CS2134 Homework 7 Spring 2016

Today? 2016

Written Part:

---

1)

```
int x;
queue<int>q;
 //[ , , , ],,theFront=0,theBack=3, currentSize=0,x=?
q.push(1);
//[1 , , , ],,theFront=0,theBack=0, currentSize=1,x=?
q.push(2);
//[1 ,2 , , ],theFront=0,theBack=1, currentSize=2,x=?
q.push(3);
//[1 ,2 ,3 , ],theFront=0,theBack=3, currentSize=3,x=?
x = q.front();
//[1 ,2 ,3 , ],theFront=0,theBack=3, currentSize=3,x=1
q.pop();
//[ , 2, 3, ],theFront=1,theBack=3, currentSize=2,x=1
q.push(4);
//[ ,2 ,3 ,4 ],theFront=1,theBack=4, currentSize=3,x=1
x = q.front();
//[ ,2 ,3 ,4 ],theFront=1,theBack=4, currentSize=3,x=2
q.push(5);
//[5 ,2 ,3 ,4 ],theFront=1,theBack=0, currentSize=4,x=2
q.pop();
//[5 , ,3 ,4 ],theFront=2,theBack=0, currentSize=3,x=2
q.push(6);
//[5 ,6 ,3 ,4 ],theFront=2,theBack=1, currentSize=4,x=2
x = q.front();
//[5 ,6 ,3 ,4 ],theFront=2,theBack=1, currentSize=4,x=3
q.pop();
//[5 ,6 , ,4 ],theFront=3,theBack=1, currentSize=3,x=3
q.pop();
//[5 ,6 , , ],theFront=0,theBack=1, currentSize=2,x=3
q.pop();
//[ ,6 , , ],theFront=1,theBack=1, currentSize=1,x=3
x = q.front();
//[ ,6 , , ],theFront=1,theBack=1, currentSize=1,x=6
 q.pop();
//[ , , , ],theFront=2,theBack=1, currentSize=0,x=6
 q.push(7);
//[ , ,7 , ],theFront=2,theBack=1, currentSize=1,x=6
```

2) 1 - 2 + 3 ^ 2

symbol	Stack	Postfix
1		1
-	-	1
2	-	1 2
+	+	1 2 -
3	+	1 2 - 3
^	+^	1 2 - 3
2	+^	1 2 – 3 2
		1 2 – 3 2 + ^

(2^3)^2

symbol	Stack	Postfix
(	(	
2	(	2
^	(^	2
3	(^	2 3
)		2 3 ^
^	^	2 3 ^
2	^	2 3 ^ 2
		2 3 ^ 2 ^

2^3^2

symbol	Stack	Postfix
2		2
^	^	2
3	^	2 3
^	^	2 3 ^
2	^	2 3 ^ 2
		2 3 ^ 2 ^

(2 + 6) / 3 - (32 + 4 * 7) * 2

symbol	Stack	Postfix
(	(	
2	(	2
+	(+	2
6	(+	2 6
)		2 6 +
/	/	2 6 +
3	/	2 6 + 3
-	-	2 6 + 3 /
(	-(	2 6 + 3 /
32	-(	2 6 + 3 / 32
+	-(+	2 6 + 3 / 32
4	-(+	2 6 + 3 / 32 4

*	-(+*	2 6 + 3 / 32 4
7	-(+*	2 6 + 3 / 32 4 7
)	-	2 6 + 3 / 32 4 7 * +
*	-*	2 6 + 3 / 32 4 7 * +
2	-*	2 6 + 3 / 32 4 7 * + 2
		2 6 + 3 / 32 4 7 * + 2 * -

3+2-4+5

symbol	Stack	Postfix
3		3
+	+	3 2
2	+	3 2 +
-	-	3 2 +
4	-	3 2 + 4
+	+	3 2 + 4 -
5	+	3 2 + 4 – 5
		3 2 + 4 – 5 +

3) 42+33^-

Input	stack
4	4
2	4 2
+	6
3	6 3
3	6 3 3
^	6 27
-	-19

3 2 ^ 3 2 * -

Input	stack
3	3
2	3 2
^	9
3	9 3
2	9 3 2
*	9 6
-	3

4 2 3 * - 3 2 ^ - 6 +

Input	stack
4	4
2	4 2

3	4 2 3
*	4 6
-	-2
3	-2 3
2	-2 3 2
^	-2 9
-	-11
6	-11 6
+	-5

4 3 + 2 * 1 -

Input	stack
4	4
3	4 3
+	7
2	7 2
*	14
1	14 1
-	13

3 5 * 1 + 4 / 6 +

Input	stack
3	3
5	3 5
*	15
1	15 1
+	16
4	16 4
/	4
6	4 6
+	10

4)

```
enum TokenType { EOL, VALUE, OPAREN, CPAREN, EXP, MULT, DIV, MOD, PLUS, MINUS};
  // PREC_TABLE matches order of Token enumeration   struct Precedence   {
      int inputSymbol;
      int topOfStack;   };
  vector<Precedence>  PREC_TABLE  =   {

{0,-1},{0,0}, {100,0},{0,99}, {6,5},
{3,4}, {3,4}, {3,4}, {1,2}, {1,2},

};
```

5)

EXP
EXP
MULT
stack: [DIV,PLUS,EOL]

6)

(a) what is the value of the root node?

(b) which node is the sibling of 4?
-
(c) which nodes are leaf nodes?
4,5,8, and 3
(d) which nodes are internal nodes?
*,+, -
(e) what is the height of the node containing ‘-’?
1
(f) what is the depth of the node containing ‘-’?
2
(g) what is the size of the tree?
7
(h) which nodes are the children of the node containing ‘+’?
4 and -
(i) which node is the parent of the node containing ‘-’?
+