

```
1 # Prints all favorites in CSV using csv.reader
2
3 import csv
4
5 # Open CSV file
6 with open("favorites.csv", "r") as file:
7
8     # Create reader
9     reader = csv.reader(file)
10
11     # Skip header row
12     next(reader)
13
14     # Iterate over CSV file, printing each favorite
15     for row in reader:
16         print(row[1])
```

```
1 # Stores favorite in a variable
2
3 import csv
4
5 # Open CSV file
6 with open("favorites.csv", "r") as file:
7
8     # Create reader
9     reader = csv.reader(file)
10
11     # Skip header row
12     next(reader)
13
14     # Iterate over CSV file, printing each favorite
15     for row in reader:
16         favorite = row[1]
17         print(favorite)
```

```
1 # Prints all favorites in CSV using csv.DictReader
2
3 import csv
4
5 # Open CSV file
6 with open("favorites.csv", "r") as file:
7
8     # Create DictReader
9     reader = csv.DictReader(file)
10
11     # Iterate over CSV file, printing each favorite
12     for row in reader:
13         favorite = row["language"]
14         print(favorite)
```

```
1 # Prints all favorites in CSV using csv.DictReader
2
3 import csv
4
5 # Open CSV file
6 with open("favorites.csv", "r") as file:
7
8     # Create DictReader
9     reader = csv.DictReader(file)
10
11     # Iterate over CSV file, printing each favorite
12     for row in reader:
13         print(row["language"])
```

```
1 # Counts favorites using variables
2
3 import csv
4
5 # Open CSV file
6 with open("favorites.csv", "r") as file:
7
8     # Create DictReader
9     reader = csv.DictReader(file)
10
11     # Counts
12     scratch, c, python = 0, 0, 0
13
14     # Iterate over CSV file, counting favorites
15     for row in reader:
16         favorite = row["language"]
17         if favorite == "Scratch":
18             scratch += 1
19         elif favorite == "C":
20             c += 1
21         elif favorite == "Python":
22             python += 1
23
24 # Print counts
25 print(f"Scratch: {scratch}")
26 print(f"C: {c}")
27 print(f"Python: {python}")
```

```
1 # Counts favorites using dictionary
2
3 import csv
4
5 # Open CSV file
6 with open("favorites.csv", "r") as file:
7
8     # Create DictReader
9     reader = csv.DictReader(file)
10
11     # Counts
12     counts = {}
13
14     # Iterate over CSV file, counting favorites
15     for row in reader:
16         favorite = row["language"]
17         if favorite in counts:
18             counts[favorite] += 1
19         else:
20             counts[favorite] = 1
21
22 # Print counts
23 for favorite in counts:
24     print(f"{favorite}: {counts[favorite]}")
```

```
1 # Sorts favorites by key
2
3 import csv
4
5 # Open CSV file
6 with open("favorites.csv", "r") as file:
7
8     # Create DictReader
9     reader = csv.DictReader(file)
10
11     # Counts
12     counts = {}
13
14     # Iterate over CSV file, counting favorites
15     for row in reader:
16         favorite = row["language"]
17         if favorite in counts:
18             counts[favorite] += 1
19         else:
20             counts[favorite] = 1
21
22 # Print counts
23 for favorite in sorted(counts):
24     print(f"{favorite}: {counts[favorite]}")
```

```
1 # Sorts favorites by value
2
3 import csv
4
5 # Open CSV file
6 with open("favorites.csv", "r") as file:
7
8     # Create DictReader
9     reader = csv.DictReader(file)
10
11     # Counts
12     counts = {}
13
14     # Iterate over CSV file, counting favorites
15     for row in reader:
16         favorite = row["language"]
17         if favorite in counts:
18             counts[favorite] += 1
19         else:
20             counts[favorite] = 1
21
22 def get_value(language):
23     return counts[language]
24
25 # Print counts
26 for favorite in sorted(counts, key=get_value, reverse=True):
27     print(f"{favorite}: {counts[favorite]}")
```



```
1 # Sorts favorites by value using lambda function
2
3 import csv
4
5 # Open CSV file
6 with open("favorites.csv", "r") as file:
7
8     # Create DictReader
9     reader = csv.DictReader(file)
10
11     # Counts
12     counts = {}
13
14     # Iterate over CSV file, counting favorites
15     for row in reader:
16         favorite = row["language"]
17         if favorite in counts:
18             counts[favorite] += 1
19         else:
20             counts[favorite] = 1
21
22 # Print counts
23 for favorite in sorted(counts, key=lambda language: counts[language], reverse=True):
24     print(f"{favorite}: {counts[favorite]}")
```

```
1 # Favorite problem instead of favorite language
2
3 import csv
4
5 # Open CSV file
6 with open("favorites.csv", "r") as file:
7
8     # Create DictReader
9     reader = csv.DictReader(file)
10
11     # Counts
12     counts = {}
13
14     # Iterate over CSV file, counting favorites
15     for row in reader:
16         favorite = row["problem"]
17         if favorite in counts:
18             counts[favorite] += 1
19         else:
20             counts[favorite] = 1
21
22 # Print counts
23 for favorite in sorted(counts, key=lambda language: counts[language], reverse=True):
24     print(f"{favorite}: {counts[favorite]}")
```

```
1 # Favorite problem instead of favorite language
2
3 import csv
4
5 # Open CSV file
6 with open("favorites.csv", "r") as file:
7
8     # Create DictReader
9     reader = csv.DictReader(file)
10
11     # Counts
12     counts = {}
13
14     # Iterate over CSV file, counting favorites
15     for row in reader:
16         favorite = row["problem"]
17         if favorite in counts:
18             counts[favorite] += 1
19         else:
20             counts[favorite] = 1
21
22 # Print count
23 favorite = input("Favorite: ")
24 if favorite in counts:
25     print(f"{favorite}: {counts[favorite]}")
```

```
1 # Searches database popularity of a problem
2
3 import csv
4
5 from cs50 import SQL
6
7 # Open database
8 db = SQL("sqlite:///favorites.db")
9
10 # Prompt user for favorite
11 favorite = input("Favorite: ")
12
13 # Search for title
14 rows = db.execute("SELECT COUNT(*) FROM favorites WHERE problem LIKE ?", "%" + favorite + "%")
15
16 # Get first (and only) row
17 row = rows[0]
18
19 # Print popularity
20 print(row["COUNT(*)"])
```