

Assignment 4

Author: Alex Clemmer

Student number: u0458675

Problem 0:

```
22.2222.22
```

```
Found a decimal number 22.2222 of value 22.222200
```

```
.Found an integer 22 of value 22
```

The RE `[0-9]+\.[0-9]+` looks for a decimal number with at least one leading digit; the second period does not fit in either RE, but the number after it does, so the RE picks it up as an integer.

```
22.22.22.22
```

```
Found a decimal number 22.22 of value 22.220000
```

```
.Found a decimal number 22.22 of value 22.220000
```

More or less the same case as above, except the second number can be parsed as a decimal, so it is.

```
u999999988888u2222u123456789"hi"(2222)(222)222-2222
```

```
uFound an integer 999999988888 of value 1316023800
```

```
uFound an integer 22222 of value 22222
```

```
uFound an integer 123456789 of value 123456789
```

```
"hi"(Found an integer 2222 of value 2222
```

```
)(Found an integer 222 of value 222
```

```
)Found an integer 222 of value 222
```

```
-Found an integer 2222 of value 2222
```

The 'u' characters are not parsed, and neither is the string "hi" or the '(' or ')' characters, and so the output really ends up being a series of integers that are sandwiched between them.

```
(999)999-2222u222222222"hi""hi"33.33
```

```
(Found an integer 999 of value 999
```

```
)Found an integer 999 of value 999
```

```
-Found an integer 2222 of value 2222
```

```
uFound an integer 222222222 of value 222222222
```

```
"hi""hi"Found a decimal number 33.33 of value 33.330000
```

Once again, the non-numeric characters do not fit in the REs, and so the output is a series of integers, with the other characters sitting between them.

```
"22""22.22"22.22 22 ""
```

```
"Found an integer 22 of value 22
```

```
"Found a decimal number 22.22 of value 22.220000
```

```
"Found a decimal number 22.22 of value 22.220000
```

```
Found an integer 22 of value 22
```

```
""
```

A similar case to all above, the numbers again are the only things that go through. The other characters end up simply output before the line.

Problem 1:

A NOTE ABOUT THIS REGEX: there isn't actually a well-formed uNID (e.g., one that does not have more than seven digits), so I just assumed you wanted me to extract all possible uNIDs. I can produce the regex that only finds well-formed.

```
Found a decimal number 22.2222 of value 22.222200
.Found an integer 22 of value 22
```

```
Found a decimal number 22.22 of value 22.220000
.Found a decimal number 22.22 of value 22.220000
```

```
Found a uID number u9999999
Found an integer 888888 of value 888888
uFound an integer 22222 of value 22222
Found a uID number u1234567
Found an integer 89 of value 89
"hi"(Found an integer 2222 of value 2222
)(Found an integer 222 of value 222
)Found an integer 222 of value 222
-Found an integer 2222 of value 2222
```

As you can see, we are clearly extracting all possible uNIDs from the corpus, while rejecting possible uNIDs that have, e.g., less than 7 digits after the initial [uU]. This is also true for the following statements:

```
(Found an integer 999 of value 999
)Found an integer 999 of value 999
-Found an integer 2222 of value 2222
Found a uID number u2222222
Found an integer 2 of value 2
"hi""hu"Found a decimal number 33.33 of value 33.330000

"Found an integer 22 of value 22
""Found a decimal number 22.22 of value 22.220000
"Found a decimal number 22.22 of value 22.220000
Found an integer 22 of value 22
""
```

Problem 2:

As you can see, even empty strings are included in this rule, and there are no false alarms.

```
Found a decimal number 22.2222 of value 22.222200
.Found an integer 22 of value 22
```

```
Found a decimal number 22.22 of value 22.220000
.Found a decimal number 22.22 of value 22.220000
```

```

Found a uID number u9999999
Found an integer 888888 of value 888888
uFound an integer 22222 of value 22222
Found a uID number u1234567
Found an integer 89 of value 89
Found a string "hi"
(Found an integer 2222 of value 2222
)(Found an integer 222 of value 222
)Found an integer 222 of value 222
-Found an integer 2222 of value 2222

(Found an integer 999 of value 999
)Found an integer 999 of value 999
-Found an integer 2222 of value 2222
Found a uID number u2222222
Found an integer 2 of value 2
Found a string "hi"
Found a string ""
Found a string "hu"
Found a decimal number 33.33 of value 33.330000

Found a string "22"
Found a string "22.22"
Found a decimal number 22.22 of value 22.220000
Found an integer 22 of value 22
Found a string ""

```

Problem 3:

As we can see, the well-formed phone numbers are clearly intact, with things that *prima facie* may look like phone numbers, but aren't, are clearly ignored and interpreted as, e.g., integers inside parentheses.

```

Found a decimal number 22.2222 of value 22.222200
.Found an integer 22 of value 22

Found a decimal number 22.22 of value 22.220000
.Found a decimal number 22.22 of value 22.220000

Found a uID number u9999999
Found an integer 888888 of value 888888
uFound an integer 22222 of value 22222
Found a uID number u1234567
Found an integer 89 of value 89
Found a string "hi"
(Found an integer 2222 of value 2222
)Found a phone number (222)222-2222

Found a phone number (999)999-2222
Found a uID number u2222222

```

```
Found an integer 2 of value 2
Found a string "hi"
Found a string ""
Found a string "hu"
Found a decimal number 33.33 of value 33.330000

Found a string "22"
Found a string "22.22"
Found a decimal number 22.22 of value 22.220000
  Found an integer 22 of value 22
  Found a string ""
```

1.3.1: