Introduction to Programming
A Note about Homeworks

• Be sure to read the homework description carefully to be sure you include all the information it requests

• We have seen homeworks drop a full letter grade because they omitted the response to an important question
Textual data

• Text data exists in many different forms

• Analyzing the text can provide insights into the intent and the language of the author
Data magnitudes

<table>
<thead>
<tr>
<th>Description</th>
<th>Words</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Senate votes per year</td>
<td>~15,000</td>
<td>~15KB</td>
</tr>
<tr>
<td>Kung Fu Panda script</td>
<td>~15,000</td>
<td>80KB</td>
</tr>
<tr>
<td>Alice in Wonderland</td>
<td>~26,000</td>
<td>170KB</td>
</tr>
<tr>
<td>The English Bible</td>
<td>~800,000</td>
<td>4.2MB</td>
</tr>
<tr>
<td>Trump’s Tweets</td>
<td>~800,000</td>
<td>~4MB</td>
</tr>
<tr>
<td>Encyclopædia Britannica</td>
<td>~40M</td>
<td>~200MB</td>
</tr>
<tr>
<td>English Wikipedia</td>
<td>~2.9B</td>
<td>~14.5GB</td>
</tr>
</tbody>
</table>

- Spreadsheets (barely)
- Ideally suited for programming language
- Analysis difficult for single computer
Programming Languages

A short and incomplete history
First Programming Language
Programming languages

- A **programming language** is a formal language to describe tasks to a machine or computer.

- A set of instructions used to control the behavior of a computer/machine is called a **program**.
First Computer Program

Invented the digital programmable computer the “Analytical Engine”
Not actually completed

Wrote the first program for the Analytical Engine (1843)

Charles Babbage

Ada Lovelace
Diagram for the computation by the Engine of the Numbers of Bernoulli. See Note G. (page 722 et seq.)

<table>
<thead>
<tr>
<th>Number</th>
<th>Nature of Operation</th>
<th>Variables acted upon</th>
<th>Variables receiving results</th>
<th>Indication of change in the value on any Variable</th>
<th>Statement of Results</th>
<th>Data</th>
<th>Working Variables</th>
<th>Result Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( V_x \times V_y )</td>
<td>( V_x, V_y )</td>
<td>( V_3, V_4, V_5 )</td>
<td>2 ( n )</td>
<td>2 ( n ) 2 ( n ) 2 ( n )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>( V_x - V_y )</td>
<td>( V_x )</td>
<td>( V_4 )</td>
<td>2 ( n - 1 )</td>
<td>2 ( n - 1 )</td>
<td>2 ( n - 1 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>( V_x + V_y )</td>
<td>( V_x )</td>
<td>( V_5 )</td>
<td>2 ( n + 1 )</td>
<td>2 ( n + 1 )</td>
<td>2 ( n + 1 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>( V_x + V_y )</td>
<td>( V_x )</td>
<td>( V_7 )</td>
<td>2 ( n )</td>
<td>0 0</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>( V_x + V_y )</td>
<td>( V_x )</td>
<td>( V_9 )</td>
<td>1 2 4</td>
<td>1 2 4</td>
<td>1 2 4 1 2 4 1 2 4 1 2 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>( V_x + V_y )</td>
<td>( V_x )</td>
<td>( V_9 )</td>
<td>( 2n-1 )</td>
<td>( 2n-1 )</td>
<td>( 2n-1 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>( V_x - V_y )</td>
<td>( V_x )</td>
<td>( V_9 )</td>
<td>( 2n+1 )</td>
<td>( 2n+1 )</td>
<td>( 2n+1 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>( V_x + V_y )</td>
<td>( V_x )</td>
<td>( V_9 )</td>
<td>( 2n )</td>
<td>( 2n )</td>
<td>( 2n )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>( V_x + V_y )</td>
<td>( V_x )</td>
<td>( V_9 )</td>
<td>( 2n+1 )</td>
<td>( 2n+1 )</td>
<td>( 2n+1 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>( V_x + V_y )</td>
<td>( V_x )</td>
<td>( V_9 )</td>
<td>( 2n-1 )</td>
<td>( 2n-1 )</td>
<td>( 2n-1 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>( V_x + V_y )</td>
<td>( V_x )</td>
<td>( V_9 )</td>
<td>( 2n+1 )</td>
<td>( 2n+1 )</td>
<td>( 2n+1 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>( V_x - V_y )</td>
<td>( V_x )</td>
<td>( V_9 )</td>
<td>( 2n )</td>
<td>( 2n )</td>
<td>( 2n )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>( V_x - V_y )</td>
<td>( V_x )</td>
<td>( V_9 )</td>
<td>( 2n+1 )</td>
<td>( 2n+1 )</td>
<td>( 2n+1 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>( V_x + V_y )</td>
<td>( V_x )</td>
<td>( V_9 )</td>
<td>( 2n-1 )</td>
<td>( 2n-1 )</td>
<td>( 2n-1 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>( V_x + V_y )</td>
<td>( V_x )</td>
<td>( V_9 )</td>
<td>( 2n+1 )</td>
<td>( 2n+1 )</td>
<td>( 2n+1 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>( V_x + V_y )</td>
<td>( V_x )</td>
<td>( V_9 )</td>
<td>( 2n-1 )</td>
<td>( 2n-1 )</td>
<td>( 2n-1 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>( V_x + V_y )</td>
<td>( V_x )</td>
<td>( V_9 )</td>
<td>( 2n+1 )</td>
<td>( 2n+1 )</td>
<td>( 2n+1 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>( V_x - V_y )</td>
<td>( V_x )</td>
<td>( V_9 )</td>
<td>( 2n )</td>
<td>( 2n )</td>
<td>( 2n )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>( V_x - V_y )</td>
<td>( V_x )</td>
<td>( V_9 )</td>
<td>( 2n+1 )</td>
<td>( 2n+1 )</td>
<td>( 2n+1 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>( V_x + V_y )</td>
<td>( V_x )</td>
<td>( V_9 )</td>
<td>( 2n-1 )</td>
<td>( 2n-1 )</td>
<td>( 2n-1 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>( V_x + V_y )</td>
<td>( V_x )</td>
<td>( V_9 )</td>
<td>( 2n+1 )</td>
<td>( 2n+1 )</td>
<td>( 2n+1 )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Alan Turing’s Universal Turing Machine 1936

- A mathematical description of a general purpose programmable computer
- Foundational for modern analytical methods in computer science
ENIAC (1946)

• One of first electronic general-purpose computers

• Used primarily for ballistic research
Writing a program to build a concordance
Concordances

- An alphabetical list of all words in a text
- Includes where the word can be found
- Includes the context surrounding the text
### A Digital Concordance

Alphabetical index of all words in a text

<table>
<thead>
<tr>
<th>Word</th>
<th>Indices</th>
<th>Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘a’</td>
<td>19,32,58, 82,…</td>
<td>A mysterious warrior treks across …</td>
</tr>
<tr>
<td>‘aaaaaaaaaaaaaaaaaaaaaaaaaaaaa!’</td>
<td>12099</td>
<td>PO: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA</td>
</tr>
<tr>
<td>aaaagghhh!’</td>
<td>14084</td>
<td>PO: AAAAGGGHH</td>
</tr>
<tr>
<td>‘aaahh!’</td>
<td>3066</td>
<td>Crowd: Oooh! Aaahh!</td>
</tr>
<tr>
<td>‘aaaoo...whoohoo...eeeeee...hee-'</td>
<td>7462</td>
<td>PO: Aaaoo...whoohoo...EEEee...hee- hee...</td>
</tr>
</tbody>
</table>
Concordances before computers

• Take a large amount of time

• Only on important texts like the Vedas, Bible, Qur’an and other notable texts, would people build concordances

• The first concordance of the Bible took 500 people

• The first concordance of the Hebrew Bible took 10 years
Cruden’s Concordance (1737)

• Created a concordance of the King James Bible
• Worked 7am - 1am, 18 hours each day
• Completed in about a year
• King James bible contains 783,137 words
• Recorded about 120 words / hour
• Contained errors and omitted parts
Our n are under persecution................. Lam 5:5 6677
upon the n of them that are slain ...... Eze 21:29 6677
which ye shall not remove your n....... Mic 2:3 6677
for my life laid down their own n....... Rom 16:4 5137

2706
316
1876
NECO See NECHOH.

NECROMANCER
spirits, or a wizard, or a n............... Deut 18:11

316
316
1876
NEDABIAH (ned-a-bi'-ah) Son of Jeconiah.
Shenazar, Jecamiah, Hoshama, and N. 1Chr 3:18 5072

316
316
318
NEED
lend him sufficient for his n............... Deut 15:8 4270
Have I n of mad men, that ye have...... 1Sa 21:15 2638
Lebanon, as much as thou shalt n....... 2Chr 2:16 6878
Ye shall not n to fight in this............ 2Chr 20:17
And that which they have n of............ Ezr 6:9 2818
that he shall have no n of spoil........ Prov 31:11 2637
You’ll write your own concordance building program at the end of this textual analysis section.

For a short book like Alice in Wonderland, it will run in about 10ms.

For the entire King James Bible, your program will probably run in less than a second.

We could probably use the same program on the English Wikipedia and it would take less than an hour.
Python the Programming Language
Python Programs

• Python is a **high-level** programming language: single lines of Python code generally specify more than one instruction to the computer

• This means that your Python code is **compiled** to **low-level** instructions called **byte-code** before your code is actually run

• The byte-code is then **interpreted** line-by-line into instructions the computer understands and executed simultaneously

• Python is **general purpose**, and is not particularly specialized to a single task, unlike a language like R, Matlab or Stata

• Python is **open-source**, and a large community supports the code
Python Programs

- You will be writing programs in the Python language.

- Often times, these programs will require code written by others known as **libraries**.

- In python, a library is a collection of **modules**, and your program will reference the individual modules.
Basic Types in Python

- Boolean Values: True  False
- Numerical Values: 0  1  2.5  -3.75
- Strings: “Hello”  ‘Hey there!’  ‘I said, “Hello”’
Numerical Types: Integers

- Python’s type for integers is `int`
- Whole numbers only
- Calculations with integers are exact, except division
- Calculations with integers are fast
- If your integer value is between ±9,223,372,036,854,775,807, your computer can do addition and subtraction in a single clock cycle of your processor (0.33ns for 3GHz processor).
Floating Point

- Python's type is called `float`
- Is like scientific notation (5E10, 5*10^{10}), but in binary (3*2^{5})
- Each float has a significand and an exponent
- The significand has 53 bits of precision, which is about 15 effective digits in decimal precision
- The exponent is limited to 11 bits
- Float values can represent a range as small as 10^{-307} and as large as 10^{308}
- Not all numbers that can be precisely represented in decimal can be represented in binary

\[ -3.5 \times 2^1 = 23.98 \]
Floating Point Errors

• Floating point has a limited precision

• Floating point can’t represent decimal numbers precisely

• Every mathematical operation in floating point is approximate

• Testing floating point numbers for equality is dangerous

\[ 0.1 + 0.2 = 0.30000000000000004 \]
A syntax error occurs during the compilation stage if your code is improperly formed.

They show up immediately, no matter where the issue exists in the code.
Exceptions

- An **exception** is **thrown** whenever the interpreter reaches a line of code that results in an improper calculation.

- These exceptions occur at the specific point in time the improper calculation is encountered.

- For a long running program, these may occur late in your program.

```python
1 int("Hello")
```

```
Traceback (most recent call last):
  File "concordance.py", line 1, in <module>
    int("Hello")
ValueError: invalid literal for int() with base 10: 'Hello'
```
More Exceptions

• SyntaxError: Your code is malformed and you will need to fix it, often results due to unclosed (), {}, []

• TypeError: The function expects a value of a specific type, but received a different one instead

• ValueError: The function expected a value with specific properties, but the value passed in did not meet them.

```
1    abs("Hello")
```

Traceback (most recent call last):
  File "concordance.py", line 1, in <module>
    abs("Hello")
TypeError: bad operand type for abs(): 'str'
Sublime Demo
Atom Demo
Python IDLE Demo
print(contents)

• Takes in a single value, can be a string, a number or boolean and outputs the result to the command line

• We will use print to display the results of your computations

• We will also use print to help find issues in our code, by printing out intermediate calculations
help()

• When used in the Python interactive session, will provide the documentation on any function or object type

• The input is just the name of the function or object type

• Don’t call the function inside help, it will run help on the return value of that function
Additional Python Resources

- Python’s Language Reference: https://docs.python.org/3/library/index.html
- Python’s Built-in Function Reference: https://docs.python.org/3/library/functions.html
- Stack Overflow: stackoverflow.com
- Google, in your query be sure to include Python