BIS 420 PROGRAMMING FOR DATA SCIENCE

PRAJAKTA POHARE
CHAPTER 8 EXERCISE 8.12

ILLINOIS STATE UNIVERSITY

ROT13 is a weak form of encryption that involves “rotating” each letter in a word by 13 places. To rotate a letter means to shift it through the alphabet, wrapping around to the beginning if necessary, so ’A’ shifted by 3 is ’D’ and ’Z’ shifted by 1 is ’A’.
Write a function called rotate_word that takes a string and an integer as parameters, and that returns a new string that contains the letters from the original string “rotated” by the given amount.
For example, “cheer” rotated by 7 is “jolly” and “melon” rotated by -10 is “cubed”.
You might want to use the built-in functions ord, which converts a character to a numeric code, and chr, which converts numeric codes to characters.
Potentially offensive jokes on the Internet are sometimes encoded in ROT13. If you are not easily offended, find and decode some of them. Solution: http: // thinkpython. com/ code/ rotate. py.

Week 8 > BIS420_PrajaktaPohare_Ch8_8.12.py > ...

```python
def rotate_word(s, n):
    result = ''
    for char in s:
        if char.isalpha():
            start = ord('A') if char.isupper() else ord('a')
            rotated_char = chr(start + (ord(char) - start + n) % 26)
            result += rotated_char
        else:
            result += char
    return result

print(rotate_word("cheer", 7))
print(rotate_word("melon", -10))
```

```
def rotate_word(s, n):
    result = ''
    for char in s:
        if char.isalpha():
            start = ord('A') if char.isupper() else ord('a')
            rotated_char = chr(start + (ord(char) - start + n) % 26)
            result += rotated_char
        else:
            result += char
    return result

print(rotate_word("cheer", 7))
print(rotate_word("melon", -10))
```