match no. 28 pp. 61-99 1992

MOLECULAR TOPOLOGY 11[1]. Y - INDICES IN HOMOGENEOUS DENDRIMERS*

Mircea V. Diudea**), Irén E. Kacsó and Ovidiu M. Minailiuc
Department of Chemistry, "Babes-Bolyai" University
Arany János Str., 11, 3400 CLUJ, ROMANIA

(received: February 1992)

Abstract

Homogeneous dendrimers are described by the aid of matrices F (layer matrix of vertex neighbourhood) and B (layer matrix of vertex degrees). From the partition in F corresponding to the dendrimer core (p=1) general relations for binary dendrimers A_xB_y are derived and exemplified as "paper" chemical molecules. The Y-indices constructed on the ground of the two layer matrices of dendrimers are used in a compactness criterion.

1. Introduction

As the branching is a natural modality to promote the tridimensionality (by the mean of rotation around the axis joining two nodes) in branched molecules, trees or corals, this intuitive property has gained a special attention in mathematical chemistry literature [2 - 12]. The branching question was primarily tackled in connection with the structure - property cor-

* This work is dedicated to the 60th aniversary of professor Alexandru T. Balaban, Polytechnic Institut Bucharest, Romania.
**) To whom all correspondence should be addressed.

relations in alkane - trees. This yielded in several graph invariants (topological indices), devised with the declared aim to express the branching (see [7, 8, 13 - 15]).
In a previous paper [12] we proposed a new index, BY, based on the B matrix (the layer matrix of vertex degrees) which proved a good ability to describe the molecular branching :

$$MY(G) = \sum_i \sum_j m_{ij} * 10^{1-j} \qquad (1)$$

$$m,M = b,B \quad [1,12]$$

$$m,M = f,F \quad [\text{this work}]$$

where m_{ij} are the entries in the F or B matrices and are defined as :

$$m_{ij} = \sum_{k,d_{ik}=j-1} m_k \qquad (2)$$

with m_k being a property (for f_k - the vertex neighbourhood and for b_k - the vertex degrees) of vertices on a shell around the vertex i, at a given distance, $d_{ik}=j-1$

Very recently a new class of hyperbranched polymers has attracted the attention of the synthesists as well as the theorists. In this regard, a very extensive review was published by Tomalia et al. [11]. The so called dendrimers can be prepared either by divergent multiplicative growthof a central core [11, 16] or by starting from the periphery and converging inward [10]. Such "starburst" structures are expected to possess novel properties [10, 11, 17].

In this paper we describe the dendritic graphs by the means of F (layer matrix of vertex neighbourhood) [18] and B matrices, and on this ground, we propose general formulae for mono- and di-centric binary dendrimers, A_xB_y.

Recursive relations for computing the MY - indices in the case of homogeneous dendrimers are derived and a compactness criterion for alkanes is proposed.

2. Layer matrices F and B in homogeneous dendrimers

Let's consider a dendrimer (fig.1,a) in which all branching nodes, A, possess the same degree (k=3) and the endgroups, B, to be monovalent (and therefore suppressible) groups. The vertex layers (or shells) surround the center(s) succesively, at the progressive edge - radius,.er. The endgroups, B, are characterized by the radius of their pattern nodes, A; these last nodes are to be considered of graph theoretical degree, k=1. Since a tree possesses either a monocenter or a dicenter [19], the dicentric dendritic graphs (fig.1,b) must be also considered.

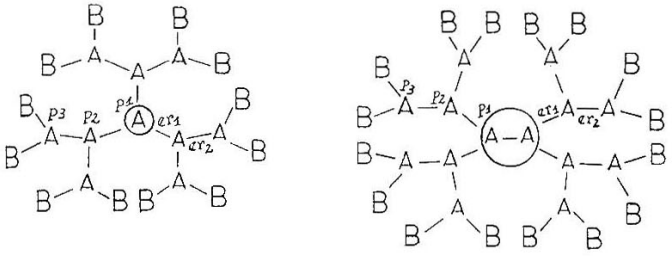


Fig.1. Monocentric (a) and dicentric (b) (k=3) dendrimers.

In the layer matrix description, any branching vertex A, becomes the center of its own relative partition (see [18]). Thus, we can write the general formula for the rows (actually partitions p of the matrices F and B, starting from the center(s)

to the periphery :

Monocentric dendrimers

F - matrix

$$p = 1 \qquad 1 \cdot \left\| k(k-1)^{j-2} \right\|_{j \in [2,\ ed/2+1]} \qquad (3)$$

$$p \in [2, ed/2] \qquad k(k-1)^{p-2} \cdot \left\| k(k-1)^{j-2}, \ (k-1)^{j-(x+1)} \right\|_{j \in [2, ed/2-p+2]}, \ {}_{j=ed/2-p+(2x+1)}^{x \in [1,(p-1)]}$$

$$\left\| (k-1)^{j-x} \right\|_{j=ed/2-p+2x,\ x \in [2,p]} \qquad (4)$$

$$p = ed/2+1 \qquad k(k-1)^{ed/2-1} \cdot \left\| (k-1)^{j-(x+1)}, \ (k-1)^{j-x} \right\| \qquad (5)$$

$$\text{with } {}_{j=2x,\ x \in [1,ed/2]} \text{ and } {}_{j=2x-1,\ x \in [2,ed/2+1]}$$

B - matrix

$$p = 1 \qquad 1 \cdot \left\| k, \ k^2(k-1)^{j-2}, \ k(k-1)^{j-2} \right\| \qquad (6)$$

$$\text{with } {}_{j \in [2,ed/2]} \text{ and } {}_{j=ed/2+1}$$

$$p \in [2, ed/2] \qquad k(k-1)^{p-2} \cdot \left\| k, \ k^2(k-1)^{j-2}, \ (2k-1)(k-1)^{j-2}, \right.$$

$$\text{with } {}_{j \in [2,ed/2-p+1]} \text{ and } {}_{j=ed/2-p+2}$$

$$\left. k(k-1)^{j-(x+1)}, \ 2(k-1)^{j-x}, \ (k-1)^{j-x} \right\| \qquad (7)$$

$$\text{with } {}_{j=ed/2-p+(2x+1),\ x \in [1,(p-1)]}, \ {}_{j=ed/2-p+2x,\ x \in [2,(p-1)]}, \ {}_{j=ed/2+p,\ x=p}$$

$$p = ed/2+1 \qquad k(k-1)^{ed/2-1} \cdot \left\| 1, \ k(k-1)^{j-(x+1)}, \ 2(k-1)^{j-x}, \right.$$

$$\text{with } {}_{j=2x,\ x \in [1,ed/2]} \text{ and } {}_{j=2x-1,\ x \in [2,ed/2]}$$

$$\left. (k-1)^{j-x} \right\|_{j=ed+1,\ x=ed/2+1} \qquad (8)$$

Dicentric dendrimers

F - matrix

$p = 1$

$$2 * \Big\| k(k-1)^{j-2}\,,\ (k-1)^{j-2} \Big\|_{j\in[2,(ed+1)/2]\ \ j=(ed+1)/2+1} \qquad (9)$$

$p\in[2,(ed+1)/2-1]$

$$2(k-1)^{p-1} * \Big\| k(k-1)^{j-2}\,,\ (k-1)^{j-(x+1)} \Big\|_{j\in[2,(ed+1)/2-p+1]\ \ j=(ed+1)/2-p+2x,\ x\in[1,p]}$$

$$\Big\| (k-1)^{j-(x+1)} \Big\|_{j=(ed+1)/2-p+(2x+1),\ x\in[1,(p-1)]} \qquad (10)$$

$p = (ed+1)/2$

$$2(k-1)^{(ed-1)/2} * \Big\| (k-1)^{j-(x+1)} \Big\|_{j=2x,\ x\in[1,(ed+1)/2]}\ \Big\|_{j=2x+1,\ x\in[1,(ed-1)/2]} \qquad (11)$$

B - matrix

$p = 1$

$$2 * \Big\| k,\ k^2(k-1)^{j-2}\,,\ (2k-1)(k-1)^{j-2}\,,\ (k-1)^{j-2} \Big\|_{j\in[2,(ed-1)/2]\ \ j=(ed+1)/2\ \ j=(ed+1)/2+1} \qquad (12)$$

$p\in[2,(ed+1)/2-1]$

$$2(k-1)^{p-1} * \Big\| k,\ k^2(k-1)^{j-2}\,,\ (2k-1)(k-1)^{j-2},\ k(k-1)^{j-(x+1)},\ 2(k-1)^{j-(x+1)} \Big\|_{j\in[2,(ed+1)/2-p]\ \ j\in[(ed+1)/2-p+1]\ \ j=(ed+1)/2-p+2x,\ x\in[1,(p-1)]\ \ j=(ed+1)/2-p+(2x+1),\ x\in[1,(p-1)]}$$

$$\Big\| (k-1)^{j-(x+1)} \Big\|_{j=(ed+1)/2+p,\ x=p} \qquad (13)$$

$$p = (ed+1)/2 \qquad 2(k-1)^{(ed-1)/2} * \Big\| 1 \;,\; \underset{\substack{j=2x \\ x\in[1,(ed-1)/2]}}{k(k-1)^{j-(x+1)}} \;,$$

$$\underset{\substack{j=2x+1 \\ x\in[1,(ed-1)/2]}}{2(k-1)^{j-(x+1)}} \;,\; \underset{\substack{j=ed+1 \\ x=(ed+1)/2}}{(k-1)^{j-(x+1)}} \Big\| \qquad (14)$$

The parameter j follows the columns in the B matrix and since $\min(j_{max})=2$ (j_{max} being the maximal value of j), then $2 \leq j \leq j_{max}$ and the exponent will take only positive integer values (zero included). For the monocentric dendrimers $j_{max}=ed/2+p$ and for the dicentric ones, $j_{max}=(ed+1)/2+p$; ed reffers to the edge - diameter of graph.

It is also useful to consider the combinatorial diameter, cd [9] which counts the nodes in the largest path of graph. Thus, ed=2er+i and cd=2cr+2-i, with i=0 for even diameter and i=1 for odd diameter, respectively, the centroids (a vertex, in monocentric dendrimers or two vertices and an edge, in dicentric ones) being taken of when the radius is calculated.

3. General formula for binary dendrimers, A_xB_y

From the partition, p=1, in the F matrix, corresponding to the dendrimer core one can derive the general coefficients for the binary dendrimers, A_xB_y , keeping in mind that $jm=r_m+1+zi$:

$$x = 1 + \sum_{r1}^{r_m} k(k-1)^{r-1} + zi(k-1)^{r_m} \qquad (15)$$

$$y = (1 + zi) * k^{1-zi} * (k-1)^{r_m+zi} \qquad (16)$$

with z=e $\longrightarrow$ zi=i and z=c $\longrightarrow$ zi=1-i, where e and c denote the edge-

and combinatorial diameter, respectively. The parameter r counts the shells around the centroid and it is equivalent to the generation, G - counter [10, 11].

Table 1 lists the relations for x and y (for k up to 6) which can be useful in computing some large "paper" molecules.

Table 1. General relations for binary compounds, A_xB_y

k	x	y	General formula $A_xB_{(k-2)x+2}$
2	$1 + zi + 2r$ *)	2	A_xB_2
3	$3^{(1-zi)}2^{(r+2zi)}-2$	$3^{(1-zi)}2^{(r+2zi)}$	A_xB_{x+2}
4	$2^{(1-zi)}3^{(r+zi)}-1$	$2^{(2-zi)}3^{(r+zi)}$	A_xB_{2x+2}
5	$1/3\left[5^{(1-zi)}2^{(2r+3zi)}-2\right]$	$5^{(1-zi)}2^{(2r+3zi)}$	A_xB_{3x+2}
6	$1/2\left[3^{(1-zi)}5^{(r+zi)}-1\right]$	$2\cdot3^{(1-zi)}5^{(r+zi)}$	A_xB_{4x+2}

$z = e \rightarrow zi = i$; $ed = 2r + i$; $i = 0$ for d = even

$z = c \rightarrow zi = 1-i$; $cd = 2r + 2 - i$; $i = 1$ for d = odd

*) r - reffers here to the r_m - the maximal value of r in dendrimer

Table 2 exemplifies several homogeneous "paper" chemical dendrimers of general formula :

$$\left[AB_{(chk-(\pi+k))}\right]_x B_{(k-2)x+2} \qquad (17)$$

where chk denotes the chemical valence and π counts the π-bonds; B within the square brackets stands for the monovalent groups (e.g. hydrogen atoms) attached to the inner branching nodes. The total number of B groups obey the equality :

$$y = \text{total B} = (chk-(\pi+k))\cdot x + (k-2)\cdot x + 2 =$$
$$= (chk-\pi-2)\cdot x + 2 \qquad (18)$$

Table 2. "Paper" chemical homogeneous dendrimers, of general formula $AB_{(chk-(\pi+k))}x^{B}(k-2)x+2$: chk - chemical valence, π - number of π-bonds, cd - combinatorial diameter; maximum compactness for π=0, k = chk; linear for k = 2

k		A	B	π	cd	Formula		Compactness
k	chk					Graph theoret.	Chemical	(k)
2	2	-S-	H	0	7	A_xB_2	S_7H_2	max.
					10		$S_{10}H_2$	(k=2)
2	3	-N=	H	1	10	A_xB_2	$N_{10}H_2$	min.,linear unsaturated
2	4	-C≡	H	2	10	A_xB_2	$C_{10}H_2$	min.,linear unsaturated
2	3	-NH-	H	0	7	A_xB_2	N_7H_9	min.,linear
					10		$N_{10}H_{12}$	saturated
2	4	$-CH_2-$	H	0	7	A_xB_2	C_7H_{16}	min.,linear
					10		$C_{10}H_{22}$	saturated
3	3	-N<	H	0	7	A_xB_{x+2}	$N_{22}H_{24}$	max.
					10		$N_{62}H_{64}$	(k=3)
3	4	-CH<	H	0	7	A_xB_{x+2}	$C_{22}H_{46}$	mean
					10		$C_{62}H_{126}$	(k=3)
4	4	-C(-)-	H	0	7	A_xB_{2x+2}	$C_{53}H_{108}$	max.
					10		$C_{242}H_{486}$	(k=4)

In saturated molecules (π=0), for a given diameter, the maximal branching is reached with k=chk, whereas linear minimal branching is obtained when k=2. Thus, a normale-alkane represents a particular case of dendrimers, so that the general expressions (3 - 14) for the layer matrices F and B can be easily written, as in Appendix 1. For comparison, Appendices 2 and 3 give other useful recurrences for the layer matrices in n-alkanes (paths), as stated by us, elsewhere [20], on the ground of the straight-chain symmetry (partitions start from the periphery, p=1, and run to the center of graph) and the proper-

ties of Y - index (see below).

4. Y - indices in a compactness criterion

Based on the above described F and B matrices in dendrimers, one can easily derive the Y - indices (cf. eq. 1 and 2) :

Monocentric dendrimers

$$FY = \left[\sum_{j=2}^{ed/2+1} k(k-1)^{j-2} \cdot 10^{1-j}\right] + \sum_{p=2}^{ed/2} k(k-1)^{p-2} \cdot$$

$$\cdot \left[\sum_{j=2}^{ed/2-p+2} k(k-1)^{j-2} \cdot 10^{1-j} + \sum_{\substack{j=ed/2-p+(2x+1) \\ x\in[1,(p-1)]}} (k-1)^{j-(x+1)} \cdot 10^{1-j} + \right.$$

$$\left. + \sum_{\substack{j=ed/2-p+2x \\ x\in[2,p]}} (k-1)^{j-x} \cdot 10^{1-j}\right] + k(k-1)^{(ed/2-1)} \cdot$$

$$\cdot \left[\sum_{\substack{j=2x \\ x\in[1,ed/2]}} (k-1)^{j-(x+1)} \cdot 10^{1-j} + \sum_{\substack{j=2x-1 \\ x\in[2,ed/2+1]}} (k-1)^{j-x} \cdot 10^{1-j}\right] \quad (19)$$

$$BY = \left[k + \sum_{j=2}^{ed/2} k^2(k-1)^{j-2} \cdot 10^{1-j} + k(k-1)^{j-2} \cdot 10^{1-j}\right] +$$

$$+ \sum_{p=2}^{ed/2} k(k-1)^{p-2} \cdot \left[k + \sum_{j=2}^{ed/2-p+1} k^2(k-1)^{j-2} \cdot 10^{1-j} + \right.$$

$$+ (2k-1)(k-1)^{j-2} * 10^{1-j} + \sum_{\substack{j=ed/2-p+2 \\ x\epsilon[1,(p-1)]}}^{j=ed/2-p+(2x+1)} k(k-1)^{j-(x+1)} * 10^{1-j} +$$

$$+ \sum_{\substack{j=ed/2-p+2x \\ x\epsilon[2,(p-1)]}} 2(k-1)^{j-x} * 10^{1-j} + (k-1)^{j-x} * 10^{1-j} \Bigg]_{\substack{j=ed/2+p \\ x=p}} +$$

$$+ k(k-1)^{ed/2-1} * \left[1 + \sum_{\substack{j=2x \\ x\epsilon[2,ed/2]}} k(k-1)^{j-(x+1)} * 10^{1-j} + \right.$$

$$\left. + \sum_{\substack{j=2x+1 \\ x\epsilon[2,ed/2]}} 2(k-1)^{j-x} * 10^{1-j} + (k-1)^{j-x} * 10^{1-j} \right]_{\substack{j=ed+1 \\ x=ed/2+1}} \qquad (20)$$

Dicentric dendrimers

$$FY = 2 * \left[\sum_{j=2}^{(ed+1)/2} k(k-1)^{j-2} * 10^{1-j} + (k-1)^{j-2} * 10^{1-j} \right]_{j=(ed+1)/2+1} +$$

$$+ \sum_{p=2}^{(ed-1)/2} 2(k-1)^{p-1} * \left[\sum_{j=2}^{(ed+1)/2-p+1} k(k-1)^{j-2} * 10^{1-j} + \right.$$

$$\left. + \sum_{\substack{j=(ed+1)/2-p+2x \\ x\epsilon[1,p]}} (k-1)^{j-(x+1)} * 10^{1-j} + \sum_{\substack{j=(ed+1)/2-p+(2x+1) \\ x\epsilon[1,(p-1)]}} (k-1)^{j-(x+1)} * 10^{1-j} \right] +$$

$$+ 2(k-1)^{(ed-1)/2} * \left[\sum_{\substack{j=2x \\ x\epsilon[1,(ed+1)/2]}} (k-1)^{j-(x+1)} * 10^{1-j} + \right.$$

$$+ \sum_{\substack{j=2x+1 \\ x\in[1,(ed+1)/2]}} (k-1)^{j-(x+1)} * 10^{1-j} \Bigg] \qquad (21)$$

$$BY = 2 * \Bigg[k + \sum_{j=2}^{(ed-1)/2} k^2(k-1)^{j-2} * 10^{1-j} + (2k-1)(k-1)^{j-2} * \sum_{j=(ed+1)/2} * 10^{1-j} + (k-1)^{j-2} * 10^{1-j} \Bigg] + \sum_{p=2}^{(ed-1)/2} 2(k-1)^{p-1} * \Bigg[k +$$

$$+ \sum_{j=2}^{(ed+1)/2-p} k^2(k-1)^{j-2} * 10^{1-j} + (2k-1)(k-1)^{j-2} * 10^{1-j} + \sum_{j=(ed+1)/2-p+1} +$$

$$+ \sum_{\substack{j=(ed+1)/2-p+2x \\ x\in[1,(p-1)]}} k(k-1)^{j-(x+1)} * 10^{1-j} + \sum_{\substack{j=(ed+1)/2-p+(2x+1) \\ x\in[1,(p-1)]}} 2(k-1)^{j-(x+1)} *$$

$$* 10^{1-j} + (k-1)^{j-(x+1)} * 10^{1-j} \Bigg] + 2(k-1)^{(ed-1)/2} * \Bigg[1 + \sum_{\substack{j=(ed+1)/2+p \\ x=p}}$$

$$+ \sum_{\substack{j=2x \\ x\in[1,(ed-1)/2]}} k(k-1)^{j-(x+1)} * 10^{1-j} + \sum_{\substack{j=2x+1 \\ x\in[1,(ed-1)/2]}} 2(k-1)^{j-(x+1)} * 10^{1-j} +$$

$$+ (k-1)^{j-(x+1)} * 10^{1-j} \Bigg] \qquad (22)$$
$$\substack{j=ed+1 \\ x=(ed+1)/2}$$

For n-alkanes, recursive relations for computing the Y -

- indices, are included in Appendices 1 to 3.

As stated in section 3, for a given diameter the maximal branched saturated molecule is the dendrimer with k=chk. It is now conceivable to consider the maximal branched dendrimer as the most compact (see also [9]) among all the molecules of a given diameter. In other words, such a dendrimer filles better a spherical space of a given topological diameter.

Thus, a compactness criterion can be constructed by dividing the Y - values of a given tree by the Y - values of the dendrimer of the same diameter. Since the maximal chk in alkanes is four, we propose k=chk=4 for the reference dendrimers.

Table 3 lists the results for the octane isomers. The ordering is given by the compactity parameter, YC, which ranges in the (0,1) interval. The Y - values for the corresponding dendrimers are given as footnotes.

Table 3. The compactness criterion, YC, in octane isomers

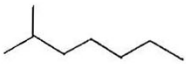
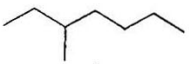
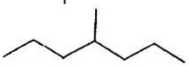
d / e c	G	FY(G)	BY(G)	YC
7 8		1.53086	16.8395	0.07863
6 7		1.55086	17.0595	0.12126
6 7		1.55286	17.0815	0.12142
6 7		1.55304	17.0835	0.12143

Table 3. (continued)

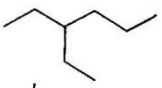
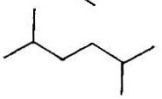
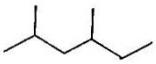
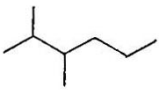
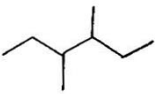
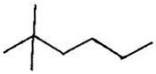
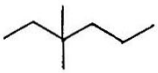
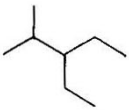
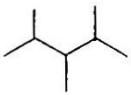
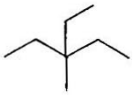
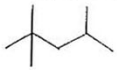
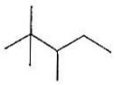
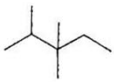
d / e c	G	FY(G)	BY(G)	YC
5 6		1.55502	17.1054	0.25454
5 6		1.57088	17.2797	0.25713
5 6		1.57304	17.3034	0.25749
5 6		1.57484	17.3232	0.25778
5 6		1.57682	17.3450	0.258105
5 6		1.59036	17.4995	0.26040
5 6		1.59482	17.5430	0.26105
4 5		1.57700	17.3470	0.40615
4 5		1.5968	17.5648	0.41125
4 5		1.5986	17.5846	0.41171

Table 3. (continued)

d		G	FY(G)	BY(G)	YC
e	c				
4	5		1.6112	17.7232	0.41496
4	5		1.6166	17.7826	0.41635
4	5		1.6184	17.8024	0.41681
3	4		1.658	18.2380	1.00000

Dendrimers, (cd, FY, BY) :

1, 0.20000, 2.20000 ; 2, 0.92000, 10.12000 ; 3, 1.65800, 18.23800 ;
4, 3.88280, 42.71080 ; 5, 6.10922, 67.20142 ;
6, 12.78945, 140.68397 ; 7, 19.46983, 214.16813 .

One can see that only one inversion appeared by crossing from Y - indices to YC. The less compact was the linear n-octane whereas the most compact was 2,2,3,3-tetramethyl-butane (the dicentric dendrimer of cd=4). In this view, in our group, other compactness criteria are in progress.

Concluding, the layer matrix description of homogeneous dendrimers enables one to "see" the whole structure by the relative partition of each of its nodes. From the partition corresponding to the dendrimer core, general relations for computing

binary dendrimers, A_xB_y, where derived. Such relations would be useful in a group analysis of heterogeneous dendrimers [21]. The proposed YC-compactness criterion orders the alkane isomers vs. the corresponding dendrimers of the same diameter.

Computer programs

Y - indices in octane isomers were calculated with the GRAPH -08 program (in FORTRAN - 77). The input of molecular graphs is in a dictionary form.

YC - criterion was tested with the program DENDR - 1 (in PASCAL, on an IBM-PC compatible microcomputer), which is based on eq. (3-14). Input data file includes the combinatorial diameter, vertex degree (chemical and graph theoretical ones) and the number of π-bonds incident in a node.

Exit file lists the matrices F and B for the dendrimers (in a linear form) and local and global Y - invariants, as well as the compactness parameter YC. This program also computes the general formula for a binary dendrimer, A_xB_y (cf. eq. 15 - 18) as shown in Appendix 4.

The source program DENDR - 1 is listed in Appendix 5.

REFERENCES

1. I. E. Kacsó, A. Baras-Diudea, M. V. Diudea, Rev. Roumaine Chim. (in press)
2. L. B. Kier and L. H. Hall, "Molecular connectivity in chemistry and drug research", (Acad. Press New York)1976
3. I. Gutman, M. Randić, Chem. Phys. Lett. 47 (1977) 15-19
4. D. Bonchev, J. V. Knop and N. Trinajstić, Math. Chem. 6 (1979) 21-47
5. A. T. Balaban, Chem. Phys. Lett. 89 (1982) 399-404
6. A. T. Balaban, Pure & Appl. Chem. 55 (1983) 199-206
7. S. H. Bertz, Discr. Appl. Math. 19 (1988) 65-83
8. D. H. Rouvray, Discr. Appl. Math. 19 (1988) 317-338
9. R. E. Davies, P. J. Freyd, J. Chem. Educ. 66 (1989) 278--281
10. C. J. Hawker and J. M. Fréchet, J. Am. Chem. Soc., 112 (1990) 7638-7647
11. D. A. Tomalia, A. M. Naylor, W. A. Goddard, Angew. Chem. 102(2) (1990) 119-238
12. M. V. Diudea, O. M. Minailiuc, A. T. Balaban, J. Comput. Chem. 12 (1991) 527-535
13. M. Randić, J. Am. Chem. Soc. 97 (1975) 6609
14. A. T. Balaban, I. Motoc, D. Bonchev, O. Mekenyan, Top. Curr. Chem. 114 (1983) 21
15. O. Mekenyan, D. Bonchev, A. T. Balaban, J. Math. Chem. 2 (1988) 347-375
16. A. B. Padies, H. K. Hall, D. A. Tomalia, J. R. McConnell, J. Org. Chem. 52 (1989) 5305
17. N. M. Bikales, Polym. J. 19 (1989) 11

18. V. A. Skorobogatov, A. A. Dorynin, MATCH 23 (1988) 105--151

19. F. Harary, "Graph Theory", Addison Wesley, Reading (Mass), 1971

20. M. V. Diudea, L. Bal, Studia Univ. Babes-Bolyai (in press)

21. M. V. Diudea et al. (manuscript in preparation)

Appendix 1. Layer matrices F and B, and the corresponding Y - indices for n-alkanes (paths) as dendrimers

Monocentric paths (k=2)

F - matrix

$p = 1$ $\quad 1 * \| 2 * {}_1{}^{j-2} \|_{j\in[2,jm]}$ $\quad jm = ed/2 + p$

$p\in[2,ed/2]$ $\quad 2 * \| 2 * {}_1{}^{j-2}{}_{j\in[2,jm-(2p-2)]}\ ,\ {}_1{}^{j-(x+1)}{}_{j=jm-[2p-(2x+1)]\ x\in[1,(p-1)]}\ ,\ {}_1{}^{j-x} \|_{j=jm-(2p-2x)\ x\in[2,p]}$

$p = ed/2+1$ $\quad 2 * \| {}_1{}^{j-(x+1)}{}_{j=jm-[2p-(2x+1)]\ x\in[1,(p-1)]}\ ,\ {}_1{}^{j-x} \|_{j=jm-(2p-2x)\ x\in[2,p]}$

B - matrix

$p = 1$ $\quad 1 * \| 2\ ,\ 4 * {}_1{}^{j-2}{}_{j\in[2,jm-1]}\ ,\ 2 * {}_1{}^{jm-2} \|$ $\quad jm = ed/2+p$

$p\in[2,ed/2]$ $\quad 2 * \| 2\ ,\ 4 * {}_1{}^{j-2}{}_{j\in[2,jm-(2p-1)]}\ ,\ 3 * {}_1{}^{j-2}{}_{j=jm-(2p-2)}\ ,\ 2 * {}_1{}^{j-(x+1)}{}_{j=jm-[2p-(2x+1)]\ x\in[1,(p-1)]}\ ,\ 2 * {}_1{}^{j-x}{}_{j=jm-(2p-2x)\ x\in[2,(p-1)]}\ ,\ {}_1{}^{jm-x} \|_{x=p}$

$p = ed/2+1$ $\quad 2 * \| 1\ ,\ 2 * {}_1{}^{j-(x+1)}{}_{j=jm-[2p-(2x+1)]\ x\in[1,(p-1)]}\ ,\ 2 * {}_1{}^{j-x}{}_{j=jm-(2p-2x)\ x\in[2,(p-1)]}\ ,\ {}_1{}^{jm-x} \|_{x=p}$

Appendix 1. (continued)

Dicentric paths (k=2)

F - matrix

$p = 1$ $\quad 2 * \left\| 2 * {}_1{}^{j-2}{}_{j\in[2,(jm-1)]} \;,\; {}_1{}^{jm-2} \right\|$ $\quad jm=(ed+1)/2+p$

$p\in[2,(ed+1)/2-1]$ $\quad 2 * \left\| 2 * {}_1{}^{j-2}{}_{j\in[2,jm-(2p-1)]} \;,\; {}_1{}^{j-(x+1)}{}_{\substack{j=jm-(2p-2x)\\ x\in[1,p]}} \;,\; {}_1{}^{j-(x+1)}{}_{\substack{j=jm-[2p-(2x+1)]\\ x\in[1,(p-1)]}} \right\|$

$p = (ed+1)/2$ $\quad 2 * \left\| {}_1{}^{j-(x+1)}{}_{\substack{j=jm-(2p-2x)\\ x\in[1,(p-1)]}} \;,\; {}_1{}^{j-(x+1)}{}_{\substack{j=jm-[2p-(2x+1)]\\ x\in[1,(p-1)]}} \right\|$

B - matrix

$p = 1$ $\quad 2 * \left\| 2 \;,\; 4 * {}_1{}^{j-2}{}_{j\in[2,jm-2]} \;,\; 3 * {}_1{}^{j-2}{}_{j=jm-1} \;,\; {}_1{}^{jm-2} \right\|$

$p\in[2,(ed+1)/2-1]$ $\quad 2 * \left\| 2 \;,\; 4 * {}_1{}^{j-2}{}_{j\in[2,jm-2p]} \;,\; 3 * {}_1{}^{j-2}{}_{j=jm-(2p-1)} \;,\; 2 * {}_1{}^{j-(x+1)}{}_{\substack{j=jm-(2p-2x)\\ x\in[1,(p-1)]}} \;,\; 2 * {}_1{}^{j-(x+1)}{}_{\substack{j=jm-[2p-(2x+1)]\\ x\in[1,(p-1)]}} \;,\; {}_1{}^{jm-(x+1)}{}_{x=p} \right\|$

$p = (ed+1)/2$ $\quad 2 * \left\| 1 \;,\; 2 * {}_1{}^{j-(x+1)}{}_{\substack{j=jm-(2p-2x)\\ x\in[1,(p-1)]}} \;,\; 2 * {}_1{}^{j-(x+1)}{}_{\substack{j=jm-[2p-(2x+1)]\\ x\in[1,(p-1)]}} \;,\; {}_1{}^{jm-(x+1)}{}_{x=p} \right\|$

Appendix 1. (continued)

Monocentric paths

$$FY = 2\sum_{j=2}^{ed/2+1} 10^{1-j} + 2\sum_{p=2}^{ed/2}\left[2\sum_{j=2}^{ed/2-p+2} 10^{1-j} + \sum_{j=ed/2-p+3}^{ed/2+p} 10^{1-j}\right] +$$

$$+ 2\sum_{j=2}^{ed+1} 10^{1-j}$$

$$BY = \left[2 + 4\sum_{j=2}^{ed/2} 10^{1-j} + 2*10^{1-j}_{j=ed/2+1}\right] + 2\sum_{p=2}^{ed/2}\left[2 + 4\sum_{j=2}^{ed/2-p+1} 10^{1-j} +\right.$$

$$\left. + 3*10^{1-j}_{j=ed/2-p+2} + 2\sum_{\substack{j=ed/2-p+(2x+1)\\ x\in[1,(p-1)]}} 10^{1-j} + 2\sum_{\substack{j=ed/2-p+2x\\ x\in[2,(p-1)]}} 10^{1-j} + 10^{1-j}_{j=ed/2+p}\right] +$$

$$+ 2 * \left[1 + 2\sum_{j=2}^{ed} 10^{1-j} + 10^{1-j}_{j=ed+1}\right]$$

Dicentric paths

$$FY = 2 * \left[2\sum_{j=2}^{(ed+1)/2} 10^{1-j} + 10^{1-j}_{j=(ed+1)/2+1}\right] + 2\sum_{p=2}^{(ed-1)/2}\left[\right.$$

$$\left.\left[2\sum_{j=2}^{(ed+1)/2-p+1} 10^{1-j} + \sum_{j=(ed+1)/2-p+2}^{(ed+1)/2+p} 10^{1-j}\right] + 2\sum_{j=2}^{ed+1} 10^{1-j}\right.$$

$$BY = 2 * \left[2 + 4\sum_{j=2}^{(ed-1)/2} 10^{1-j} + 3*10^{1-j}_{j=(ed+1)/2} + 10^{1-j}_{j=(ed+1)/2+1}\right] +$$

$$+ 2\sum_{p=2}^{(ed-1)/2}\left[2 + 4\sum_{j=2}^{(ed+1)/2-p} 10^{1-j} + 3*10^{1-j}_{j=(ed+1)/2-p+1} +\right.$$

Appendix 1. (continued)

$$+ 2\sum_{\substack{j=(ed+1)/2-p+2x \\ x\in[1,(p-1)]}} 10^{1-j} + 2\sum_{\substack{j=(ed+1)/2-p+(2x+1) \\ x\in[1,(p-1)]}} 10^{1-j} + 10^{1-j}_{\,j=(ed+1)/2+p} \Bigg] +$$

$$+ 2 * \left[1 + 2\sum_{j=2}^{ed} 10^{1-j} + 10^{1-j}_{\,j=ed+1} \right]$$

Appendix 2. Layer matrices F and B, and the corresponding Y - indices for n-alkanes (paths) starting from the periphery (p=1)

$$F(P_{2v}) = \| s * (\underbrace{2\ldots2}_{p-1} , \underbrace{1\ldots1}_{2(v-p)+1} , \underbrace{0\ldots0}_{p-1}) \| \qquad p\in[1,v] \ ; \ s=2$$

$$F(P_{2v+1}) = \| s * (\underbrace{2\ldots2}_{p-1} , \underbrace{1\ldots1}_{2(v-p)+2} , \underbrace{0\ldots0}_{p-1}) \| \qquad p\in[1,v+1]; \ s=2 \text{ for } p\in[1,v] \ ; \ s=1 \text{ for } p=v+1$$

$$B(P_{2v}) = \| s * (1 , \underbrace{2\ldots2}_{2(v-1)} , 1); s * (2 , \underbrace{4\ldots4}_{p-2} , 3 , \underbrace{2\ldots2}_{2(v-p)} ,$$

$$1 , \underbrace{0\ldots0}_{p-1}) \| \qquad p\in[2,v] \ ; \ s=2$$

$$B(P_{2v+1}) = \| s * (1 , \underbrace{2\ldots2}_{2v-1} , 1); s * (2 , \underbrace{4\ldots4}_{p-2} , 3 ,$$

$$\underbrace{2\ldots2}_{2(v-p)+1} , 1 , \underbrace{0\ldots0}_{p-1}); s * (2 , \underbrace{4\ldots4}_{p-2} , 2 , \underbrace{0\ldots0}_{p-1}) \|$$

$$p\in[2,v]; \ s=2 \qquad p=v+1; \ s=1$$

Appendix 2. (continued)

$$FY(P_{2v}) = \sum_{s=1}^{2} \sum_{p=1}^{v} \left(2 \sum_{j=1}^{p-1} 10^{-j} + \sum_{j=p}^{2v-p} 10^{-j} \right) * w_{ps}$$

$$FY(P_{2v+1}) = \sum_{s=1}^{2} \sum_{p=1}^{v} \left(2 \sum_{j=1}^{p-1} 10^{-j} + \sum_{j=p}^{2v-p+1} 10^{-j} \right) * w_{ps} +$$

$$+ \left(2 \sum_{j=1}^{v} 10^{-j} \right) * w_{v+1}$$

$$BY(P_{2v}) = \sum_{s=1}^{2} \left(1 + 2 \sum_{j=2}^{2v-1} 10^{1-j} + 10^{1-2v} \right) * w_{1s} + \sum_{s=1}^{2} \sum_{p=2}^{v} \left(2 + \right.$$

$$\left. + 4 \sum_{j=2}^{p-1} 10^{1-j} + 3 * 10^{1-p} + 2 \sum_{j=p+1}^{2v-p} 10^{1-j} + 10^{p-2v} \right) * w_{ps}$$

$$BY(P_{2v+1}) = \sum_{s=1}^{2} \left(1 + 2 \sum_{j=2}^{2v} 10^{1-j} + 10^{-2v} \right) * w_{1s} + \sum_{s=1}^{2} \sum_{p=2}^{v} \left(2 + \right.$$

$$\left. + 4 \sum_{j=2}^{p-1} 10^{1-j} + 3 * 10^{1-p} + 2 \sum_{j=p+1}^{2v-p+1} 10^{1-j} + 10^{p-2v-1} \right) *$$

$$* w_{ps} + \left(2 + 4 \sum_{j=2}^{v} 10^{1-j} + 2 * 10^{-v} \right) * w_{v+1}$$

Appendix 3. Layer matrices F and B, and the corresponding Y - indices for n-alkanes (paths), when matrices are written as sums on columns :

$$\sum_{i} F(P_v) = \| 2 * (v - j) \| ; \qquad j \in [1, (v-1)]$$

$$\sum_{i} B(P_v) = \| 2 * (v - 1) ; 4 * (v - j) + 2 \| ; \qquad j \in [2, v]$$

Appendix 3. (continued)

$$FY(P_v) = 2\sum_{j=1}^{v-1} (v - j) * 10^{-j}$$

$$BY(P_v) = 2 * (v - 1) + \sum_{j=2}^{v} \left[4 * (v - j) + 2 \right] * 10^{1-j}$$

Appendix 4. Input and output files in DENDR_I

```
     INPUT DATA

 Binary dendrimer AxBy
   A=branching nodes
   B=monovalent nodes attached to A nodes

   A= C
   B= H
 pi bonds( around A)= 0
 Dendrimer diameter
 Selected : 2 from 1- edges diameter / 2- combinatorial diameter
 cd= 10
 Chemical degree, chk= 4
 Graph theoretical degree, k= 4

     OUTPUT DATA

 Graph theoretical formula : C242 H486
 Chemical formula : C242 H486

   Matrix B (Dicentric):
 B[1]=6.24371   2*{ 4,16,48,144,189,81 }
 B[2]=6.15550   6*{ 4,16,48,63,108,162,81 }
 B[3]=5.85265   18*{ 4,16,21,36,54,108,162,81 }
 B[4]=4.84227   54*{ 4,7,12,18,36,54,108,162,81 }
 B[5]=1.47423   162*{ 1,4,6,12,18,36,54,108,162,81 }
 BY=  655.07513

   Matrix F (Dicentric):
 F[1]=0.56761   2*{ 4,12,36,108,81 }
 F[2]=0.55959   6*{ 4,12,36,27,81,81 }
 F[3]=0.53206   18*{ 4,12,9,27,27,81,81 }
 F[4]=0.44021   54*{ 4,3,9,9,27,27,81,81 }
 F[5]=0.13402   162*{ 1,3,3,9,9,27,27,81,81 }
 FY=  59.55228


     INPUT DATA

 Graph diameter of testing molecule
 Selected : 2 from 1- edges diameter / 2- combinatorial diameter
 cd= 5
 Name of molecule :  TEST
 FY (1)  or BY (2) -value= 2
 BY( for TEST)=
 1.35140

     OUTPUT DATA

 BY(dendrimer)=  42.70360
   YC=   0.03165
```

```
    INPUT DATA

Binary dendrimer AxBy
  A=branching nodes
  B=monovalent nodes attached to A nodes

  A= N
  B= H
pi bonds( around A)= 0
Dendrimer diameter
Selected : 2 from 1- edges diameter / 2- combinatorial diameter
cd= 7
Chemical degree, chk= 3
Graph theoretical degree, k= 2

    OUTPUT DATA

Graph theoretical formula : (NH)7 H2
Chemical formula : N7 H9

  Matrix B (Monocentric) :
B[1]=2.44200  1*{ 2,4,4,2 }
B[2]=2.43210  2*{ 2,4,3,2,1 }
B[3]=2.32221  2*{ 2,3,2,2,2,1 }
B[4]=1.22222  2*{ 1,2,2,2,2,2,1 }
BY=   14.39506

  Matrix F (Monocentric) :
F[1]=0.22200  1*{ 2,2,2 }
F[2]=0.22110  2*{ 2,2,1,1 }
F[3]=0.21111  2*{ 2,1,1,1,1 }
F[4]=0.11111  2*{ 1,1,1,1,1,1 }
FY=    1.30864
```

```
    INPUT DATA

Binary dendrimer AxBy
  A=branching nodes
  B=monovalent nodes attached to A nodes

  A= N
  B= H
pi bonds( around A)= 1
Dendrimer diameter
Selected : 2 from 1- edges diameter / 2- combinatorial diameter
cd= 10
Chemical degree, chk= 3
Graph theoretical degree, k= 2

    OUTPUT DATA

Graph theoretical formula : N10 H2
Chemical formula : N10 H2

  Matrix B (Dicentric):
B[1]=2.44431   2*{ 2,4,4,4,3,1 }
B[2]=2.44322   2*{ 2,4,4,3,2,2,1 }
B[3]=2.43222   2*{ 2,4,3,2,2,2,2,1 }
B[4]=2.32222   2*{ 2,3,2,2,2,2,2,2,1 }
B[5]=1.22222   2*{ 1,2,2,2,2,2,2,2,2,2,1 }
BY=    21.72840

  Matrix F (Dicentric):
F[1]=0.22221   2*{ 2,2,2,2,1 }
F[2]=0.22211   2*{ 2,2,2,1,1,1 }
F[3]=0.22111   2*{ 2,2,1,1,1,1,1 }
F[4]=0.21111   2*{ 2,1,1,1,1,1,1,1,1 }
F[5]=0.11111   2*{ 1,1,1,1,1,1,1,1,1,1 }
FY=    1.97531
```

```
    INPUT DATA

Binary dendrimer AxBy
  A=branching nodes
  B=monovalent nodes attached to A nodes

  A= COH
  B= NH2
pi bonds( around A)= 0
Dendrimer diameter
Selected : 2 from 1- edges diameter / 2- combinatorial diameter
cd= 5
Chemical degree, chk= 3
Graph theoretical degree, k= 3

    OUTPUT DATA

Graph theoretical formula : [COH]10 [NH2]12
Chemical formula : [COH]10 [NH2]12

  Matrix B (Monocentric) :
B[1]=3.96000  1*{ 3,9,6 }
B[2]=3.56400  3*{ 3,5,6,4 }
B[3]=1.34620  6*{ 1,3,4,6,2 }
BY=   22.72920

  Matrix F (Monocentric) :
F[1]=0.36000  1*{ 3,6 }
F[2]=0.32400  3*{ 3,2,4 }
F[3]=0.12240  6*{ 1,2,2,4 }
FY=    2.06640
```

Appendix 5: Source program DENDR_1

```
Program DENDR_1;


Uses  printer,crt;

Const
      imax=40;

Type
    BstocB=array[1..Jmax,0..Jmax] of integer;
    Str255=string[255];

Var
   zi,rm,chk,pib,j,d,k,rr,r:byte;
   elemB,ct,x,xx,yi,yyi:integer;
   Belem:BStocB;
   yc,fy,by,fyt,byt,ri,y,yy:real;
   BufB:Str255;
   ms,ch:char;
   A,B:string[10];


Function Power(j:byte):integer;

 Var
    rk:real;

 Begin
   if (j-2)>=0 then
     begin
        rk:=ln(k-1);
        power:=Round(exp((j-2)*rk));
     end
    else
     power:=0;
 End;{Power}


Function Powerx(j,x:byte):integer;

 Var
    rk:real;

 Begin
   rk:=ln(k-1);
   powerx:=Round(exp((j-x)*rk));
 End; {Powerx}


Function Powerx1(j,x:byte):integer;

 Var
    rk:real;

 Begin
```

```
  rk:=ln(k-1);
  powerx1:=round(exp((j-(x+1))*rk));
End; {Powerx1}


Function CalcOp(i:byte):real;

Begin
  CalcOp:=Belem[r,i]*exp((1-i)*ln(10));
End; {CalcOp}


Procedure Prima(Var Belem:BStocB);

Var
   elemB:integer;
   jm,j:byte;

Begin
  jm:=Round((d/2)+1);
  Belem[1,0]:=1;
  Belem[1,1]:=k;
  for j:=2 to Trunc(d/2) do
      begin
        elemB:=sqr(k);
        Belem[r,j]:=elemB*power(j);
      end;
  Belem[r,jm]:=k*power(jm);
End; {Prima}


Procedure PrimaDi(Var Belem:BstocB);

Var
   elemB:integer;
   jm,j:byte;

Begin
  jm:=Round(((d+1)/2)+1);
  Belem[1,0]:=2;
  Belem[1,1]:=k;
  for j:=2 to (jm-2) do
      begin
      elemB:=sqr(k);
      Belem[r,j]:=elemB*power(j);
      end;
  j:=jm-(2*r-1);
  Belem[r,j]:=(2*k-1)*Power(j);
  Belem[r,jm]:=power(jm);
End; {PrimaDi}


Procedure Corp(Var Belem:BStocB);

Var
   jm,x,j:byte;
   elemB:integer;
```

```
Begin
   jm:=Round(d/2+r);
   Belem[r,0]:=Round(k*exp((r-2)*ln(k-1)));
   Belem[r,1]:=k;
   if (jm-(2*r-1))>=2 then
      begin
        for j:=2 to (jm-(2*r-1)) do
            Belem[r,j]:=Round(sqr(k)*Power(j));
      end;
   j:=jm-(2*r-2);
   elemB:=2*k-1;
   Belem[r,j]:=Round(elemB*Power(j));
   for x:=1 to (r-1) do
       begin
         j:=jm-(2*r-(2*x+1));
         Belem[r,j]:=k*Powerx1(j,x);
       end; {for}
   for x:=2 to (r-1) do
       begin
         j:=jm-(2*r-2*x);
         if Belem[r,j]=0 then
            Belem[r,j]:=2*Powerx(j,x);
       end; {for}
   Belem[r,jm]:=Powerx(jm,r);
End; {Corp}


Procedure CorpDi(Var Belem:BStocB);

 Var
    jm,x,j:byte;
    elemB:integer;

 Begin
   jm:=Round((d+1)/2+r);
   Belem[r,0]:=Round(2*exp((r-1)*ln(k-1)));
   Belem[r,1]:=k;
   if (jm-2*r)>=2 then
      begin
        for j:=2 to (jm-2*r) do
            Belem[r,j]:=Round(sqr(k)*Power(j));
      end;
   j:=jm-(2*r-1);
   elemB:=2*k-1;
   Belem[r,j]:=Round(elemB*Power(j));
   for x:=1 to (r-1) do
       begin
         j:=jm-(2*r-2*x);
         Belem[r,j]:=k*Powerx1(j,x);
       end; {for}
   for x:=1 to (r-1) do
       begin
         j:=jm-(2*r-(2*x+1));
         if Belem[r,j]=0 then
            Belem[r,j]:=2*Powerx1(j,x);
       end; {for}
   Belem[r,jm]:=Powerx1(jm,r);
```

```
End; {CorpDi}


Procedure EndB(Var Belem:BStocB);

 Var
    j,jm,x:byte;
    elemB:integer;

 Begin
   jm:=Round(d/2+r);
   Belem[r,0]:=Round(k*exp((r-2)*ln(k-1)));
   Belem[r,1]:=1;
   for x:=1 to (r-1) do
       begin
         j:=jm-(2*r-(2*x+1));
         Belem[r,j]:=k*powerx1(j,x);
       end; {for}
   for x:=2 to (r-1) do
       begin
         j:=jm-(2*r-2*x);
         Belem[r,j]:=2*powerx(j,x);
       end; {for}
   Belem[r,jm]:=powerx1(jm,r);
 End;  {endB}


Procedure EndBDi(Var Belem:BStocB);

 Var
    j,jm,x:byte;
    elemB:integer;

 Begin
   jm:=Round((d+1)/2+r);
   Belem[r,0]:=Round(2*exp((r-1)*ln(k-1)));
   Belem[r,1]:=1;
   for x:=1 to (r-1) do
       begin
         j:=jm-(2*r-2*x);
         Belem[r,j]:=k*powerx1(j,x);
       end; {for}
   for x:=1 to (r-1) do
       begin
         j:=jm-(2*r-(2*x+1));
         Belem[r,j]:=2*powerx1(j,x);
       end; {for}
   Belem[r,jm]:=powerx1(jm,r);
 End;  {endBDi}


Procedure PrimaF(Var Belem:BStocB);

 Var
    elemB:integer;
    jm,j:byte;
```

```
Begin
   jm:=Round((d/2)+1);
   Belem[1,0]:=1;
   for j:=2 to jm do
       Belem[r,(j-1)]:=k*power(j);
End; {primaF}


Procedure PrimaFDi(Var Belem:BStocB);

 Var
    elemB:integer;
    jm,j:byte;

 Begin
   jm:=Round((d+1)/2+1);
   Belem[1,0]:=2;
   for j:=2 to (jm-1) do
      Belem[r,(j-1)]:=k*power(j);
   Belem[r,(jm-1)]:=power(jm);
 End; {primaFDi}


Procedure CorpF(Var Belem:BStocB);

 Var
    jm,x,j:byte;
    elemB:integer;

 Begin
   jm:=Round(d/2+r);
   Belem[r,0]:=Round(k*exp((r-2)*ln(k-1)));
   for j:=2 to (jm-(2*r-2)) do
         Belem[r,(j-1)]:=Round(k*Power(j));
   for x:=1 to (r-1) do
       begin
         j:=jm-(2*r-(2*x+1));
         Belem[r,(j-1)]:=powerx1(j,x);
       end; {for}
   for x:=2 to r do
       begin
         j:=jm-(2*r-2*x);
         if Belem[r,(j-1)]=0 then
            Belem[r,(j-1)]:=powerx(j,x);
       end; {for}
 End; {corpF}


Procedure CorpFDi(Var Belem:BStocB);

 Var
    jm,x,j:byte;
    elemb:integer;

 Begin
   jm:=Round((d+1)/2+r);
   Belem[r,0]:=Round(2*exp((r-1)*ln(k-1)));
```

```
   for j:=2 to (jm-(2*r-1)) do
       Belem[r,(j-1)]:=Round(k*Power(j));
   for x:=1 to (r) do
       begin
         j:=jm-(2*r-2*x);
         Belem[r,(j-1)]:=powerx1(j,x);
       end; {for}
   for x:=1 to (r-1) do
       begin
         j:=jm-(2*r-(2*x+1));
         if Belem[r,(j-1)]=0 then
            Belem[r,(j-1)]:=powerx1(j,x);
       end; {for}
 End; {corpFDi}


Procedure EndF(Var Belem:BStocB);

 Var
    j,jm,x:byte;
    elemB:integer;

 Begin
   jm:=Round(d/2+r);
   Belem[r,0]:=Round(k*exp((r-2)*ln(k-1)));
   for x:=1 to (r-1) do
       begin
         j:=jm-(2*r-(2*x+1));
         Belem[r,(j-1)]:=powerx1(j,x);
       end; {for}
   for x:=2 to r do
       begin
         j:=jm-(2*r-2*x);
         Belem[r,(j-1)]:=powerx(j,x);
       end; {for}
 End; {EndF}


Procedure EndFDi(Var Belem:BStocB);

 Var
    j,jm,x:byte;
    elemB:integer;

 Begin
   jm:=Round((d+1)/2+r);
   Belem[r,0]:=Round(2*exp((r-1)*ln(k-1)));
   for x:=1 to r do
       begin
         j:=jm-(2*r-2*x);
         Belem[r,(j-1)]:=powerx1(j,x);
       end; {for}
   for x:=1 to (r-1) do
       begin
         j:=jm-(2*r-(2*x+1));
         Belem[r,(j-1)]:=powerx1(j,x);
       end; {for}
```

```
End; {endFDi}

Procedure Scrie(ms:char;Belem:BstocB;rm:integer;Var y:real;
                Var BufB:str255);

Var
   strB:string[5];
   i:byte;
   yy:real;

Begin
  y:=0;
  for r:=1 to rm do
      begin
        yy:=0;
        Str (Belem[r,0],strB);
        BufB:='';
        BufB:=bufB+strB+'*{ ';
        i:=1;
        repeat
         begin
           if ms='F' then
             begin
               yy:=yy+CalcOp(i)/10;
             end
           else
             yy:=yy+CalcOp(i);
           Str(Belem[r,i],strB);
           bufB:=BufB+strB+',';
           i:=i+1;
         end;
        until (Belem[r,i]=0);
        Delete(BufB,(Length(BufB)),1);
        BufB:=BufB+' }';
        if ct<>2 then
          begin
            writeln(' ',ms,'[',r,']=',yy:6:5,'  ',BufB);
            writeln(lst,' ',ms,'[',r,']=',yy:6:5,'  ',BufB);
          end;
        y:=y+Belem[r,0]*yy;
    end;{for}
End; { Scrie}


Procedure Anul(Var Belem:BstocB);

Begin
  for r:=1 to jmax do
      begin
        for j:=0 to jmax do
            Belem[r,j]:=0;
      end;
End; { Anul }


Procedure Wrt(BufB:Str255);
```

```
Begin
  writeln(BufB);
  writeln(lst,BufB);
End; { Wrt }


Procedure Readd;

Begin
  Wrt('   edges diameter, ed=?  (1)');
  Wrt('      or');
  write('   combinatorial diameter, cd=? (2)    ');
  write(lst,'   combinatorial diameter, cd=? (2)    ');
  repeat
   readln(r);writeln(lst,r);
  until r in [1,2];
  { w }
  case r of
   1:begin
       write(' ed=? ');readln(d);
       writeln(lst,' ed=? ',d);
     end;
   2:begin
       write(' cd=? ');readln(d);
       writeln(lst,' cd=? ',d);
       d:=d-1;
     end;
  end;
End; { Readd }



Procedure  Input;

Begin
  Wrt('');
  Wrt('      INPUT DATA  ');Wrt('');
  Wrt(' Binary dendrimer AxBy ');
  Wrt ('   A=branching nodes ');
  Wrt ('   B=monovalent nodes attached to A nodes ');
  Wrt('');
  write('   A=? ');readln(A);
  for J:=1 to Length(A) do
      A[j]:=UpCase(A[j]);
  writeln(lst,'   A=? ',A);
  if Length(A)>1 then
     A:='['+A+']';
  write('   B=? ');readln(B);
  for j:=1 to Length(B) do
      B[j]:=UpCase(B[j]);
  writeln(lst,'   B=? ',B);
  if Length(B)>1 then
     B:='['+B+']';
  write(' pi bonds( around A)=? ');
  readln(pib);
```

```
   writeln(lst,' pi bonds( around A)=? ',pib);
   Wrt(' Dendrimer diameter');
   Readd;
   write(' Chemical degree, chk=? ');readln(chk);
   writeln(lst,' Chemical degree, chk=? ',chk);
   write(' Graph theoretical degree, k=? ');readln(k);
   writeln(lst,' Graph theoretical degree, k=? ',k);
 End; { Input }


Procedure Formul;

 Begin
   ri:=Frac(d/2);
   if ri=0 then zi:=0
    else
   zi:=1;
   rm:=Round((d-zi)/2);
   x:=1;
   for r:=1 to rm do
       x:=x+Round(k*exp((r-1)*ln(k-1)));
   x:=x+zi*Round(exp(rm*ln(k-1)));
   xx:=chk-(pib+k);
   yyi:=(k-2)*x+2;
   yi:=x*xx+yyi;
   Wrt('');Wrt ('      OUTPUT DATA  ');Wrt('');
   if xx=0 then
     begin
       writeln(' Graph theoretical formula : ',A,x,' ',B,yi);
       writeln(lst,' Graph theoretical formula : ',A,x,' ',B,yi);
       writeln(' Chemical formula : ',A,x,' ',B,yi);
       writeln(lst,' Chemical formula : ',A,x,' ',B,yi);
       Wrt('');
     end
   else
     begin
       if xx=1 then
          begin
            writeln (' Graph theoretical formula :
                       (',A,B,')',x,' ',B,yyi);
            writeln (lst,' Graph theoretical formula :
                          (',A,B,')',x,' ',B,yyi
            writeln (' Chemical formula : ',A,x,' ',B,yi);
            writeln (lst,' Chemical formula : ',A,x,' ',B,yi);
            Wrt('');
          end
       else
          begin
            writeln (' Graph theoretical formula :
                       (',A,B,xx,')',x,' ',B,yyi);
            writeln (lst,' Graph theoretical formula :
                       (',A,B,xx,')',x,' ',B,yy
            writeln (' Chemical formula : ',A,x,' ',B,yi);
            writeln (lst,' Chemical formula : ',A,x,' ',B,yi);
            Wrt('');
          end;
     end;
```

```
  End; { Formul }

Procedure Bmon;

 Begin
   rm:=Round(d/2+1);
   Anul(Belem);
   for r:=1 to rm do
     begin
      if r=1 then Prima(Belem);
      if r=rm then EndB(Belem);
      if (r<>1) and (r<>(rm)) then Corp(Belem);
     end; {for}
   Scrie('B',Belem,rm,y,bufB);
 End; { Bmon }


Procedure Fmon;

 Begin
   rm:=Round(d/2+1);
   Anul(Belem);
   for r:=1 to rm do
      begin
        if r=1 then PrimaF(Belem);
        if r=rm then EndF(Belem);
        if (r<>1) and (r<>rm) then CorpF(Belem);
      end; {for}
   Scrie('F',Belem,rm,y,bufB);
 End; { Fmon }


Procedure Bdi;

 Begin
   rm:=Round((d+1)/2);
   Anul(Belem);
   for r:=1 to rm do
      begin
        if r=1 then PrimaDi(Belem);
        if r=rm then EndBDi(Belem);
        if (r<>1) and (r<>rm) then CorpDi(Belem);
      end; {for}
   Scrie('B',Belem,rm,y,bufB);
 End;{ Bdi }


Procedure Fdi;

 Begin
     rm:=Round((d+1)/2);
    Anul(Belem);
    for r:=1 to rm do
        begin
          if r=1 then PrimaFDi(Belem);
          if r=rm then EndFDi(Belem);
```

```
          if (r<>1) and (r<>rm) then CorpFDi(Belem);
        end; {for}
    Scrie('F',Belem,rm,y,bufB);
End; { Fdi }


Procedure Wait;

Begin
writeln;
write('  Press any key  ');
repeat

until KeyPressed;
End;


BEGIN
   repeat
      begin
        ClrScr;
        writeln('     MAIN MENU ');
        writeln;
        writeln(' 1) Compute Y indices for dendrimer ? ');
        writeln(' 2) Compute compactness for molecule ? ');
        write(' 3) Exit ? ');
        repeat
         readln(ct);
        until ct in [1,2,3];
        case ct of
         1:begin
             Input;
             Formul;
             if ri=0 then
               begin
                  Wrt ( '   Matrix B (Monocentric) : ');
                  Bmon;
                  writeln(' BY= ',y:10:5);
                  writeln(lst,' BY= ',y:10:5);
                  Wrt('');
                  Wrt ( '   Matrix F (Monocentric) : ');
                  Fmon;
                  writeln(' FY= ',y:10:5);
                  writeln(lst,' FY= ',y:10:5);
                  Wrt('');
               end
             else
               begin
               Wrt('   Matrix B (Dicentric):');
               Bdi;
               writeln (' BY= ',y:10:5) ;
               writeln (lst,' BY= ',y:10:5) ;
               Wrt('');
               Wrt('   Matrix F (Dicentric):');
               Fdi;
               writeln (' FY= ',y:10:5);
               writeln (lst,' FY=',y:10:5);
```

```
            Wrt('');
            end;
            Wait;
            end;
        2:begin
            ClrScr;
            Wrt('');
            Wrt ('      INPUT DATA    ');Wrt('');
            Wrt (' Graph diameter of testing molecule ');
            Readd;
            ri:=Frac(d/2);
            write(' Name of molecule : ? ');
            readln(bufB);
            writeln(lst,' Name of molecule : ? ',bufB);
            k:=4;chk:=4;A:='C';B:='H';
            write(' FY (1)  or BY (2) -value=? ') ;
            repeat
               readln(rr);
            until rr in [1,2];
            writeln(lst,' FY (1)  or BY (2) -value=? ',r);
            case rr of
                1:begin
                    write(' FY( for ',bufB,')=? ');readln(fyt);
                    writeln(lst,' FY( for ',bufB,')=? ',fyt:8:5);
                    Wrt('');Wrt('      OUTPUT DATA  ');Wrt('');
                    if ri=0 then
                      begin
                       Fmon;
                      end
                    else
                    Fdi;
                    writeln (' FY(dendrimer)=',y:10:5);
                    writeln (lst,' FY(dendrimer)=',y:10:5);
                  end;
                2:begin
                    write(' BY( for ',bufB,')=? ');readln(fyt);
                    writeln(lst,' BY( for ',bufB,')=? ');
                    writeln(lst,fyt:8:
                    Wrt('');Wrt('      OUTPUT DATA  ');Wrt('');
                    if ri=0 then
                       begin
                         Bmon;
                       end
                    else
                    Bdi;
                    writeln (' BY(dendrimer)=',y:10:5);
                    writeln (lst,' BY(dendrimer)=',y:10:5);
                  end;
            end;
            yc:=fyt/y;
            writeln ('    YC=',yc:10:5);
            writeln (lst,'    YC=',yc:10:5);
            Wrt('');
            Wait;
          end;
      end;
  end;

  until ct=3;
END. { Program }
```