Problem for 2001 December

Proposed by Dan Jurca

Write a computer program (or design an algorithm) which one can use to display all ways of associating a product of n factors. (For example, if n=3, there are two ways- A(BC) and (AB)C.) Of course if the associative law holds, then all these ways of associating yield the same product. (This happens, for example, if the factors are matrices of such dimensions that the product exists.)

The attached output is from the proposer's program (written in C++) for the cases n=1,2,3,4,5,6.

It may be helpful to know that the number of ways of associating a product of n factors, say P_n, is given by

$$P_n = \begin{cases} 1 & \text{if } n=1, \\ \sum_{i=1}^{n-1} P_i P_{n-i} & \text{if } 1 < n; \end{cases}$$

$$= \frac{1}{n}\binom{2n-2}{n-1}, 1 \le n,$$

the (n−1)-th Catalan number.

```
1     1  A

2     1  AB

3     1  A(BC)
      2  (AB)C

4     1  A(B(CD))
      2  A((BC)D)
      3  (AB)(CD)
      4  (A(BC))D
      5  ((AB)C)D

5     1  A(B(C(DE)))
      2  A(B((CD)E))
      3  A((BC)(DE))
      4  A((B(CD))E)
      5  A(((BC)D)E)

6     1  A(B(C(D(EF))))
      2  A(B(C((DE)F)))
      3  A(B((CD)(EF)))
      4  A(B((C(DE))F))
      5  A(B(((CD)E)F))
      6  A((BC)(D(EF)))
      7  A((BC)((DE)F))
      8  A((B(CD))(EF))
      9  A(((BC)D)(EF))
     10  A((B(C(DE)))F)
     11  A((B((CD)E))F)
     12  A(((BC)(DE))F)
     13  A(((B(CD))E)F)
     14  A((((BC)D)E)F)
     15  (AB)(C(D(EF)))
     16  (AB)(C((DE)F))
     17  (AB)((CD)(EF))
     18  (AB)((C(DE))F)
```

```
 6  (AB)(C(DE))
 7  (AB)((CD)E)
 8  (A(BC))(DE)
 9  ((AB)C)(DE)
10  (A(B(CD)))E
11  (A((BC)D))E
12  ((AB)(CD))E
13  ((A(BC))D)E
14  (((AB)C)D)E
19  (AB)(((CD)E)F)
20  (A(BC))(D(EF))
21  (A(BC))((DE)F)
22  ((AB)C)(D(EF))
23  ((AB)C)((DE)F)
24  (A(B(CD)))(EF)
25  (A((BC)D))(EF)
26  ((AB)(CD))(EF)
27  ((A(BC))D)(EF)
28  (((AB)C)D)(EF)
29  (A(B(C(DE))))F
30  (A(B((CD)E)))F
31  (A((BC)(DE)))F
32  (A((B(CD))E))F
33  (A(((BC)D)E))F
34  ((AB)(C(DE)))F
35  ((AB)((CD)E))F
36  ((A(BC))(DE))F
37  (((AB)C)(DE))F
38  ((A(B(CD)))E)F
39  ((A((BC)D))E)F
40  (((AB)(CD))E)F
41  (((A(BC))D)E)F
42  ((((AB)C)D)E)F
```

Solution

```
/*      assoc.c         2001 dec 03     last modified 2002 jan 07
 *
 *      Returns in the string s the m-th way of associating a product
 *      of n factors, where the first factor is 'A', etc.
 *      Here 1 <= n <= MAXN, and 1 <= m <= p[n-1].
 */

#define MAXN 11

void
assoc( int n,unsigned int m,char *s )
{
        void assoc0( int n,unsigned int m,int,char * );

        *s = '\0';
        assoc0( n,m,0,s );
}
void
assoc0( int n,unsigned int m,int a,char *s )
{
        void appendc( char *,char ),appends( char *,char * );

        unsigned int i,j,k,p[]={1,1,2,5,14,42,132,429,1430,4862,16796,58786},
                q,r,s0,s1;
        char t[3*(MAXN-1)];

        if ( n < 3 ) {
                appendc( s,'A'+a );                     // n == 1 or n == 2
                if ( 1 < n ) {
```

```
                        appendc( s,'B'+a );             // n == 2
                }
                return;
        }
        for ( s1 = 0, i = 1;  i < n;  i++ ) {
                s0 = s1;
                s1 += p[i-1]*p[n-i-1];
                if ( m <= s1 ) {
                        break;
                }
        }
        q = (m-s0)/(j = p[n-i-1]);
        r = (m-s0)-q*j;
        if ( r == 0 ) {
                --q;
                r = j;
        }
        if ( 1 < i ) {
                appendc( s,'(' );
        }
        *t = '\0';
        assoc0( i,q+1,a,t );
        appends( s,t );
        if ( 1 < i ) {
                appendc( s,')' );
        }
        if ( 1 < n-i ) {
                appendc( s,'(' );
        }
        *t = '\0';
        assoc0( n-i,r,a+i,t );
        appends( s,t );
        if ( 1 < n-i ) {
                appendc( s,')' );
        }
}
void
appendc( char *s,char ch )
{
        while ( *s ) {
                s++;
        }
        *s++ = ch;
        *s = '\0';
}
void
appends( char *s,char *t )
{
        while ( *s ) {
                s++;
        }
        while ( *t ) {
                *s++ = *t++;
        }
        *s = '\0';
}
```