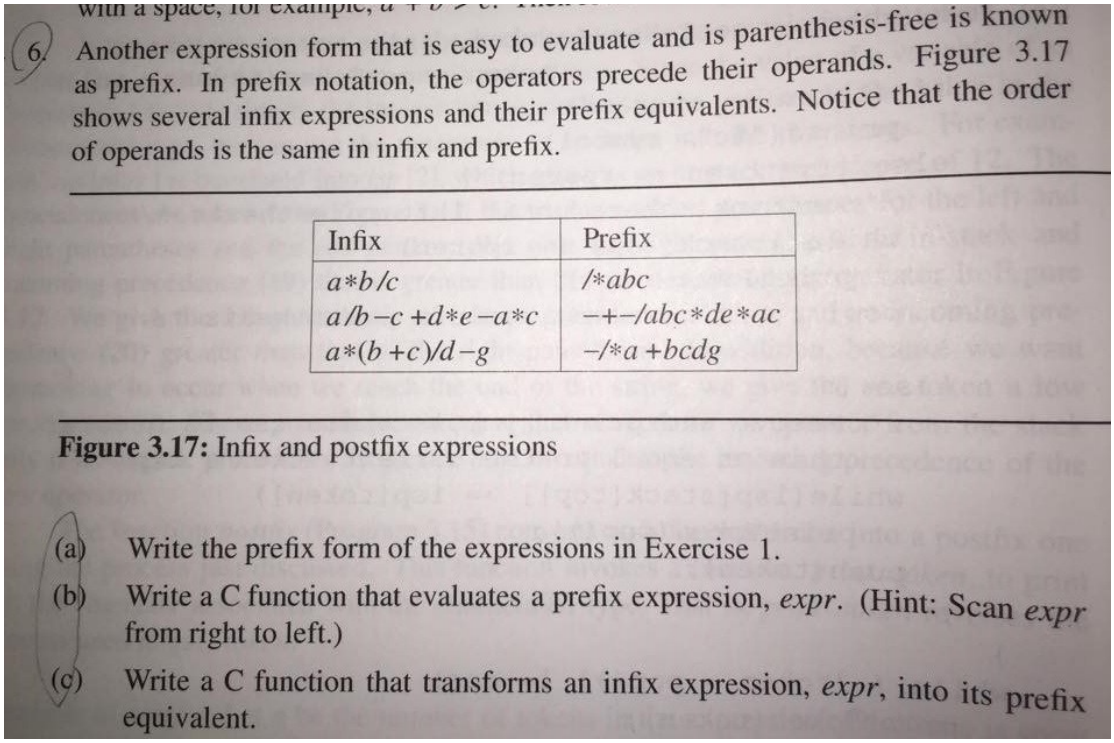
Data Structure Assignment 3(Paper Homework)

Paper homework

(Textbook p.138 Exercise 6(a))



with a space, for example, $a + b > c$. Then ...

6. Another expression form that is easy to evaluate and is parenthesis-free is known as prefix. In prefix notation, the operators precede their operands. Figure 3.17 shows several infix expressions and their prefix equivalents. Notice that the order of operands is the same in infix and prefix.

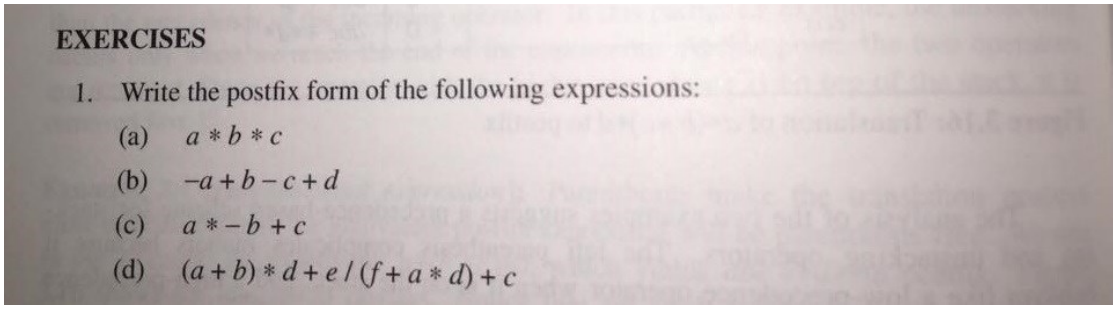
Infix	Prefix
$a*b/c$	$/*abc$
$a/b-c+d*e-a*c$	$-+-/abc*de*ac$
$a*(b+c)/d-g$	$-/*a+bcdg$

Figure 3.17: Infix and postfix expressions

(a) Write the prefix form of the expressions in Exercise 1.

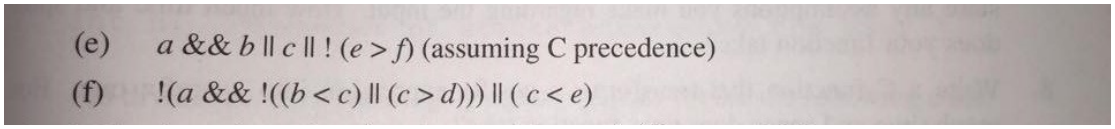
(b) Write a C function that evaluates a prefix expression, *expr*. (Hint: Scan *expr* from right to left.)

(c) Write a C function that transforms an infix expression, *expr*, into its prefix equivalent.



EXERCISES

1. Write the postfix form of the following expressions:
   (a) $a * b * c$
   (b) $-a + b - c + d$
   (c) $a * -b + c$
   (d) $(a + b) * d + e / (f + a * d) + c$



(e) $a \,\&\&\, b \,\|\, c \,\|\, !\,(e > f)$ (assuming C precedence)

(f) $!(a \,\&\&\, !((b < c) \,\|\, (c > d))) \,\|\, (c < e)$

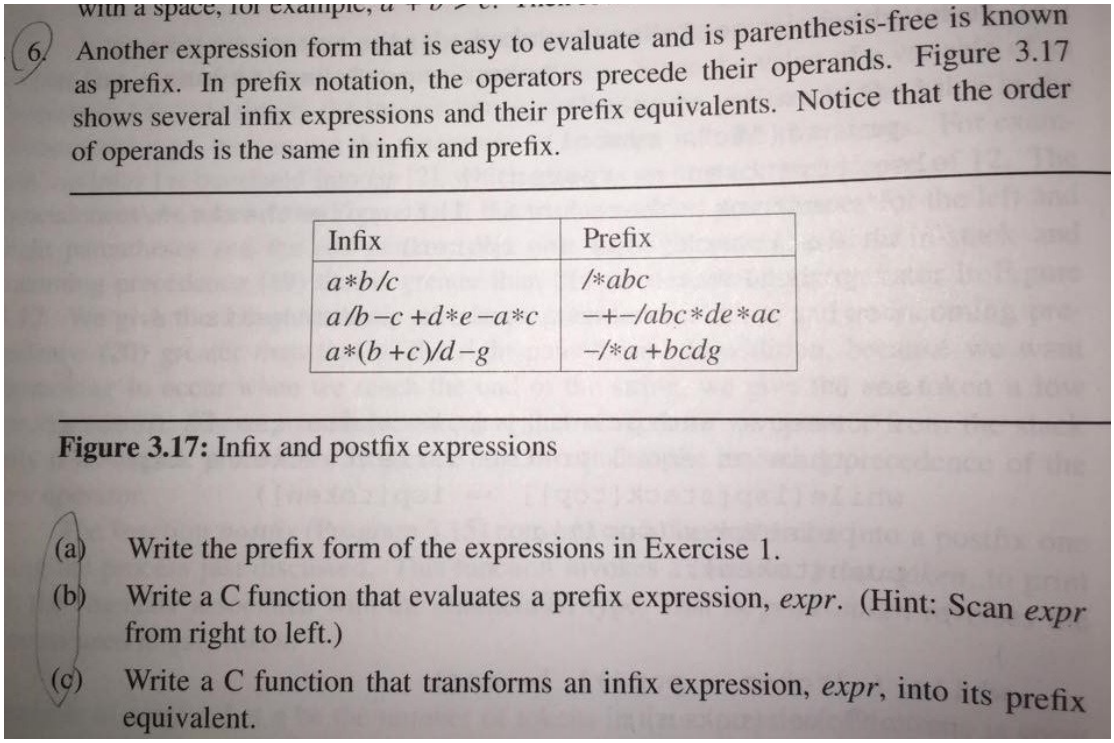
General Information

- **Deadline：2015/10/30(Please submit to TA after class)**
- **Late homework will not be accepted.**
- **Please write on A4 papers.**
- **Notice：You won't get any point if you only write the answer, please list your process and reason.**

Data Structure Assignment 3 (Programming Homework)

Programming homework I & II

(Textbook p.138 Exercise 6(b)(c))



with a space, for example, $a + b > c$. Then ...

6. Another expression form that is easy to evaluate and is parenthesis-free is known as prefix. In prefix notation, the operators precede their operands. Figure 3.17 shows several infix expressions and their prefix equivalents. Notice that the order of operands is the same in infix and prefix.

Infix	Prefix
$a*b/c$	$/*abc$
$a/b-c+d*e-a*c$	$-+-/abc*de*ac$
$a*(b+c)/d-g$	$-/*a+bcdg$

Figure 3.17: Infix and postfix expressions

(a) Write the prefix form of the expressions in Exercise 1.

(b) Write a C function that evaluates a prefix expression, *expr*. (Hint: Scan *expr* from right to left.)

(c) Write a C function that transforms an infix expression, *expr*, into its prefix equivalent.

(b)Sample Input

- + 1 *5 8 * 3 + 2 5

(b)Sample Output

20

(c)Sample Input

a+b*c/(d-e)

(c)Sample Output

Prefix : +a/*bc-de

General Information:

- **Deadline：2015/11/6 23:59.**
- **Upload your assignment to Moodle system.**
- **Upload file format: Student-Id_Name.rar , Ex.P76991094_王小明.rar**
- **Your file should consist of the following items:**
- **Source Code**
- **Readme file (Program description)**
- **Any copies will be scored as zero.Do not plagiarize!!!**