Objectives

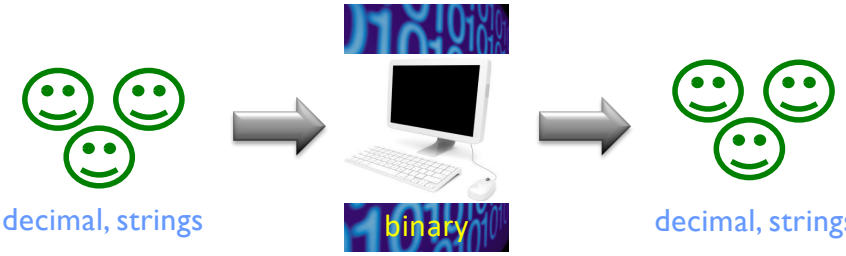
- More: computer's representations of data types
- Encryption

Review

- What is the special name for sequences, like newlines, tabs, …?
  - How do we represent them in strings?
- How does the computer represent data (e.g., numbers and text)?
- What is your algorithm for converting binary to decimal?

Review: Representations of Data

- Computer needs to represent different types of data
  - Eventually, all boils down to 1s and 0s
- Computer needs to translate between what humans know to what computer knows and back again



Seems like a divergence on strings but just wait…

Converting Binary to Decimal

- Generalize this process into an algorithm.
- Define good test cases for this algorithm/function
  - "Run" your algorithm on these test cases
- Implement as function:

```
binaryToDecimal(binaryNum)
```

(Not necessarily sequential steps)

Algorithm 1: Converting Binary → Decimal

Left to right traversal of binary number

Accumulator design pattern

Given the binary number as a string

1. Initialize the result to zero
2. The starting exponent will be the length of the string-1
3. For each bit in the binary number
   - Multiply the bit by the appropriate power of 2
   - Add this to the result
   - Reduce the exponent by 1
4. Return the result

5

Algorithm 2: Converting Binary → Decimal

Right to left traversal of binary number

Accumulator design pattern

Given the binary number as a string

1. Initialize the result to zero
2. Initialize the exponent to zero
3. Iterate over the positions of the binary number from right to left
   - Determine the bit at that position in the binary number
   - Multiply the bit by the appropriate power of 2
   - Add this to the result
   - Increase the exponent by 1
4. Return the result

6

Practice

- Implement both algorithms
  - Test!
- After implementing, you can compare with my solutions
  - `binaryToDecimalIterateOverCharacters.py`
  - `binaryToDecimalIterateOverExponents.py`

Algorithm: Converting Decimal → Binary

Given the decimal as an integer…

1. Initialize the result to the empty string
2. Repeat until the decimal is 0:
   - result = str(decimal % 2) + result
   - decimal = decimal // 2
3. Return the result

1. Try out algorithm with 22 as input
2. Implement algorithm in function `decimalToBinary`
3. Good test cases?

String Representations

- A **string** is a *sequence* of characters
- Each character is stored as a binary number
- **ASCII** (American Standard Code for Information Interchange) is one standard encoding for characters
  - Limitation: ASCII is based on the English language
  - Cannot represent other types of characters
  - Handout is just a subset
- Unicode is a new standard

Translating to/from ASCII

- Translate a character into its ASCII numeric code using **built-in function** `ord`
  - `ord('a') ==> 97`
- Translate an ASCII numeric code into its character using **built-in function** `chr`
  - `chr(97) ==> 'a'`

`ascii_table.py`
`ascii.py`

ASCII Questions

- Lowercase letters are represented by what range of numbers?
- Uppercase letters are represented by what range of numbers?
- What is the difference between the decimal encoding of 'M' and 'N'?
  - Between 'm' and 'n'?
- Explain the result of `"Zebra" < "aardvarks"` being True

ASCII Questions

- Lowercase letters are represented by what range of numbers?
  - 97—122
- Uppercase letters are represented by what range of numbers?
  - 65—90
- What is the difference between the decimal encoding of 'M' and 'N'?
  - Between 'm' and 'n'?
  - 1
- Explain the result of `"Zebra" < "aardvarks"` being True
  - `ord("Z") < ord("a")`

Translating to/from ASCII

- Translate a character into its ASCII numeric code using **built-in function** `ord`
  - `ord('a') ==> 97`
- Translate an ASCII numeric code into its character using **built-in function** `chr`
  - `chr(97) ==> 'a'`

`ascii_table.py`
`ascii.py`

Encryption

- Process of encoding information to keep it secure
- One technique: Substitution Cipher
  - Each character in message is replaced by a new character

Caesar Cipher

- Replace character with a character X places away
  - X is called the ***key***
- Julius Caesar used technique to communicate with his generals
- "Wrap around" within the lowercase letters
- Write program(s) to do this in next lab

Original Letter	Key	Encrypted Letter
'a'	1	'b'
'b'	1	'c'
'z'	1	'a'

Broader Issue: Natural Language Processing

- Why is Natural Language Processing (NLP) hard?
- What approaches were used?
  - What terms were new to you?
  - Start at the beginning: how could you implement Google suggest?
- How well did the suggested text do?
  - Would you recognize that it was machine-generated?
  - What is your favorite suggestion fail?
- What are useful and/or malicious applications of NLP? (Now and in the future)

Example Auto-Generated Text

- "By that I mean, it seemed to want to distinguish my feelings from my thoughts. To put it another way, Smart Compose seemed to want to know me."
- "The safety of any new technology often hinges on how it's regulated. If machines can learn to think for themselves, that might be a concern. But if we really want to replicate human intelligence—as most of us want to—there are several directions that researchers might explore."

17

BI: How Could I Do That?

- Or, what could I do with that?

NLP Word Generator API
`nextWord()`

```
sentence = ""
while not sentence.endswith("."):
    sentence.add(" ")
    sentence.add( nlp.nextWord() )
```


18

Looking Ahead

- Pre Lab 7 due before lab