

HOMEWORK #1:

My First Lexical Analyser

Due Date: Wednesday, February 14th, 11:59.59pm

Description:

Lexemes of a typical programming language are of different token types. Consider the following tokens types:

- **Integers** are non-empty sequences of digits optionally preceded with either a '+' or '-' sign.
- **Decimal** numbers are Integers followed by a '.', followed by a non-empty sequence of digits. (e.g. 3.14, 00.01, 123.0).
- **Scientific** numbers are Decimal numbers followed by the character 'E', followed by a **non-zero** integer. (e.g. 12.0E4, 1.23E-6).
- **Hexadecimal** numbers are non-empty sequences of digits or the characters 'A', 'B', 'C', 'D', 'E' or 'F' followed by the suffix 'H'. (e.g. 12AD0H, 123H, 1A2B3CH,).
- **Keywords** are specific strings that form the language. For this homework we will consider the the following keywords: 'while', 'else', 'if', and 'print'.
- **Identifiers** are strings that consists of a letter followed by zero or more letters, digits or the underscore; and that are **not** hexadecimal numbers (e.g. x, size, name, p3, r_val).
- **Phone Numbers** are sequences of 10 digits separated in one of the following formats: *ddd.ddd.dddd* , *(ddd)ddd-ddd* or *ddd-ddd-ddd* where *d* is a single digit. (e.g. 555.923.0100, 101-555-1111, (123)456-7890).

Write a hand-coded Lexical Analyzer to recognize Integers, Decimal numbers, Scientific numbers, Hexadecimal Numbers, Keywords, Identifiers and Phone numbers. **Manually** encode the respective automata using the **automata encoding** techniques used in class. **Do not** use flex. **Do not** use a pre-existing regular expression library.

Input & Output:

Your Lexical Analyser should read input from standard input, and output to standard output.

The first line of the input will be a positive integer N, followed by N strings to recognize, one per line.

The first line of the output should echo the number of input lines N. For every line of input, your program should output the line number and state if the string is recognized as either a Keyword, an Identifier, an Integer, a Decimal number, a Scientific number, a Hexadecimal number, a Phone number, or an invalid string. Output **must be in the format** shown in the sample output.

Sample:

Input	Output
14	14
83462	1: Integer.
-39874.454	2: Decimal.
while	3: Keyword.
ABCH	4: Hexadecimal.
+234.34E-941	5: Scientific.
124.235.234	6: INVALID!
color	7: Identifier.
-1.23E-3.5	8: INVALID!
4.	9: INVALID!
+0	10: Integer.
(111)111-1111	11: Phone.
FFFF	12: Identifier.
for4	13: Identifier.
3dfx	14: INVALID!

Submission:

Submit through the UNIX systems using the command

```
'cssubmit 3500 <section> 1'.
```

Your program should be in a language and version available in the Computer

Science department's UNIX systems. In order to accommodate different languages, your submission should include a bash script named `run.sh` that compiles and runs your program with all the necessary options and commands. For example, the following would be a `run.sh` script for C++ 11.

```
#!/bin/bash
g++ -std=c++11 *.cpp -o hw1.ex
./hw1.ex
```

A `run.sh` script for Python3 if the program is called `hw1.py3`:

```
#!/bin/bash
python3 hw1.py3
```

Your program will be evaluated using the command:

```
./run.sh < testinput.txt.
```

Hint:

```
#include <iostream>
#include <string>
using namespace std;

int main ()
{
    int T;
    string s;

    cin >> T;
    for (int i=0; i<T; i++) {
        cin >> s;
        cout << "Hello " << s << "!" << endl;
    }
    return 0;
}
```