

COP4020
Project 4
cexpr

- Your assignment is to write a program using `yacc` (`bison`) and `lex` (`flex`) that implements a calculator of C integer expressions. The calculator will process expressions until it encounters EOF or invalid syntax. Each calculation is terminated by a semicolon. Tokens can be separated by whitespace (but no comments). After each calculation, the calculator prints its result. Tokens include integer numbers and 26 predefined integer variables. There will be one variable corresponding to each of the lowercase letters in the alphabet. The table below defines the C operators you need to implement. The operators are listed in order of decreasing precedence.

Symbols	Comments	Type	Associativity
()	Parentheses	Unary	n/a
++ --	Increment decrement	Postfix	Left
++ --	Increment decrement	Prefix	Right
~	Bitwise not	Unary	Right
-	Negation	Unary	Right
* / %	Mult, div, rem	Binary	Left
+ -	Add. Sub	Binary	Left
<< >>	Shifts	Binary	Left
&	Bitwise and	Binary	Left
^	Bitwise xor	Binary	Left
	Bitwise or	Binary	Left
=	Assignment	Binary	Right

- Yacc* (*bison*) does provide mechanisms for specifying the associativity and precedence of operators in an unambiguous grammar. However, you are not allowed to use these mechanisms. Instead you should define extra non-terminals and productions to enforce the specified associativity and precedence for this assignment.
- Rather than having a default value of zero for each of the predefined variables, you will treat the default variable as having an unknown value. If a variable's value is referenced without ever having been assigned a value, then the value of the expression should be designated as unknown. A variable's value becomes unknown when it is assigned an unknown value.
- In addition to calculating an expression, two other commands are supported. The first is to dump the values of the different variables when the keyword **dump** is detected. The second is to reset all of the values of the different variables to unknown when the keyword **reset** is encountered.
- You need to create files called `cexpr.y`, `scan.l` and `makefile` that will contain the parser, scanner and makefile. The `makefile` should make an executable called `cexpr.exe`. You should submit a single zip (tar) file that includes these three files, to Canvas project link.

- The reference executable (cexpr.exe) is placed in ~uh/cop4020/proj4/cexpr.exe for your info.

Example Input:

```
a=55-3;  
b=c=a-42;  
a+b*c;  
dump  
reset  
c=6;  
a=b; ^D
```

Example Output:

```
52  
10  
152  
a: 52  
b: 10  
c: 10  
d: unknown  
e: unknown  
f: unknown  
g: unknown  
h: unknown  
i: unknown  
j: unknown  
k: unknown  
l: unknown  
m: unknown  
n: unknown  
o: unknown  
p: unknown  
q: unknown  
r: unknown  
s: unknown  
t: unknown  
u: unknown  
V: unknown  
w: unknown  
x: unknown  
y: unknown  
Z: unknown  
6  
unknown  
Calculator off.
```