

Homework 1: Lexical Analyzer

CS 421 Compiler Design and Construction

Asad Tariq & Fahad Shaikh

Habib University

Fall 2022

Due: September 11, 2022

1. 1 point Consider the following C code snippet:

```
1  int w, x, y, z;  
2  int i = 4; int j = 5;  
3  {  
4      int j = 7;  
5      i = 6;  
6      w = i + j;  
7  }  
8  x = i + j;  
9  {  
10     int i = 8;  
11     y = i + j;  
12 }  
13 z = i + j;  
14
```

Listing 1: C code snippet.

Indicate the values assigned to **x** and **z** from Listing 1.

Solution: On examination of the given C code in Listing 1, the values of **x** and **z** are as follows:

- **x** = 11
- **z** = 11

2. 1 point What is printed as output when the following C code is executed?

```
1  #define a (x+1)  
2  int x = 2;  
3  void b() {x = a; printf("%d\n", x);}  
4  void c() {int x = 1; printf("%d\n", a);}  
5  void main() {b(); c();}  
6
```

Listing 2: C code snippet.

Solution: The output of the code in Listing 2 is as follows:

3
2

3. 1 point Construct a regular expression to recognize *currency* numbers in dollars. It should be a positive decimal number rounded to the nearest one-hundredth. Currency numbers begin with the dollar sign \$, have commas separating each group of three digits to the left of the decimal point, and end with two digits to the right of the decimal point, for example, \$8,937.43 and \$7,777,777.7.

Solution: For construction of the required regular expression, we shall be relying on Table 1.

EXPRESSION	MATCHES	EXAMPLE
\wedge	beginning of a line	$\wedge abc$
c	the one non-operator character c	,
$\backslash c$	character c literally	$\backslash \$$
$\$$	end of a line	$abc\$$
$\backslash d$	digit d	5
$r\{m, n\}$	between m and n occurrences of r	$a\{3, 5\}$
$r\{n\}$	strictly n occurrences of r	$a\{2\}$
$r?$	zero or one r	$a?$
r^*	zero or many strings matching r	a^*

Table 1: A modification of the ‘Lex regular expressions’ table from Section 3.3 (Figure 3.8) of the *Dragon Book*.

Now, using Table 1, we construct the following regular expression:

$$\wedge \backslash \$ \backslash d\{1, 3\} (, \backslash d\{3\})^* (\backslash . \backslash d \backslash d) \$$$

Breakdown:

- $\wedge \backslash \$$: The string should start with the ‘\$’ character.
- $\backslash d\{1, 3\}$: One, two or three digits.
- $(, \backslash d\{3\})^*$: A group of characters - a comma followed by exactly three digits - that may appear zero or many times.
- $(\backslash . \backslash d \backslash d) \$$: The final (i.e., *ending*) portion of the string should be a decimal point followed by two digits.