Practice Problem 2: Testing Input and Output

5 minutes, 10 points
Filename: prac02 *(e.g. prac02.c, prac02.cpp, prac02.java, prac02.py2, prac02.py3)*

Description
This problem will ensure that your program is able to read input (STDIN) from the scoring website, and that the scoring website is able to read output (STDOUT) from your program.

The first line of input will include a single integer that indicates how many additional lines of input need to be read. Each additional line of input will contain two words separated by a space.

For each line of input containing NOUN and VERB, output a sentence constructed as “NOUN is VERB today!” Be sure that your output includes the **punctuation** shown in the sample output.

Sample Input

```
3
Mitchell coding
Danese testing
Larry debugging
```

Sample Output

```
Mitchell is coding today!
Danese is testing today!
Larry is debugging today!
```

Hints
Need help writing to Standard Output? See the Sample Solutions for this problem on the next page.

Learn More
Mitchell Baker was instrumental in the creation of the Mozilla Foundation, the organization that oversees development of the Firefox browser and other products. Today, she is chairperson of both the Mozilla Foundation and Mozilla Corporation.

Danese Cooper currently serves as chairperson of the Node.js Foundation. She also works in an open source role at PayPal. Previously, she served as CTO of the Wikimedia Foundation.

Larry Augustin is CEO of SugarCRM. He helped coin the term “Open Source”. In 1993 he founded VA Linux (now SourceForge, NASDAQ:LNUX), where he served as CEO until 2002.

Practice Problem 2: Sample Solutions

See below for **Practice Problem 2** solutions written in **Java, Python 2, Python 3, C, and C++**.

Each solution demonstrates how to read integers and strings from Standard Input (STDIN) and output strings to Standard Output (STDOUT). **Each solution has been tested with our scoring system.**

Each solution also demonstrates how to compile the program and how to read input from a text file during runtime. When using the following **Input**, each program will generate the following **Output**.

Sample Input (prac02.in)

```
3
Mitchell coding
Danese testing
Larry debugging
```

Sample Output

```
Mitchell is coding today!
Danese is testing today!
Larry is debugging today!
```

Sample Java Code (prac02.java)

```
import java.io.*;

public class prac02 {
    public static void main(String[] args) throws IOException {
        BufferedReader reader
            = new BufferedReader(new InputStreamReader(System.in));

        int numCases = Integer.parseInt(reader.readLine());

        while (numCases > 0) {
            --numCases;
            String line[] = reader.readLine().trim().split("\\s+");
            String noun = line[0];
            String verb = line[1];
            System.out.println(noun + " is " + verb + " today!");
        }
    }
}
```

Compile, Run with Input File

```
javac prac02.java
java prac02 < prac02.in
```

Sample Python2 Code (prac02.py2)

```
import sys

lines = sys.stdin.readlines()
num_cases = int(lines[0])

case = 1
while case < num_cases + 1:
    noun, verb = lines[case].split()
    print "%s is %s today!" % (noun, verb)
    case = case + 1
```

Run with Input File

```
python2 prac02.py2 < prac02.in
```

Sample Python3 Code (prac02.py3)

```
import sys

lines = sys.stdin.readlines()
num_cases = int(lines[0])

case = 1
while case < num_cases + 1:
    noun, verb = lines[case].split()
    print("%s is %s today!" % (noun, verb))
    case = case + 1
```

Run with Input File

```
python3 prac02.py3 < prac02.in
```

Sample C Code (prac02.c)

```
#include <stdio.h>

int main(int argc, char **argv) {
    int numCases;
    char noun[99 + 1]; /* + '\0' */
    char verb[99 + 1]; /* + '\0' */

    scanf("%d", &numCases);
    while (numCases) {
        --numCases;
        scanf("%s %s", noun, verb);
        printf("%s is %s today!\n", noun, verb);
    }
    return 0;
}
```

We recommend using the C11 standard. Be sure to return 0 in your code. A missing or different return statement might result in invalid exit codes that the scoring system interprets as a Runtime Error.

Compile, Run with Input File

```
gcc prac02.c -o prac02
./prac02 < prac02.in
```

Sample C++ Code (prac02.cpp)

```
#include <iostream>
#include <string>

using namespace std;

int main(int argc, char **argv) {
    int numCases;
    string name, verb;

    cin >> numCases;
    while (numCases) {
        --numCases;
        cin >> name >> verb;
        cout << name << " is " << verb << " today!" << endl;
    }
    return 0;
}
```

We recommend using the C++14 standard. Be sure to return 0 in your code. A missing or different return statement might result in invalid exit codes that the scoring system interprets as a Runtime Error.

Compile, Run with Input File

```
g++ prac02.cpp -o prac02
./prac02 < prac02.in
```