# CS50 for JDs

## Python Lab

## Cash

IN GOD WE TRUST
LIBERTY
2013
S

1¢

IN GOD WE TRUST
Liberty
2013
S

5¢

LIBERTY
IN GOD WE TRUST
D
2017

10¢

UNITED STATES OF AMERICA
IN GOD WE TRUST
LIBERTY
P
QUARTER DOLLAR

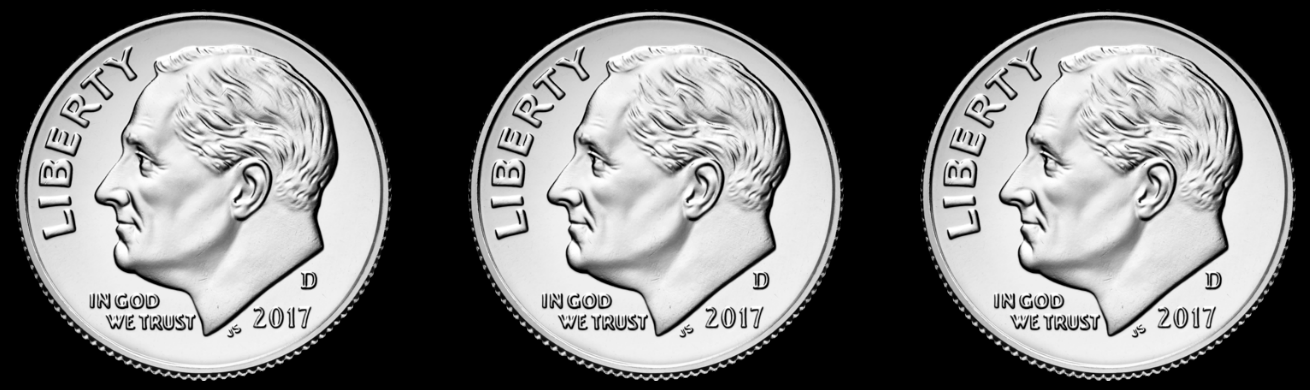
25¢

30¢

# 30¢

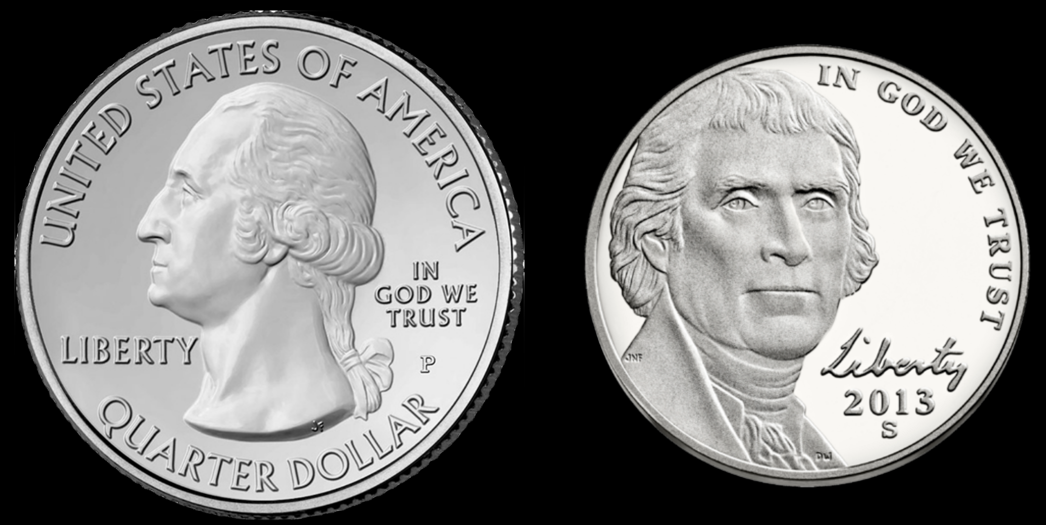
Coins: 30

# 30¢



Coins: 3

# 30¢


UNITED STATES OF AMERICA
IN GOD WE TRUST
LIBERTY
P
QUARTER DOLLAR
IN GOD WE TRUST
Liberty
2013
S

Coins: 2

## Cash

Given an amount of money, determine the fewest number of coins necessary to make change for that amount of money.

## Cash: Greedy Algorithm

Use the largest value coin possible at each step of the algorithm.

# 30¢

1¢ 5¢ 10¢ 25¢

Make change for: **30¢**

# 30¢

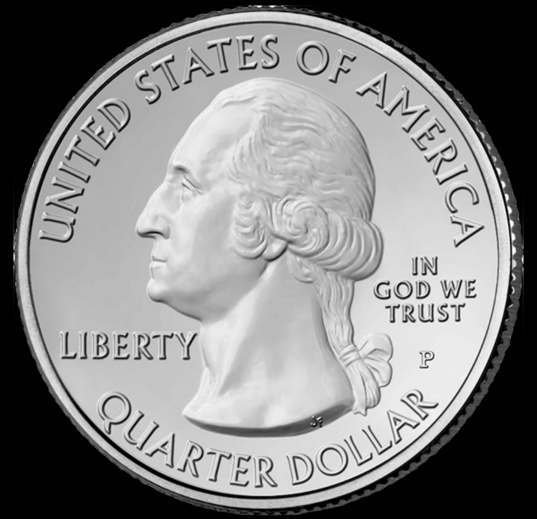
1¢

5¢

10¢

25¢

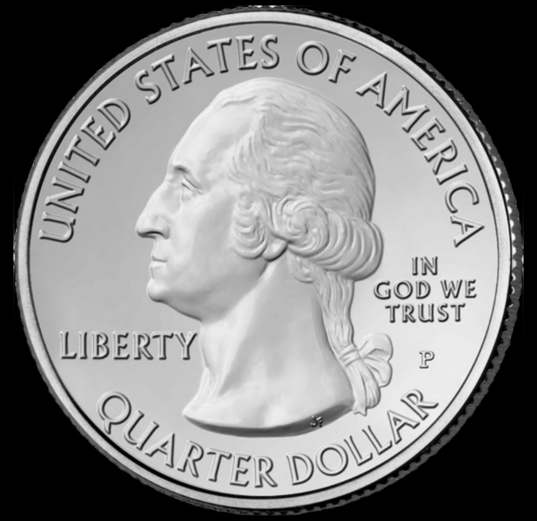
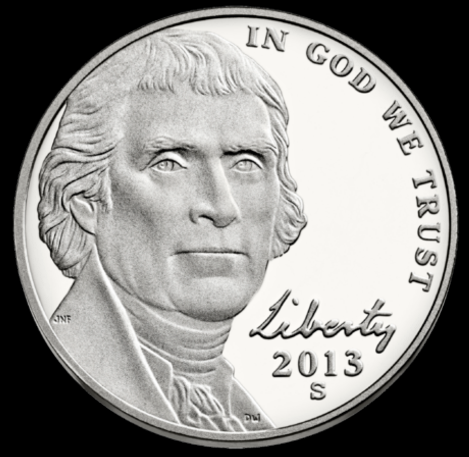
Make change for: **5¢**



30¢

1¢ 5¢ 10¢ 25¢

Make change for: **0¢**

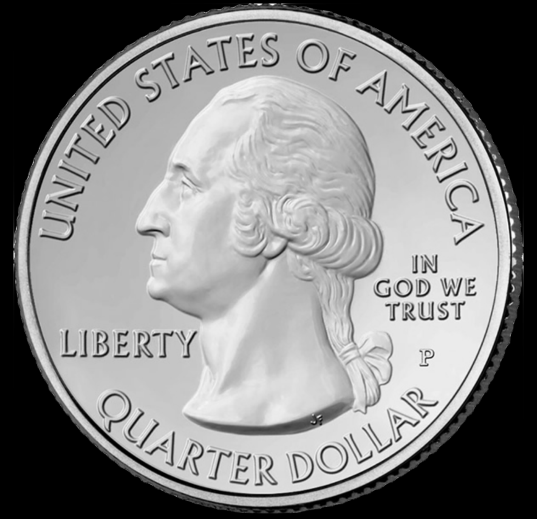
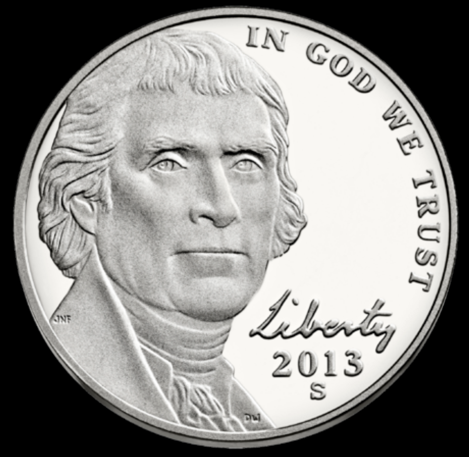
## 30¢

1¢

5¢

10¢

25¢

Coins used: **2**

# 47¢

1¢ 5¢ 10¢ 25¢

Make change for: **47¢**

## 47¢

1¢ 5¢ 10¢ 25¢

Make change for: **22¢**

# 47¢

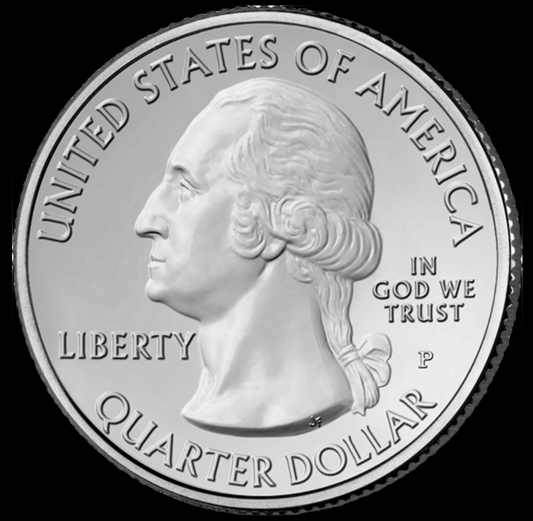
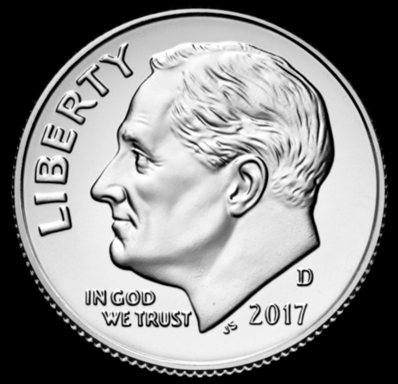
1¢

5¢

10¢

25¢

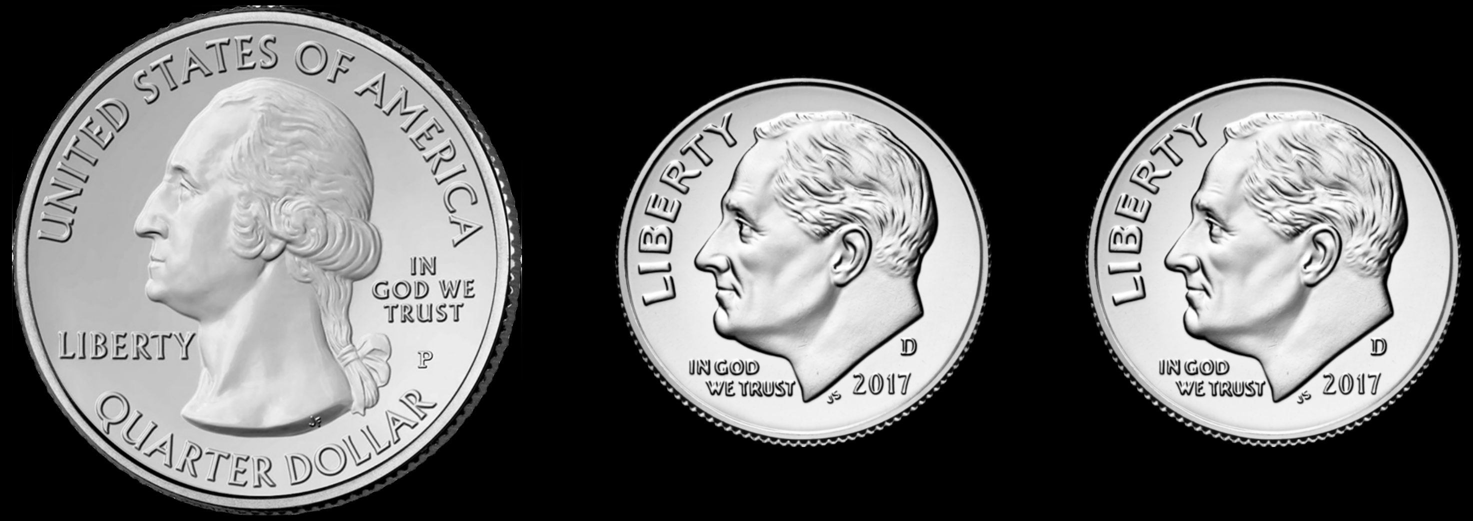
Make change for: **12¢**

# 47¢

1¢ 5¢ 10¢ 25¢

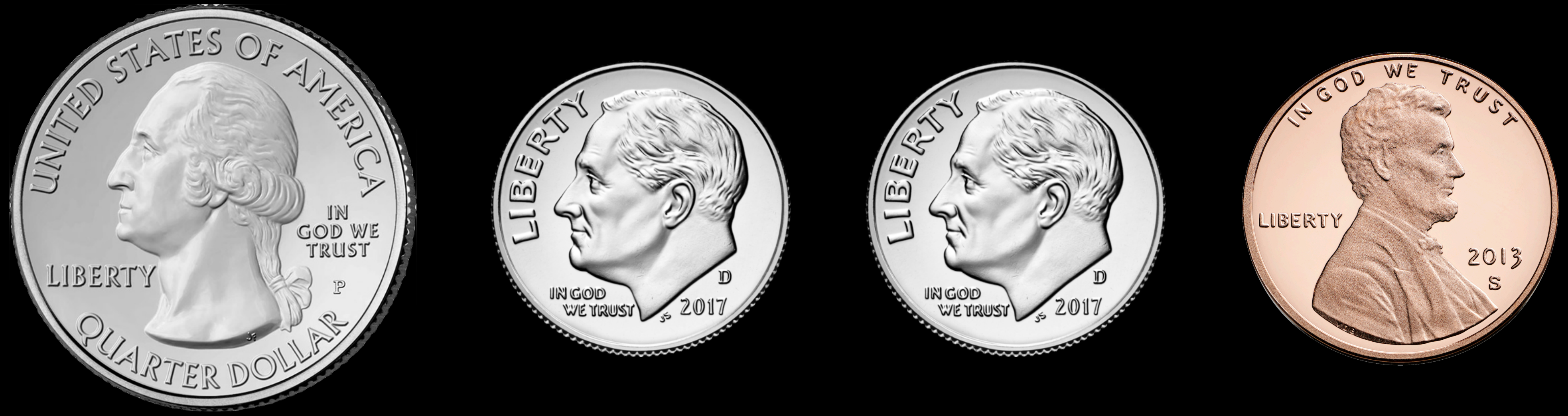
Make change for: **2¢**



47¢

1¢ 5¢ 10¢ 25¢

Make change for: **1¢**



# 47¢

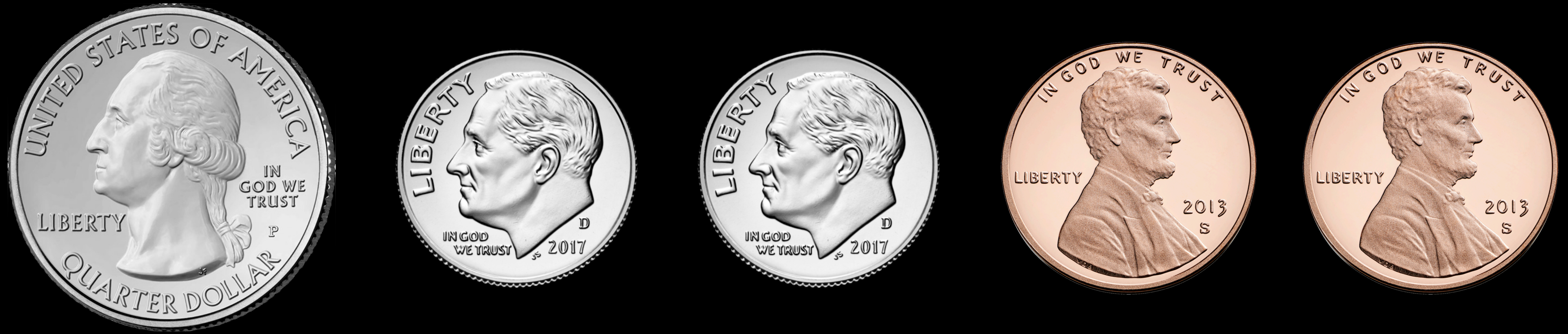
1¢

5¢

10¢

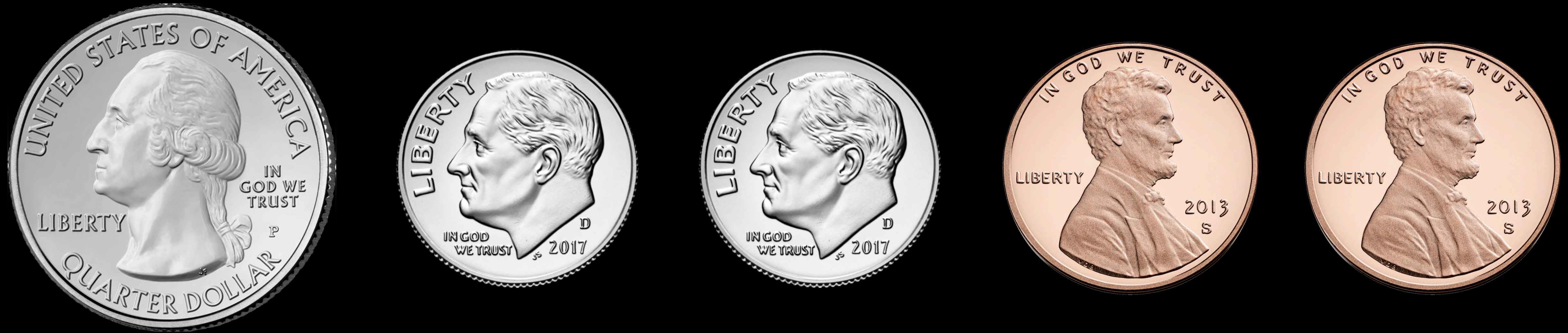
25¢

Make change for: **0¢**



47¢

1¢ 5¢ 10¢ 25¢



Coins used: **5**

# Cash

Write a program **cash.py** that prompts the user for how much change is owed and then prints the minimum number of coins with which that change can be made.

## Cash

```
$ python cash.py
Changed owed: 0.28
4
```

## Cash

Write a program **cash.py** that prompts the user for how much change is owed and then prints the minimum number of coins with which that change can be made.

## Phonebook

## CSV

| name | number |
|---|---|
| David | 617-555-0100 |
| Brian | 617-555-0101 |
| Jane | 617-555-0102 |
| Arturo | 617-555-0103 |
| Rodrigo | 617-555-0104 |
| Diana | 617-555-0105 |
| Heemyung | 617-555-0106 |
| Josh | 617-555-0107 |

```
name,number
David,617-555-0100
Brian,617-555-0101
Jane,617-555-0102
Arturo,617-555-0103
Rodrigo,617-555-0104
Diana,617-555-0105
Heemyung,617-555-0106
Josh,617-555-0107
```

```
import csv

f = open("phonebook.csv")
reader = csv.DictReader(f)
for row in reader:
    print(row["name"])
    print(row["number"])
```

## Phonebook

- Write a program **phonebook.py** that opens phonebook.csv and prints, one line per person, a sentence like:

  David's phone number is 617-555-0100

## Phonebook

- Write a program **phonebook.py** that prompts the user to type in a name.
- If the name is in phonebook.csv, the program should print that person's phone number.
- Otherwise, the program should print "Not Found."

## Phonebook

- Write a program **phonebook.py** that accepts a name as a command line argument.
- If the name is in phonebook.csv, the program should print that person's phone number.
- Otherwise, the program should print "Not Found."

## Readability

## Readability

```
$ python readability.py

Text: Congratulations! Today is your
 day. You're off to Great Places!
 You're off and away!

 Grade 3
```

# Readability

```
$ python readability.py

Text: Harry Potter was a highly unusual boy in
many ways. For one thing, he hated the summer
holidays more than any other time of year. For
another, he really wanted to do his homework but
was forced to do it in secret, in the dead of
night. And he also happened to be a wizard.

Grade 5
```

# Observations

- Longer words means higher reading level.
- More words per sentence means higher reading level.

# Coleman-Liau Index

- `index = 0.0588 * L - 0.296 * S - 15.8`
- `L` is average number of letters per 100 words.
- `S` is average number of sentences per 100 words.

## Letters, Words, Sentences

- A letter is any lowercase a-z or uppercase A-Z.
- Words are separated by spaces.
- Sentences end with periods, exclamation points, or question marks.

# Readability

Write a program **readability.py** that prompts the user for text and calculates the grade level of the text.

Use the Coleman-Liau Index:

```
0.0588 * L - 0.296 * S - 15.8
```

- `L` is average number of letters per 100 words.
- `S` is average number of sentences per 100 words.

## Strings

```
name = input("Name: ")

print("First letter of your name:")

print(name[0])
```

## Strings

```
name = input("Name: ")

print("Letters in your name:")

print(len(name))
```

## Strings

```
name = input("Name: ")
for letter in name:
    print(letter)
```

# Readability

Write a program **readability.py** that prompts the user for text and calculates the grade level of the text.

Use the Coleman-Liau Index:

```
0.0588 * L - 0.296 * S - 15.8
```

- `L` is average number of letters per 100 words.
- `S` is average number of sentences per 100 words.

CS50 for JDs