Comparing strings

How can we compute a boolean for ( "apple" < "banana")?

Strings are compared lexicographically

e.g. left to right, comparing each corresponding pair of letters

Each letter is represented as a number on the computer

code: ex: ‘A’ = 65, ‘B’ = 66, etc

every letter corresponds to a positive integer

ord(c) ← returns ascii value for character

chr(num) ← returns character for an ascii value

encoding refers to how numbers map to characters -> ascii isn’t the only one!
Exercise - Ascii values

What is the result of “apple” < “APPLE”? Explain in terms of ascii values.

What is the ord of “A”, “0”, or “ “?

Check that “apple” == “apple” is True

Check that “apple” == “Apple” is False

Check that “apple” == “apple ” is False (watch out for the extra space!)
Dot Dash

Approach 1

```python
def main():
    word = input("Enter a string: ")

    word_length = len(word)
    if word_length % 2 == 0:  # even
        result = "."
        for i in range(word_length):
            result = result + word[i] + "."
    else:
        result = "-"
        for i in range(word_length):
            result = result + word[i] + "-"

    print(result)

main()
```
Analysis:

How does the accumulator work on lines 15-17?

word = cats
word_length = 4
result = “.” # initial value

<table>
<thead>
<tr>
<th>Iteration</th>
<th>i</th>
<th>word[i]</th>
<th>result = result + word[i] + “.”</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>“c”</td>
<td>result = “.” + “c” + “.” = “.c.”</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>“a”</td>
<td>result = “.c.” + “a” + “.” = “.c.a.”</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>“t”</td>
<td>result = “.c.a.” + “t” + “.” = “.c.a.t.”</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>“s”</td>
<td>result = “.c.a.t.” + “s” + “.” = “.c.a.t.s.”</td>
</tr>
</tbody>
</table>
Dot Dash

Approach 2

```python
def main():
    word = input("Enter a string: ")
    word_length = len(word)
    if word_length % 2 == 0:  # even
        result = "."
    else:
        result = "."
    for i in range(word_length):
        if word_length % 2 == 0:  # even
            result = result + word[i] + "."
        else:
            result = result + word[i] + "."
    print(result)
main()
```
Dot Dash

Approach 3

```python
def main():
    word = input("Enter a string: ")

    word_length = len(word)

    if word_length % 2 == 0:  # even
        delimiter = "."
    else:
        delimiter = "-"

    result = delimiter
    for i in range(word_length):
        result = result + word[i] + delimiter

    print(result)

main()
```