COS126: General Computer Science **Spring 1996**

Third Midterm Exam – Solution to Problem 4

Date: Thursday, May 16, 1996

Problem 4, 25 points:

Part a, 5 points: Give the parse tree for the expression

```
((((a*b)+c)*d)+(e+f))
                   +
                 /   \
                /     \
              *        +
             / \      / \
            +   d    e   f
          /   \
         *     c
       /  \
      a    b
```

Part b, 20 points: We wish to parse expressions for polynomials. An example of the input form is

```
ax5+bx2+c
```

The whole program with added code:

```
struct tree *polyparse(void)
{
 char coef;
 char var;
 char exp;
 struct tree *l;
 struct tree *r;

 coef = get("abcdefghijklmnopqrstuvwxyz");
 l = maketree(coef, NULL, NULL);
 if (look() == '\0')
     { r = maketree('0', NULL, NULL);
       return maketree('x', l, r);
     }
 var = get("x+");
```

```
  if (var == '+')
      { r = maketree('0', NULL, NULL);
        l = maketree('x', l, r);
      }
  else
      {
        if (look() == '\0')
            { r = maketree ('1', NULL, NULL);
              return maketree('x', l, r);
            }

  /*  AT THIS POINT x HAS BEEN SEEN AND WE HAVE CHECKED FOR END OF INPUT
      THE NEXT SYMBOL COULD BE THE EXPONENT OR +
      IF IT IS +, THE EXPONENT = 1, IF NOT THE EXPONENT IS EXPLICIT
      BUILD A RIGHT TREE FOR THE EXP
      CHECK IF ALL THE INPUT HAS BEEN SEEN.
      IF IT HAS, RETURN A TREE FOR THIS x TERM.
      IF THERE IS MORE INPUT, BUILD THE TREE FOR THIS x TERM POINTED
      TO BY l (AND USED AFTER THE RECURSIVE CALL) -- '+' MUST BE NEXT */

        exp = get("23456789+");
        if (exp == '+')
            { r = maketree('1', NULL, NULL);
              l = maketree('x', l, r);
            }
        else
            {
               r = maketree(exp, NULL, NULL);
               if (look() == '\0')
                   return maketree('x', l, r);
               l = maketree('x', l, r);
               get("+");
             }
      }
  r = polyparse();
  return maketree('+', l, r );
}
```