory) is used to store data and programs, of limited capacity and volatile (lost if power off).

**storage** persistently stores large quantities of data and programs, slower access to storage than to memory, but larger than RAM.

**peripherals** are used to communicate with computer.

**system bus** connects CPU, RAM, storage, and peripherals.
Computer Architecture

1. Computer Architecture
   Hardware Components
   Programming Environments

2. Getting Started with Python
   Installing Python
   Executing Python code

3. Number Systems
   Decimal and Binary notations

4. Running Sage

5. Summary + Assignments
Programming Environments
what it takes to run programs

editor: is used to write source code

compiler: translates source code into an object, an executable program — if code is bug free

interpreter: executes high level code directly

linker: combines several objects into one single executable program

debugger: helps user to locate bugs, allowing a stepwise execution of the program

use an IDE: Integrated Development Environment

Pythons IDE is called IDLE
Programming Environments
what it takes to run programs

**editor**: is used to write source code

**compiler**: translates source code into an object, an executable program — if code is bug free

**interpreter**: executes high level code directly

**linker**: combines several objects into one single executable program

**debugger**: helps user to locate bugs, allowing a stepwise execution of the program

use an IDE: Integrated Development Environment
Python's IDE is called IDLE
Programming Environments
what it takes to run programs

**editor** : is used to write source code

**compiler** : translates source code into an object, an executable program — if code is bug free

**interpreter** : executes high level code directly

**linker** : combines several objects into one single executable program

**debugger** : helps user to locate bugs, allowing a stepwise execution of the program

use an IDE: Integrated Development Environment
Python's IDE is called IDLE
Programming Environments
what it takes to run programs

**editor** : is used to write source code

**compiler** : translates source code into an object, an executable program — if code is bug free

**interpreter** : executes high level code directly

**linker** : combines several objects into one single executable program

**debugger** : helps user to locate bugs, allowing a stepwise execution of the program

use an IDE: Integrated Development Environment

Pythons IDE is called IDLE
Programming Environments
what it takes to run programs

**editor** : is used to write source code

**compiler** : translates source code into an object, an executable program — if code is bug free

**interpreter** : executes high level code directly

**linker** : combines several objects into one single executable program

**debugger** : helps user to locate bugs, allowing a stepwise execution of the program

use an IDE: Integrated Development Environment
Pythons IDE is called IDLE
Programming Environments
what it takes to run programs

editor : is used to write source code

compiler : translates source code into an object, an executable program — if code is bug free

interpreter : executes high level code directly

linker : combines several objects into one single executable program

debugger : helps user to locate bugs, allowing a stepwise execution of the program

use an IDE: Integrated Development Environment
Pythons IDE is called IDLE
Executing Programs
how programs are executed

- high level programming languages are oriented towards the convenience of the programmer
- an assembler language offers symbols to the basic instructions for writing machine code

The Python Virtual Machine:

The Python interpreter creates bytecode that is then executed by the Python Virtual Machine at runtime.
Computer Architecture

first steps with Python

1. Computer Architecture
   Hardware Components
   Programming Environments

2. Getting Started with Python
   Installing Python
   Executing Python code

3. Number Systems
   Decimal and Binary notations

4. Running Sage

5. Summary + Assignments
Free to download from www.python.org.

**Unix**  most Linux distributions have Python installed, or else contact your system administrator. Login to icarus.cc.uic.edu using your netid.

**Mac OS X**  like with unix you can dowload the source or run Python 2.5.2 for Macintosh OS X (universal installer both for PPC and IntelMacs). Computers in SEL 2263 have Python installed.

**Windows**  run the Python 2.5.2 windows installer. Most labs on campus have Python installed.
Computer Architecture

first steps with Python

1. Computer Architecture
   Hardware Components
   Programming Environments

2. Getting Started with Python
   Installing Python
   Executing Python code

3. Number Systems
   Decimal and Binary notations

4. Running Sage

5. Summary + Assignments
Executing Python code
program prints "hello world!"

Ways to run python programs:

1. In a Python session, type commands at the prompt:
   >>> print 'hello world!'

2. Running programs at the command prompt:
   1. Save Python commands in a file, e.g.: hello.py.
   2. Type `python hello.py` at the command prompt.

3. On windows, double click a file with .py extension.

4. In IDLE, go to the `run` menu in the editor.
Executing Python code
program prints "hello world!"

Ways to run python programs:

1. In a Python session, type commands at the prompt:
   >>> print 'hello world!'

2. Running programs at the command prompt:
   1. Save Python commands in a file, e.g.: hello.py.
   2. Type `python hello.py` at the command prompt.

3. On windows, double click a file with .py extension.

4. In IDLE, go to the `run` menu in the editor.
Executing Python code
program prints "hello world!"

Ways to run python programs:

1. In a Python session, type commands at the prompt:

   >>> print 'hello world!'

2. Running programs at the command prompt:
   1. Save Python commands in a file, e.g.: hello.py.
   2. Type `python hello.py` at the command prompt.

3. On windows, double click a file with .py extension.

4. In IDLE, go to the run menu in the editor.
Executing Python code
program prints "hello world!"

Ways to run python programs:

1. In a Python session, type commands at the prompt:
   ```python
   >>> print 'hello world!'
   ```

2. Running programs at the command prompt:
   1. Save Python commands in a file, e.g.: `hello.py`.
   2. Type `python hello.py` at the command prompt.

3. On windows, double click a file with `.py` extension.

4. In IDLE, go to the `run` menu in the editor.
Executing Python code
program prints "hello world!"

Ways to run python programs:

1. In a Python session, type commands at the prompt:
   >>> print 'hello world!'

2. Running programs at the command prompt:
   1. Save Python commands in a file, e.g.: hello.py.
   2. Type python hello.py at the command prompt.

3. On windows, double click a file with .py extension.

4. In IDLE, go to the run menu in the editor.
Executing Python code
program prints "hello world!"

Ways to run python programs:

1. In a Python session, type commands at the prompt:
   ```
   >>> print 'hello world!'  
   ```

2. Running programs at the command prompt:
   1. Save Python commands in a file, e.g.: `hello.py`.
   2. Type `python hello.py` at the command prompt.

3. On windows, double click a file with `.py` extension.

4. In IDLE, go to the `run` menu in the editor.
Interactive Python code

Let us write a program that asks for our name, as input. And then, as output, writes hello followed by our name.

```python
input  The raw_input() function accepts only text input:

>>> name = raw_input('Who\'s there ? ')

Displays Who\’s there ? on screen and assigns what the user types in to the variable name.

output  With name in its argument, the print command displays the value of name:

>>> print 'hello ', name , '!

develop Python code interactively at the prompt
Interactive Python code

Let us write a program that asks for our name, as input. And then, as output, writes hello followed by our name.

```python
input  The `raw_input()` function accepts only text input:

```>>> name = raw_input('Who\’s there ? ')

Displays *Who\’s there ?* on screen and assigns what the user types in to the variable `name`.

```python
output  With `name` in its argument, the `print` command displays the value of `name`:

```>>> print 'hello ', name , '!

develop Python code interactively at the prompt.
Interactive Python code

Let us write a program that asks for our name, as input. And then, as output, writes hello followed by our name.

```
input The `raw_input()` function accepts only text input:

```python
>>> name = raw_input('Who\'s there ? ')
```

Displays Who’s there ? on screen and assigns what the user types in to the variable name.

```
output With name in its argument, the `print` command displays the value of name:

```python
>>> print 'hello ', name , '!
```

develop Python code interactively at the prompt
Interactive Python code

Let us write a program that asks for our name, as input. And then, as output, writes hello followed by our name.

```
input  The `raw_input()` function accepts only text input:

```input
>>> name = raw_input('Who\'s there ? ')
```

Displays `Who\’s there ?` on screen and assigns what the user types in to the variable `name`.

```
output  With `name` in its argument, the `print` command displays the value of `name`:

```
>>> print 'hello ', name , '!
```

`develop Python code interactively at the prompt`
our first interactive program
running at the command prompt

The file `hello_there.py` contains

```
# L-2 MCS 260 an interactive program
name = raw_input('Who\'s there ? ')
print 'hello ', name , '!
```

The # signs the start of a comment, the line following # is ignored by the interpreter.

At the command prompt, we type

```
python hello_there.py
```

On Windows, double clicking on the file with .py extension will execute the program.
our first interactive program
running at the command prompt

The file `hello_there.py` contains

```python
# L-2 MCS 260 an interactive program
name = raw_input('Who\'s there ? ')
print 'hello ', name , '!
```

The # signs the start of a comment, the line following # is ignored by the interpreter.

At the command prompt, we type

```python
python hello_there.py
```

On Windows, double clicking on the file with `.py` extension will execute the program.
our first interactive program
running at the command prompt

The file `hello_there.py` contains

```
# L-2 MCS 260 an interactive program
name = raw_input('Who\'s there ? ')
print 'hello ', name , '!
```

The # signs the start of a comment, the line following # is ignored by the interpreter.

At the command prompt, we type

```
python hello_there.py
```

On Windows, double clicking on the file with `.py` extension will execute the program.
Computer Architecture

first steps with Python

1. Computer Architecture
   Hardware Components
   Programming Environments

2. Getting Started with Python
   Installing Python
   Executing Python code

3. Number Systems
   Decimal and Binary notations

4. Running Sage

5. Summary + Assignments
Decimal Notation of Numbers

- The value of $284 = 2 \times 10^2 + 8 \times 10^1 + 4 \times 10^0$. 2,8,4 are the digits of the number, 10 is the base. The position of each digit determines its contribution to the value of the number.

- For any base $B$, a number $n$ is denoted by $m$ coefficients $c_i$, $i = m, m - 1, \ldots, 1, 0$, $0 \leq c_i < B$:

$$n = c_m B^m + c_{m-1} B^{m-1} + \cdots + c_1 B^1 + c_0 B^0.$$  

- From base five to decimal notation:

$$2104_5 = 2 \times 5^3 + 1 \times 5^2 + 0 \times 5^1 + 4 \times 5^0 = 250 + 25 + 0 + 4 = 279_{10}$$
Decimal Notation of Numbers

- The value of $284 = 2 \times 10^2 + 8 \times 10^1 + 4 \times 10^0$. 2,8,4 are the digits of the number, 10 is the base. The position of each digit determines its contribution to the value of the number.

- For any base $B$, a number $n$ is denoted by $m$ coefficients $c_i$, $i = m, m - 1, \ldots, 1, 0$, $0 \leq c_i < B$:

$$n = c_mB^m + c_{m-1}B^{m-1} + \cdots + c_1B^1 + c_0B^0.$$ 

- From base five to decimal notation:

$$2104_5 = 2 \times 5^3 + 1 \times 5^2 + 0 \times 5^1 + 4 \times 5^0 = 250 + 25 + 0 + 4 = 279_{10}$$
Decimal Notation of Numbers

- The value of \( 284 = 2 \times 10^2 + 8 \times 10^1 + 4 \times 10^0 \). 2,8,4 are the digits of the number, 10 is the base. The position of each digit determines its contribution to the value of the number.

- For any base \( B \), a number \( n \) is denoted by \( m \) coefficients \( c_i \), \( i = m, m-1, \ldots, 1, 0 \), \( 0 \leq c_i < B \):
  \[
  n = c_m B^m + c_{m-1} B^{m-1} + \cdots + c_1 B^1 + c_0 B^0.
  \]

- From base five to decimal notation:
  \[
  2104_5 = 2 \times 5^3 + 1 \times 5^2 + 0 \times 5^1 + 4 \times 5^0 \\
  = 250 + 25 + 0 + 4 \\
  = 279_{10}
  \]
Binary Numbers

- The base is two, the coefficients are bits $\in \{0, 1\}$.

- The first 16 natural numbers — need 4 bits:
  
  $\begin{align*}
  0000 &= 0 & 0001 &= 1 & 0010 &= 2 & 0011 &= 3 \\
  0100 &= 4 & 0101 &= 5 & 0110 &= 6 & 0111 &= 7 \\
  1000 &= 8 & 1001 &= 9 & 1010 &= A & 1011 &= B \\
  1100 &= C & 1101 &= D & 1110 &= E & 1111 &= F
  \end{align*}$

  The hexadecimal ‘digits’ are 0,1,2,..,9,A,B,C,D,E,F.

- It is straightforward to convert binary into hexadecimal and hexadecimal into binary numbers.
Binary Numbers

- The base is two, the coefficients are bits $\in \{0, 1\}$.

- The first 16 natural numbers — need 4 bits:
  
  \[
  \begin{align*}
  0000 &= 0 & 0001 &= 1 & 0010 &= 2 & 0011 &= 3 \\
  0100 &= 4 & 0101 &= 5 & 0110 &= 6 & 0111 &= 7 \\
  1000 &= 8 & 1001 &= 9 & 1010 &= A & 1011 &= B \\
  1100 &= C & 1101 &= D & 1110 &= E & 1111 &= F 
  \end{align*}
  \]

  The hexadecimal ‘digits’ are 0,1,2,\ldots,9,A,B,C,D,E,F.

- It is straightforward to convert binary into hexadecimal and hexadecimal into binary numbers.
Binary Numbers

- The base is two, the coefficients are bits \(\in \{0, 1\}\).

- The first 16 natural numbers — need 4 bits:
  
  \[
  \begin{align*}
  0000 &= 0 & 0001 &= 1 & 0010 &= 2 & 0011 &= 3 \\
  0100 &= 4 & 0101 &= 5 & 0110 &= 6 & 0111 &= 7 \\
  1000 &= 8 & 1001 &= 9 & 1010 &= A & 1011 &= B \\
  1100 &= C & 1101 &= D & 1110 &= E & 1111 &= F
  \end{align*}
  \]

  The hexadecimal ‘digits’ are 0,1,2,…,9,A,B,C,D,E,F.

- It is straightforward to convert binary into hexadecimal and hexadecimal into binary numbers.
Binary Numbers

• The base is two, the coefficients are bits $\in \{0, 1\}$.

• The first 16 natural numbers — need 4 bits:

  $0000 = 0 \quad 0001 = 1 \quad 0010 = 2 \quad 0011 = 3$
  $0100 = 4 \quad 0101 = 5 \quad 0110 = 6 \quad 0111 = 7$
  $1000 = 8 \quad 1001 = 9 \quad 1010 = A \quad 1011 = B$
  $1100 = C \quad 1101 = D \quad 1110 = E \quad 1111 = F$

  The hexadecimal ‘digits’ are 0,1,2,...,9,A,B,C,D,E,F.

• It is straightforward to convert binary into hexadecimal and hexadecimal into binary numbers.
### Converting Numbers
from decimal to binary

#### Convert 123 into binary format:

<table>
<thead>
<tr>
<th>$n$</th>
<th>$n/2$</th>
<th>$n \mod 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>61</td>
<td>1</td>
</tr>
<tr>
<td>61</td>
<td>30</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

123 = $1 + 2 \times 61 = 1 + 2 \times (1 + 2 \times 30)$

= $1 + 2 \times (1 + 2 \times (0 + 2 \times 15))$

= $1 + 2 \times (1 + 2 \times (0 + 2 \times (1 + 2 \times 7)))$

= ... 

So 123 = 1111011 = 7B.
Converting Numbers from decimal to binary

Convert 123 into binary format:

<table>
<thead>
<tr>
<th>n</th>
<th>n/2</th>
<th>n mod 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>61</td>
<td>1</td>
</tr>
<tr>
<td>61</td>
<td>30</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

123 = 61 × 2 + 1
61 = 30 × 2 + 1
30 = 15 × 2 + 0
15 = 7 × 2 + 1
7 = 3 × 2 + 1
3 = 1 × 2 + 1
1 = 0 × 2 + 1

123 = 1 + 2 × 61 = 1 + 2 × (1 + 2 × 30)
= 1 + 2 × (1 + 2 × (0 + 2 × 15))
= 1 + 2 × (1 + 2 × (0 + 2 × (1 + 2 × 7)))
= ... 

So 123 = 1111011 = 7B.
Converting Numbers
from decimal to binary

Convert 123 into binary format:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>61</td>
<td>1</td>
</tr>
<tr>
<td>61</td>
<td>30</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

123 = 1 + 2 × 61 = 1 + 2 × (1 + 2 × 30)
    = 1 + 2 × (1 + 2 × (0 + 2 × 15))
    = 1 + 2 × (1 + 2 × (0 + 2 × (1 + 2 × 7)))
    = ...  

So 123 = 1111011 = 7B.
Converting Numbers
from decimal to binary

Convert 123 into binary format:

<table>
<thead>
<tr>
<th>n</th>
<th>n/2</th>
<th>n mod 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>61</td>
<td>1</td>
</tr>
<tr>
<td>61</td>
<td>30</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

123 = 1 + 2 × 61 = 1 + 2 × (1 + 2 × 30)
    = 1 + 2 × (1 + 2 × (0 + 2 × 15))
    = 1 + 2 × (1 + 2 × (0 + 2 × (1 + 2 × 7)))
    = ...  

So 123 = 1111011 = 7B.
Converting Numbers
from decimal to binary

Convert 123 into binary format:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>61</td>
<td>1</td>
</tr>
<tr>
<td>61</td>
<td>30</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
123 = 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0
\]

So 123 = 1111011 = 7B.
Converting Numbers
from decimal to binary

Convert 123 into binary format:

<table>
<thead>
<tr>
<th>$n$</th>
<th>$n/2$</th>
<th>$n \mod 2$</th>
<th>$123 = n/2 \times 2 + n \mod 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>61</td>
<td>1</td>
<td>$123 = 61 \times 2 + 1$</td>
</tr>
<tr>
<td>61</td>
<td>30</td>
<td>1</td>
<td>$61 = 30 \times 2 + 1$</td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>0</td>
<td>$30 = 15 \times 2 + 0$</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>1</td>
<td>$15 = 7 \times 2 + 1$</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1</td>
<td>$7 = 3 \times 2 + 1$</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>$3 = 1 \times 2 + 1$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$1 = 0 \times 2 + 1$</td>
</tr>
</tbody>
</table>

$$
123 = 1 + 2 \times 61 = 1 + 2 \times (1 + 2 \times 30)
= 1 + 2 \times (1 + 2 \times (0 + 2 \times 15))
= 1 + 2 \times (1 + 2 \times (0 + 2 \times (1 + 2 \times 7)))
= \ldots
$$

So $123 = 1111011 = 7B$. 
Converting Numbers
from decimal to binary

Convert 123 into binary format:

<table>
<thead>
<tr>
<th>$n$</th>
<th>$n/2$</th>
<th>$n \mod 2$</th>
<th>$n = 61 \times 2 + 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>61</td>
<td>1</td>
<td>$123 = 61 \times 2 + 1$</td>
</tr>
<tr>
<td>61</td>
<td>30</td>
<td>1</td>
<td>$61 = 30 \times 2 + 1$</td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>0</td>
<td>$30 = 15 \times 2 + 0$</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>1</td>
<td>$15 = 7 \times 2 + 1$</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1</td>
<td>$7 = 3 \times 2 + 1$</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>$3 = 1 \times 2 + 1$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$1 = 0 \times 2 + 1$</td>
</tr>
</tbody>
</table>

\[
123 = 1 + 2 \times 61 = 1 + 2 \times (1 + 2 \times 30) \\
= 1 + 2 \times (1 + 2 \times (0 + 2 \times 15)) \\
= 1 + 2 \times (1 + 2 \times (0 + 2 \times (1 + 2 \times 7))) \\
= \ldots
\]

So $123 = 1111011 = 7B$. 
Converting Numbers
from decimal to binary

Convert 123 into binary format:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>61</td>
<td>1</td>
</tr>
<tr>
<td>61</td>
<td>30</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
123 = 1 + 2 \times 61 = 1 + 2 \times (1 + 2 \times 30) \\
= 1 + 2 \times (1 + 2 \times (0 + 2 \times 15)) \\
= 1 + 2 \times (1 + 2 \times (0 + 2 \times (1 + 2 \times 7))) \\
= \ldots
\]

So 123 = 1111011 = 7B.
Converting Numbers from decimal to binary

Convert 123 into binary format:

<table>
<thead>
<tr>
<th>$n$</th>
<th>$n/2$</th>
<th>$n \mod 2$</th>
<th>$123 = 61 \times 2 + 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>61</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>30</td>
<td>1</td>
<td>$61 = 30 \times 2 + 1$</td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>0</td>
<td>$30 = 15 \times 2 + 0$</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>1</td>
<td>$15 = 7 \times 2 + 1$</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1</td>
<td>$7 = 3 \times 2 + 1$</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>$3 = 1 \times 2 + 1$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$1 = 0 \times 2 + 1$</td>
</tr>
</tbody>
</table>

So $123 = 1 + 2 \times 61 = 1 + 2 \times (1 + 2 \times 30)$

$= 1 + 2 \times (1 + 2 \times (0 + 2 \times 15))$

$= 1 + 2 \times (1 + 2 \times (0 + 2 \times (1 + 2 \times 7)))$

$= \ldots$

So $123 = 1111011 = 7B$. 
Converting Numbers
from decimal to binary

Convert 123 into binary format:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>61</td>
<td>1</td>
</tr>
<tr>
<td>61</td>
<td>30</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

So 123 = 1111011 = 7B.
Converting Numbers
from decimal to binary

Convert 123 into binary format:

<table>
<thead>
<tr>
<th>$n$</th>
<th>$n/2$</th>
<th>$n \mod 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>61</td>
<td>1</td>
</tr>
<tr>
<td>61</td>
<td>30</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

123 = 1 + 2 \times 61 = 1 + 2 \times (1 + 2 \times 30)
  = 1 + 2 \times (1 + 2 \times (0 + 2 \times 15))
  = 1 + 2 \times (1 + 2 \times (0 + 2 \times (1 + 2 \times 7)))
  = \ldots

So 123 = 1111011 = 7B.
## Converting Numbers from decimal to binary

### Convert 123 into binary format:

<table>
<thead>
<tr>
<th>$n$</th>
<th>$n/2$</th>
<th>$n \mod 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>61</td>
<td>1</td>
</tr>
<tr>
<td>61</td>
<td>30</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

$123 = 61 \times 2 + 1$

$61 = 30 \times 2 + 1$

$30 = 15 \times 2 + 0$

$15 = 7 \times 2 + 1$

$7 = 3 \times 2 + 1$

$3 = 1 \times 2 + 1$

$1 = 0 \times 2 + 1$

$123 = 1 + 2 \times 61 = 1 + 2 \times (1 + 2 \times 30)$

$= 1 + 2 \times (1 + 2 \times (0 + 2 \times 15))$

$= 1 + 2 \times (1 + 2 \times (0 + 2 \times (1 + 2 \times 7)))$

$= \ldots$

So $123 = 1111011 = 7B$. 
Converting Numbers
from decimal to binary

Convert 123 into binary format:

<table>
<thead>
<tr>
<th>n</th>
<th>n/2</th>
<th>n mod 2</th>
<th>123 = n/2 \times 2 + n mod 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>61</td>
<td>1</td>
<td>123 = 61 \times 2 + 1</td>
</tr>
<tr>
<td>61</td>
<td>30</td>
<td>1</td>
<td>61 = 30 \times 2 + 1</td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>0</td>
<td>30 = 15 \times 2 + 0</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>1</td>
<td>15 = 7 \times 2 + 1</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1</td>
<td>7 = 3 \times 2 + 1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>3 = 1 \times 2 + 1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1 = 0 \times 2 + 1</td>
</tr>
</tbody>
</table>

123 = 1 + 2 \times 61 = 1 + 2 \times (1 + 2 \times 30)
= 1 + 2 \times (1 + 2 \times (0 + 2 \times 15))
= 1 + 2 \times (1 + 2 \times (0 + 2 \times (1 + 2 \times 7)))
= \ldots

So 123 = 1111011 = 7B.
### Converting Numbers from decimal to binary

#### Convert 123 into binary format:

<table>
<thead>
<tr>
<th></th>
<th>n/2</th>
<th>n mod 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>61</td>
<td>1</td>
</tr>
<tr>
<td>61</td>
<td>30</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

So 123 = 1111011 = 7B.
Converting Numbers
from decimal to binary

Convert 123 into binary format:

<table>
<thead>
<tr>
<th>$n$</th>
<th>$n/2$</th>
<th>$n \mod 2$</th>
<th>$123 = 61 \times 2 + 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>61</td>
<td>1</td>
<td>123 = 61 × 2 + 1</td>
</tr>
<tr>
<td>61</td>
<td>30</td>
<td>1</td>
<td>61 = 30 × 2 + 1</td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>0</td>
<td>30 = 15 × 2 + 0</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>1</td>
<td>15 = 7 × 2 + 1</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1</td>
<td>7 = 3 × 2 + 1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>3 = 1 × 2 + 1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1 = 0 × 2 + 1</td>
</tr>
</tbody>
</table>

$123 = 1 + 2 \times 61 = 1 + 2 \times (1 + 2 \times 30)$
$= 1 + 2 \times (1 + 2 \times (0 + 2 \times 15))$
$= 1 + 2 \times (1 + 2 \times (0 + 2 \times (1 + 2 \times 7)))$
$= \ldots$

So 123 = 1111011 = 7B.
Converting Numbers
from decimal to binary

Convert 123 into binary format:

<table>
<thead>
<tr>
<th>n</th>
<th>n/2</th>
<th>n mod 2</th>
<th>123 = 61 × 2 + 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>61</td>
<td>1</td>
<td>123 = 61 × 2 + 1</td>
</tr>
<tr>
<td>61</td>
<td>30</td>
<td>1</td>
<td>61 = 30 × 2 + 1</td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>0</td>
<td>30 = 15 × 2 + 0</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>1</td>
<td>15 = 7 × 2 + 1</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1</td>
<td>7 = 3 × 2 + 1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>3 = 1 × 2 + 1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1 = 0 × 2 + 1</td>
</tr>
</tbody>
</table>

\[
123 = 1 + 2 \times 61 = 1 + 2 \times (1 + 2 \times 30) \\
= 1 + 2 \times (1 + 2 \times (0 + 2 \times 15)) \\
= 1 + 2 \times (1 + 2 \times (0 + 2 \times (1 + 2 \times 7))) \\
= \ldots \\

So 123 = 1111011 = 7B.
Converting Numbers from decimal to binary

Convert 123 into binary format:

<table>
<thead>
<tr>
<th>n</th>
<th>n/2</th>
<th>n \mod 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>61</td>
<td>1</td>
</tr>
<tr>
<td>61</td>
<td>30</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
123 = 61 \times 2 + 1 = 30 \times 2 + 1 = 15 \times 2 + 1 = 7 \times 2 + 1 = 3 \times 2 + 1 = 0 \times 2 + 1
\]

So \(123 = 1111011 = 7B\).
## Converting Numbers

from decimal to binary

Convert 123 into binary format:

<table>
<thead>
<tr>
<th>n</th>
<th>n/2</th>
<th>n mod 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>61</td>
<td>1</td>
</tr>
<tr>
<td>61</td>
<td>30</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

So 123 = 1 + 2 × 61 = 1 + 2 × (1 + 2 × 30) = 1 + 2 × (1 + 2 × (0 + 2 × 15)) = 1 + 2 × (1 + 2 × (0 + 2 × (1 + 2 × 7))) = ... 

So 123 = 1111011 = 7B.
Converting Numbers
from decimal to binary

Convert 123 into binary format:

<table>
<thead>
<tr>
<th>n</th>
<th>n/2</th>
<th>n mod 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>61</td>
<td>1</td>
</tr>
<tr>
<td>61</td>
<td>30</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
123 = 1 + 2 \times 61 = 1 + 2 \times (1 + 2 \times 30) \\
= 1 + 2 \times (1 + 2 \times (0 + 2 \times 15)) \\
= 1 + 2 \times (1 + 2 \times (0 + 2 \times (1 + 2 \times 7))) \\
= \ldots
\]

So 123 = 1111011 = 7B.
Flowchart
conversion algorithm

n = input()

r is remainder of n/2
print r; n := n/2

n == 0?
Yes stop
No
Flowchart conversion algorithm

```
n = input()
r is remainder of n/2
print r; n := n/2
```

```
n == 0?
Yes stop
No
```
Flowchart conversion algorithm

```
n = input()

r is remainder of n/2
print r; n := n/2

n == 0?
Yes stop
No
```
Running Sage

Ways to compute with Sage:

- a notebook blends commands with commentary
  - graphical user interface
  - runs in a browser
- command line use if no graphical output is needed and more dedicated to computationally intensive jobs
- language is Python, with some variation

Python:

```python
>>> 68/25
2
```

Sage:

```python
sage: 68/25
68/25
```
Running Sage

Ways to compute with Sage:

- a notebook blends commands with commentary
  - graphical user interface
  - runs in a browser

- command line use if no graphical output is needed and more dedicated to computationally intensive jobs

- language is Python, with some variation

Python:

```python
>>> 68/25
2
```

Sage:

```sage
sage: 68/25
68/25
```
Running Sage

Ways to compute with Sage:

- a notebook blends commands with commentary
  - graphical user interface
  - runs in a browser

- command line use if no graphical output is needed and more dedicated to computationally intensive jobs

- language is Python, with some variation

Python:

```python
>>> 68/25
2
```

Sage:

```sage
sage: 68/25
68/25
```
Running Sage

Ways to compute with Sage:

- a notebook blends commands with commentary
  - graphical user interface
  - runs in a browser

- command line use if no graphical output is needed and more dedicated to computationally intensive jobs

- language is Python, with some variation

Python:

```python
>>> 68/25
2
```

Sage:

```
sage: 68/25
68/25
```
Number Systems
with Python and Sage

Hexadecimal conversions in Python with the % operator:

```python
>>> "%X" % 123
'7B'
>>> '%x' % 123
'7b'
```
Summary + Assignments

Recommended reading:

- Python tutorial on http://docs.python.org/tut/tut.html
- sections 1.4 & 1.5 of Computer Science. An Overview
- chapter 1 and start of Chapter 3 of Python Power!

Assignments:

1. Given the base and a sequence of coefficients of a number, give the algorithm to evaluate the number.
2. Write down pseudocode for the algorithm to compute the binary representation of a number.
3. Compute examples of general number conversions from any base to any other base.
4. What is the algorithm for such general conversions?