Записки научных
семинаров ПОМИ
Том 281, 2001 г.

N. A. Vavilov

DO IT YOURSELF STRUCTURE CONSTANTS FOR LIE ALGEBRAS OF TYPES E_l

ABSTRACT. We compare two algorithms to compute structure table of the Lie algebras of type E_l with respect to a Chevalley base: the usual inductive algorithm and an algorithm based on the use of Frenkel-Kac cocycle. It turns out that Frenkel-Kac algorithm is several dozen times faster but with the 'natural' choice of the bilinear form and the sign function it does not give the result in a positive Chevalley base. We show how to modify the sign function to get the right choice of structure constants. Cohen, Griess and Lisser obtained similar result by varying the bilinear form. We recall the hyperbolic realisation of root systems of type E_l which dramatically simplify calculations as compared with the usual Euclidean realisation. We give Mathematica definitions which realise root systems and implement both the inductive and Frenkel-Kac algorithms. Using these definitions one can compute the whole structure table for E_8 within a quarter of an hour at a home computer. At the end of the paper we reproduce tables of roots according to `HeightLex` and the resulting tables of structure constants.

In this paper we give "Mathematica" definitions for the root systems of types E_l and the structure constants of the complex simple Lie algebras of these type in a Chevalley base. Also we reproduce the resulting tables. Our algorithm based on the use of the Frenkel-Kac cocycle is **several dozen times faster**, than the usual inductive algorithms. It allows to calculate the whole structure table in a few minutes as opposed to many hours of computer time required by the inductive algorithms.

An explicit knowledge of structure constants is essential in many problems in the theory of Chevalley groups, as, for example, the study of conjugacy classes [28], [23], [24], etc., the study of representations [5], [14], generation problems [29], etc. Of course, the only difficult cases are those

The author gratefully acknowledges the support of the *Consiglio Nazionale delle Ricerche* and the *Alexander von Humboldt-Stiftung* during the preparation of the present work. The final stage of the work was partially supported by the RFFI project 00-01-00441.

of the root systems of types F_4 and E_l (observe though that the group of type F_4 is a twisted form of the group of type E_6 and the structure constants for F_4 may be easily derived from those for E_6). In these cases the constants have been calculated several times, compare [25], [5], [23], [24], [14]. As far as we understand all of the published explicit tables are based on the reduction algorithm proposed by J.Tits [35] (see also [6]).

There is a much more direct way – the one based on the *Frenkel–Kac cocycle* – to determine the structure constants for the Lie algebras corresponding to a simply laced root system Φ (see [12], [30], [27], [19], [13]), which is both more elegant and computationally more efficient, than the inductive algorithms. This construction is essentially as follows. One takes a bilinear form f on the lattice of roots $Q(\Phi)$ and sets $N(\alpha,\beta) = \varepsilon(\alpha)\varepsilon(\beta)\varepsilon(\alpha+\beta)(-1)^{f(\alpha,\beta)}$, where $\varepsilon(\alpha)$ is a certain "sign" subject to the condition $\varepsilon(\alpha)\varepsilon(-\alpha) = -1$.

However, the choice of structure constants in [14] (namely the one when structure constants are supposed to be positive for all extraspecial pairs, see [35], [6] and 5^0 below) seems to be *the* right choice. For example if one looks at the realisations of the minimal modules for the Chevalley groups of types E_6 and E_7 in the unipotent radicals of the appropriate parabolic subgroups of the groups of types E_7 and E_8, then for this choice of structure constants *all* signs in the action of the fundamental root unipotents $x_\alpha(1)$ and $x_{-\alpha}(1)$ at vectors of the weight base happen to be positive. At the same time the "natural" choices of the form f and the sign ε for the Frenkel-Kac cocycle (see section 5^0) lead to a *different* structure constant table. So to obtain the desirable choice of structure constants we have to change the functions f and/or ε.

The paper is organised as follows. In section 1^0 we mention two concrete situations where the explicit knowledge of the structure constants was required. In section 2^0 we reproduce the hyperbolic construction of the root systems of type E_l. In section 3^0 we recall definition of structure constants (see [6], [17], [30], [1], [2], [29], [32] for more detailed information concerning expression of the structure constants $N_{\alpha\beta ij}$, $\eta_{\alpha\beta}$ arizing in the theory of Chevalley groups in terms of the constants $N(\alpha,\beta)$ for Lie algebras). In 4^0 we recall some basic properties of the structure constants $N(\alpha,\beta)$. In 5^0 we describe an inductive algorithm for the calculation of $N(\alpha,\beta)$ (see [35], [6], [14]). In 6^0 we recall the Frenkel-Kac construction and describe a choice of the functions f and ε which leads to the same structure constant table, as in [14].

Finally in 7^0, 8^0 and 9^0 we collect "Mathematica" definitions for the root systems of type E_l and structure constants $N(\alpha,\beta)$ for the inductive algorithm and the Frenkel-Kac algorithm respectively. At the end of the paper we reproduce tables of roots ordered according to height and lexicographically within a given height and the resulting tables of structure constants. Our tables essentially coincide with those in [14], but the order of roots considered there was not strictly lexicographical (the table for E_8 was not reproduced in [14], but has been available from the authors of that publication upon request – in that table the order deviates even more from the lexicographic one, than in the published tables for E_6 and E_7).

After this work was completely finished the author learned from A.M.Cohen that a similar algorithm has been used in [8] to calculate structure constants for Lie algebras of type E_l (compare also [7]). However in [8] the form f rather than the sign ε has been modified. See also [39], [41], [42] for a visual approach which allows to calculate any single structure constant, both of the Lie algebra itself and of a minimal representation without use of a computer in less than one minute.

1^0. Motivation. This is the first of the proposed series of 5 publications devoted to a computer study of minimal modules for exceptional Chevalley groups. The goal of this work is to list explicitly root elements $x_\alpha(\xi)$, $w_\alpha(\varepsilon)$ and $h_\alpha(\varepsilon)$ and defining equations of the exceptional groups on these modules.

There have been two concrete stimuli to undertake this manual work. The first one of them is described in [36] – [41]. Namely in the approach to the main structure theorems for the Chevalley groups over commutative rings presented in [36]–[38] one has to check that a certain product of root elements fixes a vector of a minimal module, usually a vector $v=(v_\lambda)$ from the orbit of the highest weight vector. It is easy to construct such a product which adds to most of the coordinates expressions of the form $\pm v_\lambda v_\mu \pm v_\mu v_\lambda$, where λ and μ are some weights of the representation. One has to explicitly know the signs of the action to be able to conclude at this point that what we add is actually zero and not just $\pm 2v_\lambda v_\mu$. The same product adds to the remaining coordinates expressions of the form $\pm v_{\lambda_1}v_{\mu_1} \pm v_{\lambda_2}v_{\mu_2} \pm \ldots \pm v_{\lambda_m}v_{\mu_m}$ which coincide – possibly up to signs – with the equations defining the orbit of the highest weight vector. One has to know the equations explicitly to be sure that what we add is actually zero. (As mentioned in [36] there are ways to handle such problems without calculating the action of all root elements at the minimal module, see 4^0 and 6^0 below).

I have encountered analogous difficulty in the analysis of the (2,3)-generation of finite groups of Lie type (see [9]) which we are carrying through with Lino Di Martino [10], [11]. When one tries to construct an involution and an element of order 3 which generate an exceptional Chevalley group of type Φ, the first natural thought is to try to imitate the remarkable "duplication" idea with which Ch.Tamburini has solved the problem for the large classical cases (see, for example, [33], where this idea is developed in a more general context), namely, to construct these elements as products of generating elements of orders 2 and 3 in subgroups of type A_l and some elements from the extended Weyl group $\widetilde{W}(\Phi)$. E.g. in $\mathrm{G}(\mathrm{E}_6, K)$ one constructs at some point a matrix which comes from $\mathrm{G}(\mathrm{A}_5, K)$ in an appropriate representation and a monomial matrix. One knows that both of them lie in some groups of type E_6 conjugate by a diagonal involution. Is it the same group? Again to answer this question you need signs.

The approach of [36]–[41] is a further development of the methods introduced in K-theory of exceptional groups by H.Matsumoto [22] and M.R.Stein [31]. Usually for the applications in the structure theory and K-theory it is enough to consider a Chevalley group just on some very small modules – the *basic* or *minimal* modules. However even for those small modules no attempt to fix signs in the action of the elementary unipotents at the root base has ever been made in this theory: everyone simply writes something like $x_\alpha(\xi)v^\lambda = v^\lambda \pm v^{\lambda+\alpha}$ (see [22], [31], [36]). At some instances (for example in the analysis of the relations among unipotent elements of root type necessary to obtain an invariant presentation of the Steinberg group) this explicit choice of signs becomes vital. Of course, in some smallest cases, like for example, for the group of type G_2 their action has been explicitly tabulated in the literature (see, for example, [26], [18]). For some other cases *some* of the matrices representing $x_\alpha(1)$ were listed in various places, mainly in connection with the study of embeddings among Chevalley groups (see, for example, [34]).

However systematic explicit calculations with the exceptional groups require reliable explicit tables of root elements and defining equations *in corpore* and/or fast algorithms which allow to calculate these things. This urged us to fix signs in the actions above in such a way that they are compatible with a fixed choice – or rather *the* fixed choice – of the structure constants. Knowing these signs we are able to calculate with the elements of the Chevalley groups as matrices of degrees 7, 26, 27, 56, 248, etc. The corresponding algorithms and the resulting tables will appear in the subsequent publications by the author, some of which are joint with E.B.Plotkin, see [37], [40], [41] for a description of the whole project.

2^0. The root systems of type E_l. Here we reproduce a construction of the root systems of type E_l in the $(l+1)$-dimensional Minkowski space due to Yu.I.Manin [21]. This realisation is much more convenient than the usual one since here all of the coordinates are integers. Unlike Manin we stick to the usual numbering of fundamental roots [3].

Let U be a real vector space of dimension $l+1$ with a nondegenerate symmetric bilinear inner product $(\ ,\ ): U \times U \longrightarrow \mathbb{R}$ of signature $(l,1)$. We will denote this space by $\mathbb{R}^{l,1}$. By definition there exists an orthogonal base $e_0, e_1, \dots, e_l$ of U such that $(e_0, e_0) = -1$ while $(e_i, e_i) = 1$ for $1 \leqslant i \leqslant l$. First, let $l = 8$. For a vector v from the space $U \cong \mathbb{R}^{8,1}$ denote by

$$v = \lambda e_0 + \mu_1 e_1 + \dots + \mu_8 e_8$$

its expression as a linear combination of the base vectors. Denote by L the sublattice in U consisting of those vectors v all of whose coordinates $\lambda, \mu_1, \dots, \mu_8$ are integers.

Consider the set Φ consisting of the vectors $v \in L$ whose coordinates satisfy the following system of equations:

$$\begin{cases} 3\lambda - (\mu_1 + \dots + \mu_8) = 0, \\ -\lambda^2 + \mu_1^2 + \dots + \mu_8^2 = 2. \end{cases}$$

It is obvious that all the solutions of this system such that $0 \leqslant \lambda \leqslant 3$, $\mu_1 \geqslant \dots \geqslant \mu_8$, are listed in the following table:

λ	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7	μ_8
0	1	0	0	0	0	0	0	-1
1	1	1	1	0	0	0	0	0
2	1	1	1	1	1	1	0	0
3	2	1	1	1	1	1	1	1

It is easy to check (see [21]) that all the integer solutions of this system are obtained from these ones by the following transformations:

i) Arbitrary permutations of $\mu_1, \dots, \mu_8$,
ii) Simultaneous change of signs by all of the coordinates.

Thus *up to sign* an element of Φ should have one of the following forms:

$$\begin{cases} \beta_{ij} = e_i - e_j, \quad i > j, \\ \gamma_{ijh} = e_0 + e_i + e_j + e_h, \\ \eta_{ij} = 2e_0 + e_1 + \ldots + \widehat{e_i} + \ldots + \widehat{e_j} + \ldots + e_8, \\ \zeta_i = 3e_0 + e_1 + \ldots + 2e_i + \ldots + e_8, \end{cases} \tag{1}$$

where indices $i, j, h = 1, \dots, 8$ are pair-wise distinct and the hat "$\widehat{\ }$" over a summand means that this summand has to be omitted. Thus the cardinality of Φ equals $2\mathrm{C}_8^2 + 2\mathrm{C}_8^3 + 2\mathrm{C}_8^2 + 2\mathrm{C}_8^1 = 240$, which is already a good omen.

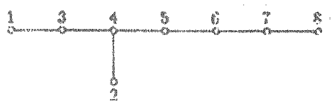
Indeed, Φ is a root system in the hyperplane V orthogonal to the vector $3e_0 + e_1 + \ldots + e_8$. First, note that Minkowski inner product on U induces a positive definite inner product on V. In fact, we have only to check that the scalar square of any non-zero vector from V is positive. Let $v = \lambda e_0 + \mu_1 e_1 + \ldots + \mu_8 e_8 \in V$, $v \neq 0$, be such a vector. Then $3\lambda = \mu_1 + \ldots + \mu_8$ and, since $v \neq 0$, at least one of μ_i is non-zero. Then

$$9(v, v) = 9(\mu_1^2 + \ldots + \mu_8^2) - (\mu_1 + \ldots + \mu_8)^2 =$$
$$= \mu_1^2 + \ldots + \mu_8^2 + (\mu_1 - \mu_2)^2 + (\mu_1 - \mu_3)^2 + \ldots + (\mu_7 - \mu_8)^2 > 0.$$

Now consider the following eight vectors from Φ:

$$\begin{cases} \alpha_1 = e_2 - e_1, \\ \alpha_2 = e_0 + e_1 + e_2 + e_3, \\ \alpha_i = e_i - e_{i-1}, \quad i = 3, \ldots 8. \end{cases} \tag{2}$$

These vectors are linearly independent and thus Φ generates V. The scalar squares of the vectors from Φ are equal to 2 by definition and thus the crystallographic condition $2(\alpha, \beta)/(\alpha, \alpha) \in \mathbb{Z}$ follows from the fact that all inner products (α, β), $\alpha, \beta \in \Phi$, are integers. It is easy to check that $w_\alpha(\Phi) \subseteq \Phi$ for any $\alpha \in \Phi$ (this is obvious for $\alpha = \alpha_i$, $i \neq 2$; may be verified directly for $\alpha = \alpha_2$ and for the general case follows from the fact that w_{α_i}, $1 \leqslant i \leqslant 8$, generate a group which acts transitively on Φ). Thus Φ is a root system. Since the inner products of the vectors $\alpha_1, \dots, \alpha_8$ are the ones required by the Dynkin diagram of type E_8,



this is actually a root system of type E_8 and $\Pi = \{\alpha_1, \dots, \alpha_8\}$ may be chosen as its fundamental system. The roots β_i, γ_{ijh}, η_{ij} and ζ_i listed above form the set Φ^+ of positive roots with respect to the fundamental system Π.

Now to obtain a root system of type E_7 one has to take those roots from E_8, for which $\mu_8 = 0$ (or, what is the same, for which α_8 does not appear in their expression as a linear combination of the fundamental roots). Analogously, as a root system of type E_6 one may take those roots from E_8, for which $\mu_7 = \mu_8 = 0$ (or, what is the same, for which α_7 and α_8 do not appear in their expression as a linear combination of the fundamental roots).

We will use *Dynkin notation* for the roots of E_8. Namely, if

$$\alpha = p\alpha_1 + q\alpha_2 + r\alpha_3 + s\alpha_4 + t\alpha_5 + u\alpha_6 + v\alpha_7 + w\alpha_8$$

for some integers p, q, r, s, t, u, v, w, then we write

$$\alpha = \begin{smallmatrix} prstuvw \\ q \end{smallmatrix}.$$

Of course we will describe the roots from E_7 and E_6 by the same sort of picture.

The sum of the coefficients

$$\mathrm{ht}(\alpha) = p + q + r + s + t + u + v + w$$

is called the *height* of the root α. Every root system has a unique root of maximal height with respect to a given fundamental system. This root is called the *maximal* or the *highest* root. Thus in our notation the maximal roots of E_6, E_7 and E_8 have the form

$$2e_0 + e_1 + \ldots + e_6 = (2,1,1,1,1,1,1) = \begin{smallmatrix} 12321 \\ 2 \end{smallmatrix},$$

$$2e_0 + e_2 + \ldots + e_7 = (2,0,1,1,1,1,1,1) = \begin{smallmatrix} 234321 \\ 2 \end{smallmatrix},$$

$$3e_0 + e_1 + \ldots + e_7 + 2e_8 = (3,1,1,1,1,1,1,1,2) = \begin{smallmatrix} 2465432 \\ 3 \end{smallmatrix}.$$

The corresponding heights are 11, 17 and 29 respectively. To save space in the tables we write roots in their *string Dynkin forms* in which the root α above is rendered by the sequence of its coefficients $pqrstuvw$. Thus, for example, the string Dynkin forms of the maximal roots are 122321, 2234321 and 23465432 respectively.

In what follows we consider a total ordering of positive roots which is *regular* (i.e. a root of smaller height always precedes a root of larger height) and *lexicographic* at the roots of a given height. We will write $\alpha \prec \beta$ if α precedes β with respect to this ordering. By definition this means that either $\mathrm{ht}(\alpha) < \mathrm{ht}(\beta)$ or $\mathrm{ht}(\alpha) = \mathrm{ht}(\beta)$ and the integer represented by the string Dynkin form of α is *bigger* than the integer represented by the string Dynkin form of β (that is why we have to reverse the order of roots in 7^0). The tables of positive roots in the Dynkin form so ordered appear at the end of this paper.

3^0. The structure constants. Our general references for the contents of this section are [4], [6], [16], [32]. Let Φ be any reduced irreducible root system; L be a complex simple Lie algebra of type Φ; H be a Cartan subalgebra of L. Consider the root subspaces decomposition

$$L = H \oplus \sum_{\alpha\in\Phi} L_\alpha,$$

of L with respect to H. Recall that the root spaces L_α are one-dimensional H-invariant subspaces in L. Choose $e_\alpha \in L_\alpha$, $e_\alpha \neq 0$. One can always normalise e_α in such a way that $[e_\alpha, e_{-\alpha}] = h_\alpha$ is the coroot, corresponding to α. Then $[h_\alpha, e_\beta] = \langle\alpha,\beta\rangle e_\beta$ and $[h_\alpha, h_\beta] = 0$. If we choose a fundamental root system Π in Φ, then the correspponding fundamental coroots h_α, $\alpha \in \Pi$, form a base of H and the formulae above show that the only relevant structure constants are those which arise in the products $[e_\alpha, e_\beta]$ for pairs of roots α, β which are not opposite. Let $\alpha, \beta \in \Phi$ be such that $\alpha + \beta \neq 0$. Set

$$[e_\alpha, e_\beta] = N(\alpha,\beta) e_{\alpha+\beta}.$$

The numbers $N(\alpha,\beta)$ are called the *structure constants* of the Lie algebra L with respect to the system e_α. Strictly speaking, these constants form only a part of the structure tensor of the algebra, but as we've seen above, the only interesting one, so we'll stick to this usage.

A base of the form e_α, $\alpha \in \Phi$; h_β, $\beta \in \Pi$, is usually called a *Weyl base*. One may normalise a Weyl base in such a way that all the structure constants are integers. Such a normalisation of a Weyl base is called a

Chevalley base. For a Chevalley base one has $N(\alpha,\beta)=0,\pm1,\pm2,\pm3$. The elements e_α, $\alpha\in\Phi$, forming part of a Chevalley base are called a *Chevalley system* (see [4], [15]).

Recall that a root system Φ is called *simply laced* if $\langle\alpha,\beta\rangle=0,\pm1$ for any two linearly independent roots $\alpha,\beta\in\Phi$. For an irreducible root system this is equivalent to saying that all roots of Φ have the same length. For a simply laced system always $N(\alpha,\beta)=0,\pm1$, so the only problem is to determine the *signs* of the structure constants.

Now, let K be a field and $G=G(\Phi,K)$ be a Chevalley group of type Φ over K. For any root $\alpha\in\Phi$ and any element $\xi\in K$ denote by $x_\alpha(\xi)$ the corresponding elementary root unipotent in G. For elements x, y of a group G we denote by $[x,y]$ their commutator $xyx^{-1}y^{-1}$. Let now $\alpha,\beta\in\Phi$, $\alpha+\beta\neq0$, and $\xi,\eta\in K$. Then the *Chevalley commutator formula* asserts that

$$[x_\alpha(\xi),x_\beta(\eta)]=\prod x_{i\alpha+j\beta}(N_{\alpha\beta ij}\xi^i\eta^j),$$

where the product on the right hand side is taken over all roots of the form $i\alpha+j\beta\in\Phi$; $i,j\in\mathbb{N}$, in a fixed order and the constants $N_{\alpha\beta ij}$ do not depend on ξ,η (though they may in general depend on the order). The integers $N_{\alpha\beta ij}$ are called the *structure constants* of the Chevalley group and they take only the values ±1, ±2, ±3. It is obvious that $N_{\alpha\beta11}=N(\alpha,\beta)$ are just the structure constants of L in the corresponding Chevalley base.

If all roots in the subsystem generated by α and β have equal lengths, then at most one positive linear combination of α and β may be a root, namely their sum. This shows that for simply laced root systems – in particular, for the root systems of type E_l – the structure constants for the group are just the structure constants for the corresponding Lie algebra. For other types they may be easily expressed via these structure constants (see [6], [32], [29], [1], [2], [17], [29], [37]).

4^0. Properties of the structure constants. Here we recall some well-known properties of the numbers $N(\alpha,\beta)$ *with respect to a Chevalley base* (see, for example, [6]). The following properties are obvious:

$$N(\alpha,\beta)=0,\quad\text{if}\quad\alpha+\beta\notin\Phi,\alpha+\beta\neq0,\tag{3}$$

$$N(\alpha,\beta)=-N(\beta,\alpha),\tag{4}$$

$$N(\alpha,-\beta)=N(\beta,-\alpha).\tag{5}$$

Let now $-t\alpha+\beta$ be the first member of the α-series of roots through

β. This means that $-t\alpha+\beta\in\Phi$ but $-(t+1)\alpha+\beta\notin\Phi$. Then

$$N(\alpha,\beta)=\pm(t+1), \tag{6}$$

$$N(\alpha,\beta)N(-\alpha,-\beta)=-(t+1)^2. \tag{7}$$

The proof of the following two formulae uses Jacobi identity. First, let α,β,γ be three roots such that $\alpha+\beta+\gamma=0$. Then

$$N(\alpha,\beta)/(\gamma,\gamma)=N(\beta,\gamma)/(\alpha,\alpha)=N(\gamma,\alpha)/(\beta,\beta). \tag{8}$$

Now, let $\alpha,\beta,\gamma,\delta$ be four roots such that $\alpha+\beta+\gamma+\delta=0$ but no pair of these roots are opposite. Then

$$\begin{aligned}N(\alpha,\beta)N(\gamma,\delta)/(\alpha+\beta,\alpha+\beta)+N(\beta,\gamma)N(\alpha,\delta)/(\beta+\gamma,\beta+\gamma)+\\ N(\gamma,\alpha)N(\beta,\delta)/(\gamma+\alpha,\gamma+\alpha)=0.\end{aligned} \tag{9}$$

Consider the matrix $N=(N(\alpha,\beta))$, $\alpha,\beta\in\Phi^+$. Formula (4) says that this matrix is antisymmetric. Now Formula (5) (or (7)) expresses the matrix $(N(\alpha,\beta))$, $\alpha,\beta\in\Phi^-$ via N. The same formula expresses $(N(\alpha,\beta))$, $\alpha\in\Phi^-$, $\beta\in\Phi^+$ via $(N(\alpha,\beta))$, $\alpha\in\Phi^+$, $\beta\in\Phi^-$. It remains only to express $(N(\alpha,\beta))$, $\alpha\in\Phi^+$, $\beta\in\Phi^-$ via N. Indeed, applying Formula (8) we get

$$N(\alpha,\beta)=\begin{cases}N(\beta,-\alpha-\beta)(\alpha+\beta,\alpha+\beta)/(\alpha,\alpha), & \text{if }\ \alpha+\beta\in\Phi^+,\\ N(-\alpha-\beta,\alpha)(\alpha+\beta,\alpha+\beta)/(\beta,\beta), & \text{if }\ \alpha+\beta\in\Phi^-.\end{cases} \tag{10}$$

This completely reduces calculation of all structure constants $N(\alpha,\beta)$ to the calculation of N.

For simply laced systems the formulae above may be substantially simplified. Namely, when all roots of Φ have the same length, then for all roots $\alpha,\beta,\alpha+\beta\in\Phi$ one necessarily has $\alpha-\beta\notin\Phi$, so that $t=0$ and $N(\alpha,\beta)=\pm1$. Thus Formulae (8) and (9) become

$$N(\alpha,\beta)=N(\beta,\gamma)=N(\gamma,\alpha),\quad \text{if}\quad \alpha+\beta+\gamma=0, \tag{11}$$

and

$$\begin{aligned}N(\alpha,\beta)N(\gamma,\delta)+N(\beta,\gamma)N(\alpha,\delta)+N(\gamma,\alpha)N(\beta,\delta)=0,\\ \text{if}\quad \alpha+\beta+\gamma+\delta=0.\end{aligned} \tag{12}$$

Observe that in the last case only two of the summands can be nonzero. Indeed, $N(\alpha,\beta)\neq0$ and $N(\beta,\gamma)\neq0$ implies that $\alpha+\beta,\beta+\gamma\in\Phi$.

Thus β forms angles $2\pi/3$ with both α and γ. Suppose $N(\alpha,\gamma)\neq 0$. Then the angle between α and γ also equals $2\pi/3$ and thus α,β,γ lie in one plane. But then $\delta=0$, a contradiction. Thus (9) is equivalent to a piece of the 2-cocycle equation:

$$N(\beta,\gamma)N(\alpha,\beta+\gamma)=N(\alpha+\beta,\gamma)N(\alpha,\beta). \tag{13}$$

This fact plays crucial role in what follows. Actually this equation alone settles many cases where [36] refers to the explicit knowledge of signs.

5^0. Inductive algorithms for calculation of structure constants. There is a natural inductive procedure based on the formulae from the preceding section, which expresses all structure constants in terms of the structure constants for certain pairs of roots, the so called extraspecial pairs, see [35], [6]. In turn signs of structure constants for the extraspecial pairs may be taken arbitrarily (it is customary to take all of these signs to be "+").

Namely, recall that a pair (α,β) of positive roots is called *special* if $\alpha+\beta\in\Phi$ and $\alpha\prec\beta$ with respect to the ordering described in 2^0. A pair (α,β) is called *extraspecial* if it is special and for any special pair (γ,δ) such that $\alpha+\beta=\gamma+\delta$ one has $\alpha\prec\gamma$. Then the values of structure constants $N(\alpha,\beta)$ may be taken arbitrarily at the extraspecial pairs and all other structure constants may be uniquely determined using only properties (4), (7), (8) and (9) (see [6], p.58–60).

Here is a variant of this procedure from the paper [14] which works for simply laced systems. Let $\Pi=\{\alpha_1,\dots,\alpha_l\}$. Set $N(\alpha_i,\beta)=1$ if $\alpha_i+\beta\in\Phi^+$ and there is no $j<i$ such that $\alpha_i+\beta=\alpha_j+\bar\beta$ for some $\bar\beta\in\Phi^+$. If $\alpha,\beta,\alpha+\beta\in\Phi^+$, let i be minimal such that $\alpha+\beta=\alpha_i+\bar\beta$ for some $\bar\beta\in\Phi^+$. Then either $\alpha-\alpha_i\in\Phi^+$, or $\beta-\alpha_i\in\Phi^+$, but not both. Now $N(\alpha,\beta)$ is expressed as follows

$$N(\alpha,\beta)=\begin{cases}-N(\alpha_i,\beta-\alpha_i)N(\beta-\alpha_i,\alpha), & \text{if } \beta-\alpha_i\in\Phi^+,\\[2mm] \quad N(\alpha_i,\alpha-\alpha_i)N(\alpha-\alpha_i,\beta), & \text{if } \alpha-\alpha_i\in\Phi^+.\end{cases} \tag{14}$$

We believe that the formula above is a corrected version of the formulae on page 410 of [14]. The formulae in [14] contain two misprints. For the second of them it is obvious, since β does not appear on the right hand side at all. Quite mysteriously the following text refers to that formula (it should be substituted by the following phrase: "since either $\beta-\beta_i$ comes before β in the ordering or $\alpha-\alpha_i$ comes before α in the ordering").

It is worse with the first formula for $N(\alpha,\beta)$ since apparently a sign is missing. Here is an argument from a letter of E.B.Plotkin to the author dated 05 March 1990. Let $\alpha+\beta=\alpha_i+\bar\beta$. Then applying the formula (12) to $\alpha+\beta-\alpha_i-\bar\beta=0$ we get

$$N(\alpha,\beta)N(-\alpha_i,-\beta)+N(\beta,-\alpha_i)N(\alpha,-\bar\beta)+\ N(-\alpha_i,\alpha)N(\beta,-\bar\beta)=0.$$

If $\beta-\alpha_i\in\Phi^+$, then $\alpha-\alpha_i\notin\Phi^+$ and thus $N(-\alpha_i,\alpha)=0$. Thus

$$N(\alpha,\beta)N(-\alpha_i,-\bar\beta)+N(\beta,-\alpha_i)N(\alpha,-\bar\beta)=0.$$

But since $N(-\alpha_i,-\bar\beta)=N(\bar\beta,\alpha_i)=-1$, one has $N(\alpha,\beta)=N(\beta,-\alpha_i)N(\alpha,-\bar\beta)$. Now $N(\beta,-\alpha_i)=N(\beta-\alpha_i,\alpha_i)$ since $\beta-\alpha_i\in\Phi^+$.

Recall that $N(\alpha,\beta)=N(\beta,\gamma)=N(\gamma,\alpha)$ if $\alpha+\beta+\gamma=0$. Since $\alpha+(\beta-\alpha_i)-\bar\beta=0$, one has $N(-\bar\beta,\alpha)=N(\alpha,\beta-\alpha_i)$ and thus $N(\alpha,-\bar\beta)=N(\beta-\alpha_i,\alpha)$. Finally, if $\beta-\alpha_i\in\Phi^+$ one has

$$N(\alpha,\beta)=N(\beta-\alpha_i,\alpha_i)N(\beta-\alpha_i,\alpha)=-N(\alpha_i,\beta-\alpha_i)N(\beta-\alpha_i,\alpha).$$

Of course this does not influence tables in the paper [14] – the tables are correct.

6^0. Frenkel-Kac cocycle. However for simply laced root systems there is a much more elegant and efficient way to calculate structure constants, which is due to I.Frenkel and V.Kac [12] (see also [30], [27], [19], [13]).

Let $Q(\Phi)$ be the *root lattice* of the root system Φ, i.e. the $\mathbf{Z}$-lattice spanned by Φ endowed with the inner product induced by the inner product in V. Any fundamental root system $\Pi=\{\alpha_1,\dots,\alpha_l\}$ forms a $\mathbf{Z}$-base of $Q(\Phi)$. In particular, the lattice $Q(\Phi)$ is *even*, i.e. $(x,x)\in 2\mathbf{Z}$ for all $x\in Q(\Phi)$.

Now denote by f any bilinear $\mathbf{Z}$-valued form on $Q(\Phi)$ such that

$$(x,y)\equiv f(x,y)+f(y,x)\pmod 2$$

and, moreover,

$$\frac{1}{2}(x,x)\equiv f(x,x)\pmod 2$$

for all $x,y\in Q(\Phi)$ (there is a funny misprint in the first of these formulae in [30]).

It is obvious that such a form exists. In fact, take Π as a base of $Q(\Phi)$ and define $f(\alpha_i,\alpha_j)$ by the following formulae:

$$f(\alpha_i,\alpha_j)=\begin{cases}0, & \text{if } 1\leqslant i<j\leqslant l,\\ \frac{1}{2}(\alpha_i,\alpha_i), & \text{if } 1\leqslant i\leqslant l,\\ (\alpha_i,\alpha_j), & \text{if } 1\leqslant j<i\leqslant l.\end{cases}\tag{15}$$

Of course, usually the inner product is normalised so that $(\alpha_i,\alpha_i)=2$ and thus $f(\alpha_i,\alpha_i)=1$.

Let $\varepsilon(\alpha)$ be a function on $Q(\Phi)$ such that $\varepsilon(x)=0$ if $x\notin\Phi$ and $\varepsilon(\alpha)\varepsilon(-\alpha)=-1$ for all $\alpha\in\Phi$. Now for all linearly independent $\alpha,\beta\in\Phi$ we may set

$$N(\alpha,\beta)=\varepsilon(\alpha)\varepsilon(\beta)\varepsilon(\alpha+\beta)(-1)^{f(\alpha,\beta)}. \tag{16}$$

It is easy to show that these numbers actually form a system of structure constants for a Lie algebra of type Φ.

For example one may set $\varepsilon(x)$ to be the sign of a vector $x\in Q(\Phi)$ with respect to the fixed fundamental system Π:

$$\varepsilon(x)=\begin{cases}+1, & \text{if } x\in\Phi^+,\\ -1, & \text{if } x\in\Phi^-,\\ 0, & \text{if } x\notin\Phi.\end{cases} \tag{17}$$

Unfortunately this natural sign does not lead to the choice of structure constants we want. That's why we have to modify the sign as follows. Consider the canonical ordering of positive roots described in 2^0. Then we set $\varepsilon(\alpha)$ to be -1 if the position of α in Φ^+ with respect to this ordering falls into one of the following sets:

$$\{24,26,28,29,30,32,35\}$$

for the system E_6;

$$\{28,32,35,37,38,41,42,44,47,49,50,52,54,55,58,59,61,63\}$$

for the system E_7 and

$$\{32,37,40,44,45,49,51,53,56,58,60,61,63,65,66,68,70,71,73,77,$$
$$78,80,82,86,87,88,89,94,95,96,97,101,102,103,107,108,111,113\}$$

for the system E_8. For all other positive roots we set $\varepsilon(\alpha)=+1$. These values of $\varepsilon(\alpha)$ were determined by a computer program to ensure that the values of structure constants $N(\alpha,\beta)$ at the extraspecial pairs are all $+1$ (see 9^0).

7^0. Program: root systems. In this section we give "Mathematica" definitions for the root system of types E_l in the hyperbolic realisation described in 2^0. As usual the text in typewriter type represents the actual computer program. All objects whose names start with a capital letter are internal “Mathematica” objects described in [43].

We define Kronecker’s delta by `delta[i_,j_]:= If[i==j,1,0]`. Now we define the standard base in the $(l+1)$-dimensional Minkowski space:

```
hort[l_,j_]:=Block[{i},Table[delta[i,j],{i,l+1}]]
        /; 1<=j<=l+1;
```

Here `hort` stands for ‘hyperbolic ort’, since we refer to the Minkowski space as *hyperbolic*. This terminology seems to be more consistent with the usage of the word ‘hyperbolic’ outside algebra, than reserving the word ‘hyperbolic’ for the spaces of maximal Witt index – the *Artin spaces*. Of course, for planes the two meanings coincide. For the construction of the roots we need another hyperbolic vector, all of whose coordinates are 1’s, the ‘hyperbolic unit row’:

```
hurow[l_]:=Table[1,{l+1}];
```

The ‘hyperbolic inner product’ is defined as follows

```
hip[u_,v_]:=Block[{i},-u[[1]]*v[[1]]+Sum[u[[i]]*v[[i]],
                {i,2,Length[u]}]] /; Length[u]==Length[v];
```

Now we can define all the roots listed in (1) as follows. Of course the value of l which was implicit in the notation of 2^0 should be now explicitly introduced as a parameter. Notice the shift of indices by one, so that we actually return to the numbering from 2^0.

```
beta[l_,i_,j_]:=hort[l,i+1]-hort[l,j+1] /; i!=j!=0;
gamma[l_,i_,j_,h_]:=hort[l,1]+hort[l,i+1]
            +hort[l,j+1]+hort[l,h+1] /; i!=j!=h!=0;
eta[7,i_]:=hurow[7]+hort[7,1]-hort[7,i+1] /; i!=0;
eta[8,i_,j_]:=hurow[8]+hort[8,1]-hort[8,i+1]-hort[8,j+1]
                /; i!=0 && j!=0 && i!=j;
zeta[8,i_]:=hurow[8]+2hort[8,1]+hort[8,i+1] /; i!=0;
```

Finally we can directly define the root systems in terms of these vectors. Of course, it is enough to have the lists of *positive* roots. The following definitions are self-explanatory. `Flatten` is used here to suppress the nested structure created by the presence of several iteration variables. Of course, `hroots` stands for ‘hyperbolic roots’, the name `roots` being reserved for the Dynkin form of the roots.

```
betaroots[l_]:=Block[{i,j},Flatten[Table[
               beta[l,i,j],{i,l},{j,i-1}],1]];
gammaroots[l_]:=Block[{i,j,h},Flatten[Table[
       gamma[l,i,j,h],{i,l},{j,i+1,l},{h,j+1,l}],2]];
etaroots[7]=Block[{i},Table[eta[7,i],{i,7}]];
etaroots[8]=Block[{i,j},Flatten[Table[
eta[8,i,j],{i,8},{j,i+1,8}],1]];
zetaroots[8]=Block[{i},Table[zeta[8,i],{i,8}]];
hroots[6]=Join[betaroots[6],gammaroots[6],
               {hurow[6]+hort[6,1]}];
hroots[7]=Join[betaroots[7],gammaroots[7],etaroots[7]];
hroots[8]=Join[betaroots[8],gammaroots[8],etaroots[8],
               zetaroots[8]];
```

Notice the use of '`=`' and not '`:=`' in these definitions. Since the lists of roots are short we prefer to form them as actual lists rather than define a condition for a vector to belong to such a list – it is much faster.

Next we define the fundamental subsystems of the root systems. As in 2^0 set

```
alpha[l_,1]:=beta[l,2,1];
alpha[l_,2]:=gamma[l,1,2,3];
alpha[l_,i_]:=beta[l,i,i-1] /; 3<=i<=l;
```

These are precisely the fundamental roots of E_l in the same numbering as in 2^0. The set consisting of all `alpha`'s forms a fundamental root system Π, which we call '`rootbase`':

```
rootbase[l_]:=Block[{i},Table[alpha[l,i],{i,l}]];
```

Now we start converting '`hroots`' to Dynkin form. The following function calculates coefficients of u in linear expression with respect to the fundamental system:

```
rootform[l_][u_]:=LinearSolve[Transpose[rootbase[l]],u];
```

The roots first come raw, later we cook them.

```
rawroots[l_]:=Map[rootform[l],hroots[l]];
```

This is already the set of roots in Dynkin form, but not yet properly ordered. Now we have to order them by height. The height of a root is the sum of its coefficients:

```
height[u_]:=Apply[Plus,u];
```

To select the roots of a given height h we have to define a logical function which takes value '`True`' on the vectors of height h and '`False`' elsewhere. Recall that the names of logical functions tend to finish with 'Q' for 'Question'.

```
heightQ[h_][u_]:=If[height[u]==h,True,False];
```

Now we select from the raw roots the ones which have height h and order them lexicographically, i.e. in the order *reversed* with respect to the usual "Mathematica" order:

```
roots[l_,h_]:=Reverse[Sort[Select[rawroots[l],
            heightQ[h]]]];
```

It remains only to join these lists:

```
roots[6]=Apply[Join,Block[{h},Table[roots[6,h],{h,11}]]];
roots[7]=Apply[Join,Block[{h},Table[roots[7,h],{h,17}]]];
roots[8]=Apply[Join,Block[{h},Table[roots[8,h],{h,29}]]];
```

Again since these lists are not going to change once they are calculated it is much better at this point to assign to `roots[l]` these calculated values by `Set` rather than by `SetDelayed`. Finally we define logical functions which take values `True` or `False` depending on whether u is a positive root or not, a negative root or not, a root or not respectively:

```
positiveQ[l_,u_]:=MemberQ[roots[l],u];
negativeQ[l_,u_]:=MemberQ[roots[l],-u];
rootQ[l_,u_]:=MemberQ[roots[l],u] ||
            MemberQ[roots[l],-u];
```

The actual tables of roots at the end of this paper were obtained from the lists `roots[l]` using the functions `Partition`, `MatrixForm`, `String-Form`, `TeXForm` and a TEX macro `\dynkin` which constructs the Dynkin form of a root from its coefficients.

8^0. Program: inductive algorithm. Since "Mathematica" is an interpreted language, one does not have to worry about writing a step-by-step algorithm. One has only to rewrite properties from 4^0 in "Mathematica" form and set structure constants to be 1 for the extraspecial pairs. We denote the structure constant $N(x, y)$ by `nu[l,x,y]` (here again the parameter l which was implicit in the notations of 3^0 should appear explicitly). Then (3) is expressed as follows:

```
nu[l_,x_,y_]:=0 /; !rootQ[l,x+y];
```

The formula (5) is expressed as follows:

```
nu[l_,x_,y_]:=-nu[l,-x,-y] /; negativeQ[l,x];
```

The following line is essentially formula (10):

```
nu[l_,x_,y_]:=If[positiveQ[l,x+y],
        nu[l,x+y,-y],nu[l,-x-y,x]] /; positiveQ[l,x] &&
        negativeQ[l,y] && rootQ[l,x+y];
```

This completely reduces calculation of `nu[l,x,y]` to the case when both x and y are positive. We may use (4) to reduce to the case when x precedes y. The following function gives the position of a positive root x in the list `roots[l]`. If x is not a root it will remain unevaluated, but it is used only together with explicit condition that x *is* a positive root.

```
position[l_,x_]:=First[First[Position[roots[l],x]]];
```

Our next rule reduces calculation of `nu[l,x,y]` to the special pairs, i.e. to the case when `position[l,x]<position[l,y]`:

```
nu[l_,x_,y_]:=-nu[l,y,x] /;
        positiveQ[l,x] && positiveQ[l,y] &&
        positiveQ[l,x+y] && position[l,x]>position[l,y];
```

Now we have to impose some further condition on `nu` which guarantees reduction to the extraspecial pairs. We have already used Formulae (4), (7), (8) or some of their variants. From [6] (see the beginning of 5^0) we know that we have to impose some variant of (9). Since (14) is suspicious, we proceed directly from (12) which is certainly correct. So our final recurrent condition on `nu` will be:

```
nu[l_,x_,y_]:=Block[{i,z,w}, For[i=1,i<=l,i++,
```

```
        If[positiveQ[l,x+y-IdentityMatrix[l][[i]]],
        z=IdentityMatrix[l][[i]];w=x+y-z;i=9]]; Return[
        (nu[l,y,-z]*nu[l,x,-w]+nu[l,-z,x]*nu[l,y,-w])/
        nu[l,z,w]]] /; positiveQ[l,x] && positiveQ[l,y] &&
        positiveQ[l,x+y] && position[l,x]<position[l,y];
```

It remains only to set `nu[l,x,y]` to be +1 at the extraspecial pairs. One may note that the fundamental roots, i.e. the first l members of the list `list[l]` are just the corresponding rows of the identity matrix.

```
nu[l_,x_,y_]:=1 /; positiveQ[x] && positiveQ[y] &&
        positiveQ[x+y] && position[l,x]<=8 &&
        Block[{j},Apply[And,Table[!positiveQ[x+y-
        IdentityMatrix[l][[j]]],{j,position[l,x]-1}]]];
```

In view of the results from [35], [6] quoted in 4^0 these conditions determine the function `nu` uniquely. To construct the corresponding table we define a function `mu` which depends on the `position` of the roots

```
mu[l_,i_,j_]:=nu[l,roots[l][[i]],roots[l][[j]]];
```

Tables expressing values of these functions are reproduced at the end of this paper. Namely, the constant $N(\alpha,\beta)$ – or, rather, *its sign* – stands at the intersection of the row with the head α and the column with the head β. The roots α and β are rendered by their string Dynkin forms. Since these tables were constructed from the internal "Mathematica" tables of values of the function `mu` in the same way as the lists of roots, i.e. using the functions `MatrixForm`, `StringForm` and `TeXForm`, they should be free of errors. The tablerules were inserted by two mod 5 counters furnished by the function `Mod` and an appropriate alignment preamble written by hand (see [20], p.246–247).

9^0. Program: Frenkel-Kac algorithm. Here we describe a much faster algorithm which leads to the same tables of the structure constants.

First we construct the Gram matrix of the bilinear form f described in 6^0. We call this Gram matrix `igor`.

```
igor[l_]:=Block[{i,j},Table[Which[i<j,0,i==j,1,
        i>j,hip[alpha[l,i],alpha[l,j]]],{i,l},{j,l}]];
```

Now we can easily calculate the value of f at any pair of roots. We denote the form f by **frenkel**. **Dot** stands here for the inner product of the vectors x and y, which corresponds to the diagonal part of **igor**.

```
frenkel[l_,i_,j_]:=Block[{h,k,x=roots[l][[i]],
        y=roots[l][[j]]},Return[Dot[x,y]+
        Sum[x[[h]]*y[[k]]*igor[l][[h,k]],{h,l},{k,h-1}]]];
```

Now we divide the integers from 1 to $|\mathrm{E}_l^+|$ into two parts, `plus[l]`, such that the corresponding roots appear with the sign 'plus' and `minus[l]`, such that the corresponding roots appear with the sign 'minus'.

```
plus[6]=Join[Range[23],{25,27,31,33,34,36}];
plus[7]=Join[Range[27],{29,30,31,33,34,36,39,40,43,45,
        46,48,51,53,56,57,60,62}];
plus[8]=Join[Range[31],{33,34,35,36,38,39,41,42,43,46,
        47,48,50,52,54,55,57,59,62,64,67,69,72,74,75,76,
        79,81,83,84,85,90,91,92,93,98,99,100,104,105,106,
        109,110,112,114,115,116,117,118,119,120}];
range[l_]:=Which[l==6,36,l==7,63,l==8,120,True,"?"];
minus[l_]:=Complement[Range[range[l]],plus[l]];
```

These values of **plus** were calculated by another program. Originally one would set `plus[l_]:=Range[range[l]]`. Then the values of the function **kac** below were calculated for the pairs (i,j), $i = 1,\dots,8$, $j = 1,\dots,N$, where N is the number of positive roots, and compared with the corresponding values of **mu**. As soon as discrepancy would be discovered for some pair (i,j) the position h of the sum of the i-th and the j-th positive root was thrown out of `plus[l]`. It is clear that throwing h out of this list we do not change the signs in the columns with numbers strictly less than j and correct one sign in the j-th column. Proceeding like this we can correct all the signs. Of course, once the lists `plus[l]` are there, one does not need that program anymore.

Further we define the functions of l and one or two more integral variables which express the signs $\varepsilon(\alpha)$ and $\varepsilon(\alpha+\beta)$ respectively:

```
victor[l_,i_]:=Which[MemberQ[plus[l],i],1,
          MemberQ[minus[l],i],-1];
```

```
victor[l_,i_,j_]:=Block[{z=roots[l][[i]]+roots[l][[j]],
          answer="?"},Which[MemberQ[roots[l],z],
          answer=victor[position[l,z]],True,answer=0];
          Return[answer]];
```

Now we can define structure constants by the formula

```
kac[l_,i_,j_]:=victor[l,i]*victor[l,j]*victor[l,i,j]*
          Power[-1,frenkel[l,i,j]];
```

It is easy to check (for a computer) that for all admissible values of l, i and j one has `kac[l,i,j]==mu[l,i,j]`. Of course, the calculation of `kac` is *much* faster, especially if one saves[1] the results of the intermediate calculations of `igor`, `frenkel` and `victor` and then redefines these functions by `Set` instead of `SetDelayed`. As a matter of comparison I could observe that the calculation of just one row somewhere in the middle of the E_8-table by the induction algorithm required one to two hours of computer time, whereas by the Frenkel-Kac algorithm the whole table can be calculated in less than a quarter of an hour (at the final stage we used Mathematica 2.0 for DEC RISC). Soft copies of the tables and the programs in DOS, Mac and UNIX formats are available from the author upon request at vavilov@mathematik.uni-bielefeld.de.

Acknowledgements. This work was done during my stays at the *Univerità di Milano* and the *Universität Bielefeld*. *Very* special thanks should go to my hosts L.Di Martino and A.Bak for their friendly attention and support. Joint work with L.Di Martino on the (2,3)-generation of groups of Lie type was one of my main motivations. I would like to thank A.M.Cohen, I.B.Frenkel, E.B.Plotkin (joint work with whom was another major motivation) and G.M.Seitz for useful discussions. P.Gilkey and G.M.Seitz sent me at least two versions of their tables. When I came to Bielefeld I had no experience with UNIX whatsoever. Practical lessons by U.Rehmann were much more helpful than the manuals. The Dynkin diagram is typeset using the package of TEX macros '`coxdia.tex`' written by J.Bornhöft. H.Abels and A.A.Semenov carefully read the manuscript and corrected some misprints.

[1] In his book "Computers are upsetting" Donald Knuth says "Better save than sorry".

Positive roots of $\mathbf{E}_6$

10000 0	00000 1	01000 0	00100 0	00010 0
00001 0	11000 0	00100 1	01100 0	00110 0
00011 0	11100 0	01100 1	00110 1	01110 0
00111 0	11100 1	11110 0	01110 1	00111 1
01111 0	11110 1	11111 0	01210 1	01111 1
11210 1	11111 1	01211 1	12210 1	11211 1
01221 1	12211 1	11221 1	12221 1	12321 1
12321 2				

Positive roots of $\mathbf{E}_7$

100000 0	000000 1	010000 0	001000 0	000100 0
000010 0	000001 0	110000 0	001000 1	011000 0
001100 0	000110 0	000011 0	111000 0	011000 1
001100 1	011100 0	001110 0	000111 0	111000 1
111100 0	011100 1	001110 1	011110 0	001111 0
111100 1	111110 0	012100 1	011110 1	001111 1
011111 0	112100 1	111110 1	111111 0	012110 1
011111 1	122100 1	112110 1	111111 1	012210 1
012111 1	122110 1	112210 1	112111 1	012211 1
122210 1	122111 1	112211 1	012221 1	123210 1
122211 1	112221 1	123210 2	123211 1	122221 1
123211 2	123221 1	123221 2	123321 1	123321 2
124321 2	134321 2	234321 2		

Positive roots of $\mathbf{E}_8$

1000000 0	0000000 1	0100000 0	0010000 0	0001000 0
0000100 0	0000010 0	0000001 0	1100000 0	0010000 1
0110000 0	0011000 0	0001100 0	0000110 0	0000011 0
1110000 0	0110000 1	0011000 1	0111000 0	0011100 0
0001110 0	0000111 0	1110000 1	1111000 0	0111000 1
0011100 1	0111100 0	0011110 0	0001111 0	1111000 1
1111100 0	0121000 1	0111100 1	0011110 1	0111110 0
0011111 0	1121000 1	1111100 1	1111110 0	0121100 1
0111110 1	0011111 1	0111111 0	1221000 1	1121100 1
1111110 1	1111111 0	0122100 1	0121110 1	0111111 1
1221100 1	1122100 1	1121110 1	1111111 1	0122110 1
0121111 1	1222100 1	1221110 1	1122110 1	1121111 1
0122210 1	0122111 1	1232100 1	1222110 1	1221111 1
1122210 1	1122111 1	0122211 1	1232100 2	1232110 1
1222210 1	1222111 1	1122211 1	0122221 1	1232110 2
1232210 1	1232111 1	1222211 1	1122221 1	1232210 2
1232111 2	1233210 1	1232211 1	1222221 1	1233210 2
1232211 2	1233211 1	1232221 1	1243210 2	1233211 2
1232221 2	1233221 1	1343210 2	1243211 2	1233221 2
1233321 1	2343210 2	1343211 2	1243221 2	1233321 2
2343211 2	1343221 2	1243321 2	2343221 2	1343321 2
1244321 2	2343321 2	1344321 2	2344321 2	1354321 2
2354321 2	1354321 3	2354321 3	2454321 2	2454321 3
2464321 3	2465321 3	2465421 3	2465431 3	2465432 3

Structure constants for $\mathbf{E}_6$

	100 0 0 010 0 0 001 0 0 000 1 0 000 0 1 000 0 0	0 1 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 1 1 1 0 0 0 0 1 1 0 0 0 0	0 1 0 0 0 0 0 1 1 0 0 1 1 0 1 0 1 1 1 1 1 0 0 1 1 1 0 0 0 0	0 1 1 0 0 0 1 0 1 1 0 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 0 0 0 1	0 1 1 0 0 0 1 0 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 0 1 0 1	1 1 0 1 1 1 1 1 1 1 1 1 1 2 1 2 1 2 2 2 1 1 1 1 1 0 1 1 0 1	0 1 1 1 1 1 1 1 1 1 1 2 1 2 2 2 2 2 2 3 2 1 2 2 2 1 1 1 1 1	1 2 2 3 2 1
100000	0 0 + 0 0	0 0 0 + 0	0 0 + 0 +	0 0 0 + 0	+ 0 0 + +	0 0 + 0 0	+ 0 0 0 0	0
010000	0 0 0 + 0	0 0 0 + +	0 + 0 0 +	+ 0 + 0 0	+ 0 + 0 0	0 0 0 0 0	0 0 0 0 +	0
001000	− 0 0 + 0	0 0 + 0 +	0 0 0 + 0	+ 0 0 0 +	0 0 0 0 0	+ 0 0 0 +	0 0 + 0 0	0
000100	0 − − 0 +	0 − 0 0 0	+ 0 0 0 0	0 0 0 + 0	0 + 0 0 +	0 + 0 0 0	0 0 0 + 0	0
000010	0 0 0 − 0	+ 0 − − 0	0 − − 0 0	0 − 0 0 0	0 0 0 0 0	0 0 + 0 +	0 + 0 0 0	0
000001	0 0 0 0 −	0 0 0 0 −	0 0 0 − −	0 0 − − 0	0 − 0 − 0	− 0 0 − 0	0 0 0 0 0	0
101000	0 0 0 + 0	0 0 + 0 +	0 0 0 + 0	+ 0 0 0 +	0 0 0 − 0	0 0 − 0 0	− 0 0 0 0	0
010100	0 0 − 0 +	0 − 0 0 0	+ 0 0 0 −	0 0 − 0 0	− 0 − 0 0	0 0 0 0 0	0 0 0 + 0	0
001100	− − 0 0 +	0 0 0 0 0	+ 0 0 − 0	0 0 0 0 −	0 + 0 0 0	0 + 0 0 0	0 0 − 0 0	0
000110	0 − − 0 0	+ − 0 0 0	0 0 − 0 0	0 − 0 0 0	0 0 0 0 −	0 − 0 0 0	0 + 0 0 0	0
000011	0 0 0 − 0	0 0 − − 0	0 − − 0 0	0 − 0 0 0	0 0 0 − 0	− 0 0 − 0	0 0 0 0 0	0
101100	0 − 0 0 +	0 0 0 0 0	+ 0 0 − 0	0 0 0 − −	0 0 0 0 −	0 0 0 0 0	+ 0 0 0 0	0
011100	− 0 0 0 +	0 0 0 0 +	+ 0 0 0 0	+ 0 − 0 0	0 0 − 0 0	0 0 0 0 0	0 0 − 0 0	0
010110	0 0 − 0 0	+ − 0 + 0	0 + 0 0 0	0 0 0 0 0	+ 0 + 0 0	0 0 0 0 0	0 + 0 0 0	0
001110	− − 0 0 0	+ 0 + 0 0	0 0 0 0 0	0 − 0 0 +	0 0 0 0 0	0 − 0 0 −	0 0 0 0 0	0
000111	0 − − 0 0	0 − 0 0 0	0 0 − 0 0	0 − 0 + 0	0 + 0 0 0	0 0 0 − 0	0 0 0 0 0	0
111100	0 0 0 0 +	0 0 0 0 +	+ 0 0 0 +	+ 0 0 0 0	+ 0 0 0 0	0 0 0 0 0	+ 0 0 0 0	0
101110	0 − 0 0 0	+ 0 + 0 0	0 0 + 0 0	0 0 0 0 +	0 0 0 0 +	0 0 + 0 0	0 0 0 0 0	0
011110	− 0 0 − 0	+ 0 0 0 0	0 + 0 0 0	− 0 0 0 0	0 0 + 0 0	0 0 0 0 −	0 0 0 0 0	0
010111	0 0 − 0 0	0 − 0 + 0	0 + 0 0 −	0 0 − 0 0	0 0 0 0 0	0 0 0 − 0	0 0 0 0 0	0
001111	− − 0 0 0	0 0 + 0 0	0 0 0 − 0	0 − 0 0 0	0 + 0 0 0	+ 0 0 0 0	0 0 0 0 0	0
111110	0 0 0 − 0	+ 0 0 − 0	0 0 0 0 0	− 0 0 0 0	− 0 0 0 0	0 0 + 0 0	0 0 0 0 0	0
101111	0 − 0 0 0	0 0 + 0 0	0 0 + − 0	0 0 0 − 0	0 0 0 − 0	0 0 0 0 0	0 0 0 0 0	0
011210	− 0 0 0 0	+ + 0 0 0	+ 0 0 0 0	0 0 0 0 0	0 0 + 0 0	0 + 0 0 0	0 0 0 0 0	0
011111	− 0 0 − 0	0 0 0 0 +	0 + 0 0 0	0 0 − 0 0	0 0 0 0 0	+ 0 0 0 0	0 0 0 0 0	0
111210	0 0 − 0 0	+ 0 0 0 0	+ 0 0 0 0	0 0 0 0 0	− 0 0 0 −	0 0 0 0 0	0 0 0 0 0	0
111111	0 0 0 − 0	0 0 0 − +	0 0 0 0 +	0 0 0 0 0	0 0 0 − 0	0 0 0 0 0	0 0 0 0 0	0
011211	− 0 0 0 −	0 + 0 0 0	0 0 0 0 0	0 0 − 0 0	0 − 0 0 0	0 0 0 0 0	0 0 0 0 0	0
112210	0 0 0 0 0	+ 0 0 0 0	+ 0 0 0 0	+ 0 0 0 +	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0
111211	0 0 − 0 −	0 0 0 0 0	0 0 0 0 +	0 0 0 + 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0
011221	− 0 0 0 0	0 + 0 0 0	0 − 0 0 0	0 − 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0
112211	0 0 0 0 −	0 0 0 0 −	0 0 0 − 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0
111221	0 0 − 0 0	0 0 0 + 0	0 0 + 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0
112221	0 0 0 − 0	0 0 − 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0
112321	0 − 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0
122321	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0

Structure constants for E_7

	1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0	0 0 1 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0	0 0 0 1 0 0 0 0 0 1 0 0 0 1 1 1 0 0 1 1 1 1 0 0 0 0 1 1 0 0 0 0 1 0 0	0 0 0 0 1 1 0 0 0 1 0 1 0 0 1 1 1 1 0 1 1 1 1 1 0 0 0 1 1 0 0 0 0 1 0	1 0 0 0 0 0 1 1 0 0 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 0 0 0 0 1	1 1 0 0 0 1 0 1 1 1 1 1 1 1 0 1 1 2 1 1 1 1 1 1 1 0 1 0 1 1 0 0 0 0 1	0 1 0 1 1 1 1 2 1 1 1 0 1 0
1000000	0 0 + 0 0	0 0 0 0 +	0 0 0 0 +	0 + 0 0 0	0 + 0 + 0	0 0 + + 0	+ 0
0100000	0 0 0 + 0	0 0 0 0 +	+ 0 0 + 0	0 + + 0 0	+ 0 0 + +	0 + 0 0 0	+ 0
0010000	− 0 0 + 0	0 0 0 + 0	+ 0 0 0 0	+ 0 + 0 0	0 0 + 0 +	0 0 0 0 +	0 +
0001000	0 − − 0 +	0 0 − 0 0	0 + 0 0 0	0 0 0 + 0	0 + 0 0 0	+ 0 0 + 0	0 0
0000100	0 0 0 − 0	+ 0 0 − −	0 0 + − −	0 0 0 0 −	0 0 0 0 0	0 0 0 0 0	0 0
0000010	0 0 0 0 −	0 + 0 0 0	− 0 0 0 0	− − 0 0 0	− − 0 0 0	− 0 − 0 0	0 −
0000001	0 0 0 0 0	− 0 0 0 0	0 − 0 0 0	0 0 − 0 0	0 0 − − 0	0 − 0 − 0	0 0
1010000	0 0 0 + 0	0 0 0 + 0	+ 0 0 0 0	+ 0 + 0 0	0 0 + 0 +	0 0 − 0 +	0 0
0101000	0 0 − 0 +	0 0 − 0 0	0 + 0 0 0	0 − 0 + 0	− 0 0 − 0	0 − 0 0 0	− 0
0011000	− − 0 0 +	0 0 0 0 0	0 + 0 0 0	− 0 0 + 0	0 0 − 0 0	+ 0 0 0 −	0 0
0001100	0 − − 0 0	+ 0 − 0 0	0 0 + 0 −	0 0 0 0 −	0 0 0 0 0	0 0 0 − 0	0 0
0000110	0 0 0 − 0	0 + 0 − −	0 0 0 − −	0 0 0 0 −	0 0 0 0 0	0 0 − 0 0	0 −
0000011	0 0 0 0 −	0 0 0 0 0	− 0 0 0 0	− − 0 0 0	− − 0 0 0	− 0 − 0 0	0 −
1011000	0 − 0 0 +	0 0 0 0 0	0 + 0 0 0	− 0 0 + 0	0 − − 0 0	0 0 0 − −	0 0
0111000	− 0 0 0 +	0 0 0 0 0	+ + 0 0 0	0 0 + + 0	− 0 0 0 +	0 − 0 0 0	0 0
0101100	0 0 − 0 0	+ 0 − 0 +	0 0 + + 0	0 0 0 0 0	0 0 0 + 0	0 + 0 0 0	+ 0
0011100	− − 0 0 0	+ 0 0 + 0	0 0 + 0 0	0 0 0 0 −	0 0 + 0 0	0 0 0 0 +	0 0
0001110	0 − − 0 0	0 + − 0 0	0 0 0 0 −	0 0 0 0 −	0 + 0 0 0	+ 0 0 0 0	0 0
0000111	0 0 0 − 0	0 0 0 − −	0 0 0 − −	0 0 0 0 −	0 0 0 0 0	0 0 − 0 0	0 −
1111000	0 0 0 0 +	0 0 0 0 0	+ + 0 0 0	0 + + + 0	0 0 0 + +	0 0 0 0 0	+ 0
1011100	0 − 0 0 0	+ 0 0 + 0	0 0 + 0 +	0 0 0 0 0	0 0 + 0 0	0 0 0 + +	0 0
0111100	− 0 0 − 0	+ 0 0 0 0	0 0 + + 0	0 0 − 0 0	0 0 0 0 −	0 + 0 0 0	0 0
0101110	0 0 − 0 0	0 + − 0 +	0 0 0 + 0	0 − 0 0 0	− 0 0 0 0	0 0 0 0 0	− 0
0011110	− − 0 0 0	0 + 0 + 0	0 0 0 0 0	− 0 0 0 −	0 0 0 0 0	+ 0 0 0 −	0 +
0001111	0 − − 0 0	0 0 − 0 0	0 0 0 0 −	0 0 0 0 −	0 + 0 0 0	+ 0 0 − 0	0 0
1111100	0 0 0 − 0	+ 0 0 0 −	0 0 + 0 0	0 0 − 0 0	0 0 0 − −	0 0 0 0 0	− 0
1011110	0 − 0 0 0	0 + 0 + 0	0 0 0 0 +	− 0 0 0 0	0 − 0 0 0	0 0 − 0 −	0 0
0112100	− 0 0 0 0	+ 0 + 0 0	0 + + 0 0	0 0 0 + 0	0 0 0 0 0	0 + 0 0 0	0 0
0111110	− 0 0 − 0	0 + 0 0 0	+ 0 0 + 0	0 0 0 0 0	− 0 0 0 +	0 0 0 0 0	0 +
0101111	0 0 − 0 0	0 0 − 0 +	0 0 0 + 0	0 − 0 0 0	− 0 0 + 0	0 + 0 0 0	0 0
0011111	− − 0 0 0	0 0 0 + 0	0 0 0 0 0	− 0 0 0 −	0 0 + 0 0	+ 0 0 0 0	0 +
1112100	0 0 − 0 0	+ 0 0 0 0	0 + + 0 0	0 0 0 + 0	0 0 0 − 0	0 0 0 − 0	− 0

	1	1	0	0	1	1	1	0	0	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	2
	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	2	1	2	1	2	2	2	2
	1	1	1	1	2	1	1	1	1	2	1	1	1	2	2	1	1	2	2	1	2	2	2	2	2	2	2	2	2	3	3
	1	1	2	1	2	2	1	2	2	2	2	2	2	2	2	2	2	3	2	2	3	3	2	3	3	3	3	3	4	4	4
	1	1	1	1	1	1	1	2	1	1	2	1	2	2	1	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3
	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	2	1	1	2	1	1	2	1	2	2	2	2	2	2	2
	0	1	0	1	0	0	1	0	1	0	0	1	1	0	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1
1000000	0	0	+	+	0	0	0	+	+	0	0	0	+	0	0	0	+	0	0	0	0	0	0	0	0	0	0	0	0	+	0
0100000	0	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	+	0	0	0	+	0	0	+	0	+	0	0	0	0
0010000	0	0	0	0	0	+	0	0	0	0	+	+	0	0	0	+	0	0	0	+	0	0	0	0	0	0	0	0	+	0	0
0001000	+	0	0	+	0	0	+	0	0	0	0	0	0	+	0	0	0	0	+	0	0	0	+	0	0	0	0	+	0	0	0
0000100	0	0	+	0	0	+	0	0	+	+	0	+	0	0	+	0	0	0	0	0	0	0	0	0	+	+	0	0	0	0	0
0000010	0	0	0	0	−	0	0	0	0	0	0	0	+	0	0	+	0	0	+	0	0	+	0	+	0	0	0	0	0	0	0
0000001	−	0	−	0	0	−	0	−	0	−	−	0	0	−	0	0	0	−	0	0	−	0	0	0	0	0	0	0	0	0	0
1010000	0	0	−	0	0	0	0	−	−	0	0	0	−	0	0	0	−	0	0	0	0	0	0	0	0	0	0	0	+	0	0
0101000	0	−	0	0	0	0	0	0	0	0	0	0	0	+	0	0	0	0	+	0	0	0	+	0	0	0	−	0	0	0	0
0011000	+	0	0	0	0	0	+	0	0	0	−	0	0	0	0	−	0	0	0	−	0	0	0	0	0	0	0	+	0	0	0
0001100	−	0	0	−	0	0	−	0	0	+	0	0	0	0	+	0	0	0	0	0	0	0	−	0	0	+	0	0	0	0	0
0000110	0	0	0	0	−	0	0	0	−	0	0	−	0	0	−	0	0	0	0	0	0	+	0	+	0	0	0	0	0	0	0
0000011	0	0	0	0	−	0	0	−	0	0	−	0	0	−	0	0	0	−	0	0	−	0	0	0	0	0	0	0	0	0	0
1011000	0	0	0	−	0	0	0	+	0	0	0	0	+	0	0	0	+	0	0	0	0	0	0	0	0	0	0	+	0	0	0
0111000	0	-1	0	0	0	0	0	0	0	0	−	0	0	0	0	−	0	0	0	−	0	0	0	0	0	0	−	0	0	0	0
0101100	0	+	0	0	0	0	0	0	0	+	0	0	0	0	+	0	0	0	0	0	0	0	−	0	−	0	0	0	0	0	0
0011100	−	0	0	0	0	−	−	0	0	0	0	−	0	0	0	0	0	0	0	+	0	0	0	0	0	+	0	0	0	0	0
0001110	0	0	0	+	−	0	+	0	0	0	0	0	0	0	−	0	0	0	−	0	0	0	0	+	0	0	0	0	0	0	0
0000111	0	0	+	0	−	+	0	0	0	+	0	0	0	0	0	0	0	−	0	0	−	0	0	0	0	0	0	0	0	0	0
1111000	0	0	0	0	0	0	0	+	0	0	0	0	+	0	0	0	+	0	0	0	0	0	0	0	0	0	−	0	0	0	0
1011100	0	0	+	+	0	0	0	0	+	0	0	0	0	0	0	0	−	0	0	0	0	0	0	0	0	+	0	0	0	0	0
0111100	0	+	0	0	0	−	0	0	0	0	0	−	0	0	0	0	0	0	0	+	0	0	0	0	−	0	0	0	0	0	0
0101110	0	−	0	0	−	0	0	0	0	0	0	0	0	0	−	0	0	0	−	0	0	−	0	0	0	0	0	0	0	0	0
0011110	0	0	0	0	0	0	+	0	0	0	0	+	0	0	0	+	0	0	0	0	0	0	0	+	0	0	0	0	0	0	0
0001111	−	0	0	0	−	0	0	0	0	+	0	0	0	+	0	0	0	0	0	0	−	0	0	0	0	0	0	0	0	0	0
1111100	0	0	+	0	0	0	0	0	+	0	0	0	0	0	0	0	−	0	0	0	0	0	0	0	−	0	0	0	0	0	0
1011110	0	0	0	−	0	0	0	0	−	0	0	0	−	0	0	0	0	0	0	0	0	0	0	+	0	0	0	0	0	0	0
0112100	+	+	0	0	0	0	+	0	0	0	0	0	0	0	0	0	0	0	0	+	0	0	+	0	0	0	0	0	0	0	0
0111110	0	−	0	0	0	0	0	0	0	0	0	+	0	0	0	+	0	0	0	0	0	−	0	0	0	0	0	0	0	0	0
0101111	0	0	0	0	−	0	0	0	0	+	0	0	0	+	0	0	0	+	0	0	0	0	0	0	0	0	0	0	0	0	0
0011111	−	0	0	0	0	−	0	0	0	0	−	0	0	0	0	0	0	0	0	0	−	0	0	0	0	0	0	0	0	0	0
1112100	0	0	0	−	0	0	0	0	0	0	0	0	0	0	0	0	−	0	0	0	0	0	+	0	0	0	0	0	0	0	0

	10000 01000 00100 00010 00001 00000 00000	0 0 1 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0	0 0 0 1 0 0 0 0 0 1 0 0 0 1 1 1 0 0 1 1 1 1 0 0 0 0 1 1 0 0 0 0 1 0 0	0 0 0 0 1 1 0 0 0 1 0 1 0 0 1 1 1 1 0 1 1 1 1 1 0 0 0 1 1 0 0 0 0 1 0	1 0 0 00 0 1 1 00 1 1 0 10 1 1 1 11 1 1 1 11 0 0 1 11 0 0 0 01	1 1 00 0 1 0 11 1 1 1 11 0 1 1 21 1 1 1 11 1 0 1 01 1 0 0 00 1	0 1 0 1 1 1 1 2 1 1 1 0 1 0
1111110	000-0	0 +0 0 -	+0 0 0 0	0 +0 0 0	0 0 0 0+	0 0 -0 0	+0
1011111	0-000	0 0 0 +0	0 0 0 0 +	- 0 0 0 0	0 - +00	0 0 - +0	0 0
0112110	-000-	0 + +0 0	0 0 0 0 0	0 0 0 - 0	- 0 0 00	- 0 00 0	0 0
0111111	-00-0	0 0 0 0 0	+0 0 +0	0 0 - 0 0	- 0 0 00	0 +00 0	0 1
1122100	00000	+0 0 0 0	0 + +0 0	0 0 + +0	0 0 +0+	0 0 00 +	0 0
1112110	00-0-	0 +0 0 0	0 0 0 0 0	0 +0 - 0	0 +0 00	0 0 00 0	+0
1111111	000-0	0 0 0 0 -	+0 0 0 0	0 +- 0 0	0 0 0 -0	0 0 -0 0	0 0
0112210	-0000	0 + +0 0	0 0 +- 0	0 0 0 0 -	0 0 0 00	0 0 00 0	0 0
0112111	-000-	0 0 +0 0	0 +0 0 0	0 0 0 0 0	- 0 0 00	- +00 0	0 0
1122110	0000-	0 +0 0 0	- 0 0 0 0	- 0 0 - 0	0 0 0 0-	0 0 00 -	0 0
1112210	00-00	0 +0 0 +	0 0 +0 +	0 0 0 0 0	0 0 0 00	0 0 00 0	+0
1112111	00-0-	0 0 0 0 0	0 +0 0 0	0 +0 0 0	0 +0 -0	0 0 0- 0	0 0
0112211	-0000	- 0 +0 0	0 0 0 - 0	0 0 0 0 -	0 0 0 00	0 +00 0	0 0
1122210	000-0	0 +0 - 0	0 0 +0 0	0 0 0 0 0	0 0 0 0-	0 0 00 -	0 0
1122111	0000-	0 0 0 0 0	- +0 0 0	- 0 +0 0	0 0 +00	0 0 00 0	0 0
1112211	00-00	- 0 0 0 +	0 0 0 0 +	0 0 0 0 0	0 0 0 -0	0 0 0- 0	0 0
0112221	-0000	0 0 +0 0	0 0 0 - 0	0 0 0 0 -	+0 0 00	+0 00 0	0 +
1123210	0-000	0 +0 0 0	0 0 +0 0	0 0 0 +0	0 0 0 00	0 0 00 -	0 0
1122211	000-0	- 0 0 - 0	0 0 0 0 0	0 0 +0 0	0 0 +00	0 0 00 0	0 0
1112221	00-00	0 0 0 0 +	0 0 0 0 +	0 - 0 0 0	0 - 0 00	0 0 -0 0	0 0
1223210	00000	0 +0 0 0	0 0 +0 0	0 0 0 +0	0 0 0 0+	0 0 00 0	+0
1123211	0-000	- 0 0 0 0	0 - 0 0 0	0 0 0 0 0	0 0 +00	0 0 0+0	0 0
1122221	000-0	0 0 0 - 0	+0 0 0 0	+0 0 0 0	0 0 0 00	0 0 -0 0	0 -
1223211	00000	- 0 0 0 0	0 - 0 0 0	0 0 - 0 0	0 0 0 -0	0 - 00 0	0 0
1123221	0-00-	0 0 0 0 0	0 0 0 0 0	+0 0 0 0	0 +0 00	+0 00 0	0 0
1223221	0000-	0 0 0 0 0	- 0 0 0 0	0 - 0 0 0	- 0 0 00	0 0 00 0	0 0
1123321	0-000	0 0 0 +0	0 0 0 0 +	0 0 0 0 +	0 0 0 00	0 0 00 0	0 0
1223321	000-0	0 0 0 0 -	0 0 0 - 0	0 0 0 0 0	0 0 0 00	0 0 00 0	0 0
1224321	00-00	0 0 - 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 00	0 0 00 0	0 0
1234321	-0000	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 00	0 0 00 0	0 0
2234321	00000	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 00	0 0 00 0	0 0

	1111110	1011111	0112110	0111111	1122100	1112110	1111111	0112210	0112111	1122110	1112210	1112111	0112211	1122210	1122111	1112211	0112221	1123210	1122211	1112221	1223210	1123211	1122221	1223211	1123221	1223221	1123321	1223321	1224321	1234321	2234321
1111110	0	0	0	0	0	0	0	0	−	0	0	0	−	0	0	0	0	0	0	0	0	−	0	0	0	0	0	0	0	0	0
1011111	0	0	+	0	0	0	0	+	0	0	0	0	0	0	0	0	0	0	0	0	−	0	0	0	0	0	0	0	0	0	0
0112110	0	−	0	0	0	0	−	0	0	0	0	0	0	0	0	+	0	0	+	0	0	0	0	0	0	0	0	0	0	0	0
0111111	0	0	0	0	0	−	0	0	0	0	−	0	0	0	0	0	0	+	0	0	0	0	0	0	0	0	0	0	0	0	0
1122100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	−	0	0	−	0	0	0	0	0	0	0	0	0	0	0
1112110	0	0	0	+	0	0	0	0	0	0	0	0	−	0	0	0	0	0	+	0	0	0	0	0	0	0	0	0	0	0	0
1111111	0	0	+	0	0	0	0	+	0	0	0	0	0	0	0	0	0	+	0	0	0	0	0	0	0	0	0	0	0	0	0
0112210	0	−	0	0	0	0	−	0	0	0	0	−	0	0	−	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0112111	+	0	0	0	0	0	0	0	0	0	−	0	0	−	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1122110	0	0	0	0	0	0	0	0	0	0	0	0	−	0	0	−	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1112210	0	0	0	+	0	0	0	0	+	0	0	0	0	0	−	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1112111	0	0	0	0	0	0	0	+	0	0	0	0	0	−	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0112211	+	0	0	0	0	+	0	0	0	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1122210	0	0	0	0	0	0	0	0	+	0	0	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1122111	0	0	0	0	0	0	0	+	0	0	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1112211	0	0	−	0	0	0	0	0	0	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0112221	0	0	0	0	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1123210	0	0	0	−	0	0	−	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1122211	0	0	−	0	0	−	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1112221	0	0	0	0	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1223210	0	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1123211	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1122221	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1223211	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1123221	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1223221	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1123321	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1223321	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1224321	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1234321	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2234321	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Structure constants for $\mathbf{E}_8$

	10000	00010	00000	10000	00110	00001
	01000	00001	00000	01100	00101	10001
	00100	00010	10000	11010	00111	01001
	00010	00001	11000	11111	00111	11101
	00001	00000	01100	00111	10011	11111
	00000	10000	00110	00001	11000	11110
	00000	01000	00011	00000	11000	00110
	00000	00100	00001	00000	01000	00010
10000000	00+00	00000	+0000	0+0+0	0000+	0+000
01000000	000+0	00000	++000	+00++	000+0	0++00
00100000	-00+0	0000+	0+000	00+0+	00000	+0+00
00010000	0--0+	000-0	00+00	00000	+000+	000++
00001000	000-0	+000-	-00+0	--000	0+-00	00000
00000100	0000-	0+000	0-00+	00--0	000--	0000-
00000010	00000	-0+00	00-00	0000-	00000	--000
00000001	00000	0-000	000-0	00000	-0000	00-00
10100000	000+0	0000+	0+000	00+0+	00000	+0+00
01010000	00-0+	000-0	00+00	000-0	+00-0	0-0+0
00110000	--00+	00000	00+00	00-00	+0000	-00++
00011000	0--00	+00-0	000+0	0-000	0+-00	00000
00001100	000-0	0+00-	-000+	--000	00-00	00000
00000110	0000-	00+00	0-000	00--0	000--	0000-
00000011	00000	-0000	00-00	0000-	00000	--000
10110000	0-00+	00000	00+00	00-00	+000-	-00+0
01110000	-000+	00000	0++00	0000+	+00-0	00++0
01011000	00-00	+00-0	+00+0	+0000	0+000	0+000
00111000	--000	+000+	000+0	00000	0+-00	+0000
00011100	0--00	0+0-0	0000+	0-000	00-0+	0000+
00001110	000-0	00+0-	-0000	--000	00-00	00000
00000111	0000-	00000	0-000	00--0	000--	0000-
11110000	0000+	00000	0++00	000++	+0000	0+++0
10111000	0-000	+000+	000+0	0+000	0+000	+0000
01111000	-00-0	+0000	000+0	+000-	0+000	00-00
01011100	00-00	0+0-0	+000+	+00-0	000-0	00000
00111100	--000	0+00+	0000+	00-00	00-00	0000+
00011110	0--00	00+-0	00000	0-000	00-0+	0000+
00001111	000-0	0000-	-0000	--000	00-00	00000
11111000	000-0	+0000	-00+0	0000-	0+000	0--00

	1 0 0 0 0 0 1 1 1 0 1 1 1 0 1 1 2 1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 0 0 0	0 1 1 1 0 0 1 1 0 1 0 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 0 1 1 1 1 0 0 1 0 1 0 0 0 0	0 0 0 1 1 1 1 0 1 1 1 0 1 2 1 1 1 1 2 2 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 0 1 1 0 0	1 1 0 0 0 1 0 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 2 1 1 1 1 1 1 1 1 1 0 1 1 0 1 0 0 1	1 1 1 1 0 1 1 1 1 1 2 1 1 1 1 2 2 2 1 2 1 2 1 1 2 1 1 1 1 1 0 0 1 1 1 0 0 0 1 0	0 1 1 1 1 1 1 1 1 1 1 2 2 1 1 2 2 2 2 2 1 2 1 2 1 1 1 1 1 1 1 0 1 1 1 1 0 0 0 1
10000000	0 + + 0 +	0 0 0 0 +	+ 0 + 0 0	0 0 + + +	0 0 0 0 +	+ 0 0 0 0
01000000	+ 0 0 0 +	+ 0 0 + 0	0 0 + 0 0	0 + 0 0 0	0 0 0 0 0	0 0 0 0 0
00100000	0 0 0 + 0	+ + 0 0 0	0 + 0 0 +	0 0 0 0 0	0 + + 0 0	0 0 0 + +
00010000	0 0 + 0 0	0 0 + 0 0	+ 0 0 0 0	+ 0 0 0 +	0 0 0 + 0	0 + 0 0 0
00001000	0 0 0 0 0	0 0 0 0 +	0 0 0 0 +	0 0 0 + 0	+ 0 + 0 0	+ 0 + 0 +
00000100	0 − 0 0 0	0 − 0 0 0	0 0 0 − 0	0 0 0 0 0	0 0 0 0 +	0 0 0 + 0
00000010	− 0 − 0 0	0 0 − 0 −	0 0 0 0 −	0 0 − 0 0	− − 0 0 0	0 − 0 0 0
00000001	0 0 0 − −	0 0 0 − 0	− 0 0 0 0	− 0 0 − 0	0 0 − 0 −	0 0 − − 0
10100000	0 − 0 + 0	+ 0 0 0 −	0 + 0 0 0	0 0 − − 0	0 0 0 0 −	− 0 0 0 0
01010000	− 0 0 0 −	0 0 0 − 0	0 0 − 0 0	0 − 0 0 0	0 0 0 0 0	0 + 0 0 0
00110000	0 0 0 − 0	0 0 + 0 0	0 − 0 0 0	+ 0 0 0 0	0 − 0 + 0	0 0 0 − 0
00011000	0 0 − 0 0	0 0 − 0 0	− 0 0 0 0	− 0 0 0 −	+ 0 0 − 0	0 0 + 0 0
00001100	0 − 0 0 0	0 − 0 0 0	0 0 0 − 0	0 0 0 − 0	0 0 − 0 0	− 0 − 0 −
00000110	0 − 0 0 0	0 − 0 0 0	0 0 0 − 0	0 0 − 0 0	0 − 0 0 0	0 − 0 0 0
00000011	− 0 − 0 0	0 0 − 0 −	0 0 0 0 −	0 0 − 0 0	− − 0 0 0	0 − 0 0 0
10110000	0 0 − − 0	0 0 0 0 0	− − 0 0 0	0 0 + 0 −	0 0 0 0 +	0 0 0 0 0
01110000	− 0 0 0 0	+ 0 0 − 0	0 0 0 0 0	0 − 0 0 0	0 − 0 0 0	0 0 0 − 0
01011000	+ 0 0 0 +	0 0 0 + 0	0 0 + 0 0	0 + 0 0 0	+ 0 0 0 0	0 0 + 0 0
00111000	0 0 0 + 0	0 0 − 0 0	0 + 0 0 −	− 0 0 0 0	0 0 − − 0	0 0 0 0 −
00011100	0 0 0 0 0	0 0 0 0 0	+ 0 0 − 0	+ 0 0 0 +	0 0 0 + 0	0 0 − 0 0
00001110	0 − 0 0 0	0 − 0 0 +	0 0 0 − +	0 0 0 0 0	+ 0 0 0 0	+ 0 0 0 +
00000111	0 − 0 0 0	0 − 0 0 0	0 0 0 − 0	0 0 − 0 0	0 − 0 0 +	0 − 0 + 0
11110000	0 0 0 0 +	+ 0 0 0 0	0 0 + 0 0	0 0 + 0 0	0 0 0 0 +	0 0 0 0 0
10111000	0 0 + + 0	0 0 0 0 +	+ + 0 0 0	0 0 0 + +	0 0 0 0 0	+ 0 0 0 0
01111000	+ 0 0 0 0	− 0 0 + 0	0 0 0 0 −	0 + 0 0 0	0 0 − 0 0	0 0 0 0 −
01011100	0 0 0 0 −	0 0 0 − 0	0 0 − − 0	0 − 0 0 0	0 0 0 0 0	0 0 − 0 0
00111100	0 0 0 − 0	0 + 0 0 0	0 − 0 0 0	+ 0 0 0 0	0 0 + + 0	0 0 0 + +
00011110	0 0 − 0 0	0 0 − 0 0	0 0 0 − 0	0 0 0 0 −	+ 0 0 − 0	0 + 0 0 0
00001111	0 − 0 0 0	0 − 0 0 +	0 0 0 − +	0 0 0 − 0	+ 0 − 0 0	0 0 − 0 0
11111000	0 0 0 0 −	− 0 0 0 +	0 0 − 0 0	0 0 0 + 0	0 0 0 0 0	+ 0 0 0 0

	0 0 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 3 2 2 2 2 2 2 1 2 1 1 1 1 1 1 0 1 1 0 1 0 0 1	1 1 0 1 1 1 1 1 2 1 1 1 1 2 2 2 2 2 3 3 2 2 2 2 2 2 1 2 1 1 1 1 1 0 1 0 1 1 0 0	1 1 1 0 1 1 1 1 1 2 2 2 1 1 2 2 2 2 2 3 2 2 2 2 2 2 1 2 2 1 1 1 1 2 1 0 1 1 1 0	1 1 1 1 1 1 1 1 1 2 2 2 2 1 2 3 3 2 2 3 2 2 2 2 2 2 1 2 2 2 1 1 1 2 1 0 1 1 1 0	1 1 1 1 1 2 1 1 1 2 2 2 2 2 2 3 3 3 2 3 2 3 2 2 3 1 2 2 2 2 1 1 1 2 1 1 0 1 1 0	1 1 1 1 1 2 1 1 2 2 2 2 2 2 2 3 3 3 4 3 2 3 2 3 3 2 2 2 2 2 1 1 2 1 1 1 1 1 0 1
10000000	+ + 0 0 0	0 0 + 0 0	0 0 0 + 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
01000000	0 0 + 0 0	0 0 0 0 +	0 0 0 0 0	+ + 0 0 0	0 + + 0 0	0 + + 0 0
00100000	0 0 0 0 0	+ + 0 0 0	0 0 + 0 0	0 0 0 + 0	0 0 0 0 0	0 0 0 + 0
00010000	0 0 0 + 0	0 0 0 0 0	+ + 0 0 0	0 0 + 0 0	0 0 0 + +	0 0 0 0 +
00001000	0 0 0 0 +	0 0 0 0 0	0 0 0 0 0	+ 0 0 0 +	0 0 + 0 0	+ 0 + 0 0
00000100	0 + 0 + 0	0 + 0 0 +	0 + 0 0 +	0 + 0 0 0	+ 0 0 0 0	0 0 0 0 0
00000010	0 0 − 0 0	0 0 + − 0	0 0 + 0 0	0 0 + 0 0	0 0 + 0 0	+ + 0 0 +
00000001	− 0 0 − 0	− 0 0 0 −	− 0 0 0 −	− 0 0 0 −	0 − 0 0 −	0 0 0 − 0
10100000	− − 0 0 0	0 0 − 0 0	0 0 0 − 0	0 0 0 0 0	0 0 0 0 0	0 0 0 + 0
01010000	0 0 0 + 0	0 0 0 0 0	+ + 0 0 0	0 0 + 0 0	0 − 0 + 0	0 − 0 0 0
00110000	0 0 0 0 0	− − 0 0 0	0 0 − 0 0	0 0 0 − 0	0 0 0 0 +	0 0 0 0 +
00011000	0 0 0 0 +	0 0 0 0 0	− 0 0 0 0	0 0 − 0 +	0 0 0 − 0	+ 0 0 0 0
00001100	0 0 0 0 −	0 0 0 0 +	0 0 0 0 +	0 + 0 0 0	+ 0 0 0 0	0 0 − 0 0
00000110	0 − − 0 0	0 − 0 − 0	0 − 0 0 0	0 − 0 0 0	− 0 0 0 0	0 + 0 0 +
00000011	− 0 − 0 0	− 0 0 − 0	− 0 0 0 0	− 0 0 0 −	0 − 0 0 −	0 0 0 − 0
10110000	+ + 0 0 0	0 0 + 0 0	0 0 0 + 0	0 0 0 0 0	0 0 0 0 +	0 0 0 0 +
01110000	0 0 0 0 0	− − 0 0 0	0 0 − 0 0	0 0 0 − 0	0 − 0 0 0	0 − 0 0 0
01011000	0 0 0 0 +	0 0 0 0 0	− 0 0 0 0	− 0 − 0 0	0 0 − − 0	0 0 − 0 0
00111000	0 0 0 0 0	+ 0 0 0 0	0 0 + 0 0	0 0 0 + +	0 0 0 0 0	+ 0 0 0 0
00011100	0 0 0 − −	0 0 0 0 0	0 − 0 0 +	0 0 0 0 0	+ 0 0 + 0	0 0 0 0 0
00001110	0 0 − 0 +	0 0 0 − 0	0 0 0 0 0	0 − 0 0 0	− 0 − 0 0	− 0 0 0 0
00000111	0 0 − + 0	0 0 0 − +	0 0 0 0 +	0 0 0 0 0	0 − 0 0 −	0 0 0 − 0
11110000	+ + 0 0 0	0 0 + 0 0	0 0 0 + 0	0 0 0 0 0	0 − 0 0 0	0 − 0 0 0
10111000	− 0 0 0 0	0 0 − 0 0	0 0 0 − 0	0 0 0 0 +	0 0 0 0 0	+ 0 0 0 0
01111000	0 0 0 0 0	+ 0 0 0 0	0 0 + 0 0	− 0 0 + 0	0 0 − 0 0	0 0 − 0 0
01011100	0 0 0 − −	0 0 0 0 −	0 − 0 0 0	0 − 0 0 0	0 0 0 + 0	0 0 + 0 0
00111100	0 0 0 0 0	0 + 0 0 0	0 0 0 0 +	0 0 0 − 0	+ 0 0 0 0	0 0 0 0 0
00011110	0 0 0 0 +	0 0 0 − 0	0 + 0 0 0	0 0 + 0 0	− 0 0 0 0	− 0 0 0 −
00001111	0 0 − 0 0	0 0 0 − +	0 0 0 0 +	+ 0 0 0 +	0 0 0 0 0	0 0 0 − 0
11111000	− 0 0 0 0	0 0 − 0 0	0 0 0 − 0	− 0 0 0 0	0 0 − 0 0	0 0 − 0 0

	1 1 1 1 1	1 2 1 1 1	2 1 1 2 1	1 2 1 2 1	2 1 2 2 2	2 2 2 2 2
	2 1 2 2 2	1 2 2 2 2	2 2 2 2 2	2 2 2 2 2	2 3 3 2 3	3 3 3 3 3
	2 2 3 2 2	2 3 3 2 2	3 3 2 3 3	2 3 3 3 3	3 3 3 4 4	4 4 4 4 4
	3 3 4 4 3	3 4 4 4 3	4 4 4 4 4	4 4 4 4 5	5 5 5 5 5	6 6 6 6 6
	2 3 3 3 3	3 3 3 3 3	3 3 3 3 3	4 3 4 4 4	4 4 4 4 4	4 5 5 5 5
	2 2 2 2 2	3 2 2 2 3	2 2 3 2 3	3 3 3 3 3	3 3 3 3 3	3 3 4 4 4
	2 2 1 1 2	2 1 1 2 2	1 2 2 2 2	2 2 2 2 2	2 2 2 2 2	2 2 2 3 3
	1 1 0 1 1	1 0 1 1 1	1 1 1 1 1	1 1 1 1 1	1 1 1 1 1	1 1 1 1 2
10000000	0 0 + 0 0	0 0 + 0 0	0 + 0 0 +	0 0 + 0 +	0 + 0 0 0	0 0 0 0 0
01000000	0 + 0 0 0	+ 0 0 0 0	0 0 0 0 0	0 0 0 0 +	+ 0 0 + 0	0 0 0 0 0
00100000	0 0 0 + 0	0 0 0 + 0	0 0 + 0 0	+ 0 0 0 0	+ 0 + 0 0	0 0 0 0 0
00010000	0 0 0 0 +	0 0 0 0 +	0 0 0 0 0	0 0 + + 0	0 0 0 0 +	0 0 0 0 0
00001000	+ 0 0 0 0	0 0 0 0 0	0 0 + 0 +	0 + 0 0 0	0 0 0 0 0	+ 0 0 0 0
00000100	0 + 0 0 +	0 0 0 + 0	0 + 0 + 0	0 0 0 0 0	0 0 0 0 0	0 + 0 0 0
00000010	0 0 0 + 0	0 0 + 0 0	+ 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 + 0 0
00000001	0 0 - 0 0	0 - 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 + 0
10100000	0 0 0 + 0	0 0 0 + 0	0 0 + 0 0	+ 0 0 0 -	0 - 0 0 0	0 0 0 0 0
01010000	0 - 0 0 0	- 0 0 0 0	0 0 0 0 0	0 0 + + 0	0 0 0 - 0	0 0 0 0 0
00110000	0 0 0 0 +	0 0 0 0 +	0 0 0 0 0	- 0 0 + 0	0 0 - 0 0	0 0 0 0 0
00011000	+ 0 0 0 0	0 0 0 0 -	0 0 0 0 +	0 + 0 0 0	0 0 0 0 -	0 0 0 0 0
00001100	- 0 0 0 0	0 0 0 + 0	0 + 0 + 0	0 0 0 0 0	0 0 0 0 0	- 0 0 0 0
00000110	0 0 0 + 0	0 0 + 0 0	+ 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 - 0 0 0
00000011	0 0 - 0 0	0 - 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 - 0 0
10110000	0 0 0 0 +	0 0 0 0 +	0 0 0 0 0	- 0 - 0 0	0 + 0 0 0	0 0 0 0 0
01110000	0 - 0 0 0	- 0 0 0 0	0 0 0 0 0	- 0 0 + 0	+ 0 0 0 0	0 0 0 0 0
01011000	0 0 0 0 0	+ 0 0 0 0	0 0 0 0 +	0 + 0 0 0	0 0 0 + 0	0 0 0 0 0
00111000	+ 0 0 0 0	0 0 0 0 -	0 0 - 0 0	0 + 0 0 0	0 0 + 0 0	0 0 0 0 0
00011100	- 0 0 0 -	0 0 0 0 0	0 + 0 + 0	0 0 0 0 0	0 0 0 0 +	0 0 0 0 0
00001110	0 0 0 + 0	0 0 + 0 0	+ 0 0 0 0	0 0 0 0 0	0 0 0 0 0	+ 0 0 0 0
00000111	0 0 - 0 0	0 - 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 + 0 0 0
11110000	0 - 0 0 0	- 0 0 0 0	0 0 0 0 0	- 0 - 0 -	0 0 0 0 0	0 0 0 0 0
10111000	+ 0 0 0 0	0 0 0 0 -	0 0 - 0 -	0 0 0 0 0	0 - 0 0 0	0 0 0 0 0
01111000	0 0 0 0 0	+ 0 0 0 0	0 0 - 0 0	0 + 0 0 0	- 0 0 0 0	0 0 0 0 0
01011100	0 + 0 0 0	0 0 0 0 0	0 + 0 + 0	0 0 0 0 0	0 0 0 - 0	0 0 0 0 0
00111100	- 0 0 0 -	0 0 0 - 0	0 0 0 + 0	0 0 0 0 0	0 0 - 0 0	0 0 0 0 0
00011110	0 0 0 0 0	0 0 + 0 0	+ 0 0 0 0	0 0 0 0 0	0 0 0 0 -	0 0 0 0 0
00001111	0 0 - 0 0	0 - 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	- 0 0 0 0
11111000	0 0 0 0 0	+ 0 0 0 0	0 0 - 0 -	0 0 0 0 +	0 0 0 0 0	0 0 0 0 0

	10000 01000 00100 00010 00001 00000 00000 00000	00010 00001 00010 00001 00000 10000 01000 00100	00000 00000 10000 11000 01100 00110 00011 00001	10000 01100 11010 11111 00111 00001 00000 00000	00110 00101 00111 00111 10011 11000 11000 01000	00001 10001 01001 11101 11111 11110 00110 00010
10111100	0-000	0+00+	0000+	0+-00	0000-	00000
01121000	-0000	+00+0	00++0	00000	++000	000+0
01111100	-00-0	0+000	0+00+	+0000	000-0	00+00
01011110	00-00	00+-0	+0000	+00-0	000-0	0+000
00111110	--000	00+0+	00000	00-00	00-00	+000+
00011111	0--00	000-0	00000	0-000	00-0+	0000+
11121000	00-00	+0000	00++0	00000	++000	0-0+0
11111100	000-0	0+000	-+00+	000+0	00000	00+00
10111110	0-000	00+0+	00000	0+-00	0000-	+0000
01121100	-000-	0+0+0	0000+	00000	-00-0	000--
01111110	-00-0	00+00	0+000	+000-	000-0	00000
01011111	00-00	000-0	+0000	+00-0	000-0	0+000
00111111	--000	0000+	00000	00-00	00-00	+000+
11221000	00000	+0000	00++0	0000+	++000	+0++0
11121100	00-0-	0+000	0000+	000+0	-000+	000-0
11111110	000-0	00+00	-+000	000+-	00000	0-000
10111111	0-000	0000+	00000	0+-00	0000-	+0000
01122100	-0000	0+0+0	000++	-0000	0+-00	00000
01121110	-000-	00++0	00+00	00000	000-0	000+-
01111111	-00-0	00000	0+000	+000-	000-0	00+00
11221100	0000-	0+000	0-00+	00-00	-0000	00--0
11122100	00-00	0+000	+00++	0+000	0+000	00000
11121110	00-0-	00+00	00+00	000+0	0000+	0-0+0
11111111	000-0	00000	-+000	000+-	00000	0-+00
01122110	-0000	-0++0	00000	-0000	0--00	00000
01121111	-000-	000+0	00+00	00000	-00-0	0000-
11222100	000-0	0+00-	000++	00000	0+000	00-00
11221110	0000-	00+00	0-+00	00-0+	00000	+00+0
11122110	00-00	-0+00	+0000	0+000	0-000	0-000
11121111	00-0-	00000	00+00	000+0	-000+	0-000

	1 0 0 0 0 0 1 1 1 0 1 1 1 0 1 1 2 1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 0 0 0	0 1 1 1 0 0 1 1 0 1 0 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 0 1 1 1 1 0 0 1 0 1 0 0 0 0	0 0 0 1 1 1 1 0 1 1 1 0 1 2 1 1 1 1 2 2 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 0 1 1 0 0	1 1 0 0 0 1 0 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 2 1 1 1 1 1 1 1 1 1 0 1 1 0 1 0 0 1	1 1 1 1 0 1 1 1 1 1 2 1 1 1 1 2 2 2 1 2 1 2 1 1 2 1 1 1 1 1 0 0 1 1 1 0 0 0 1 0	0 1 1 1 1 1 1 1 1 1 1 2 2 1 1 2 2 2 2 2 1 2 1 2 1 1 1 1 1 1 1 0 1 1 1 1 0 0 0 1
10111100	0 − 0 − 0	0 0 0 0 0	− − 0 0 0	0 0 0 − −	0 0 0 0 −	− 0 0 0 0
01121000	+ 0 0 0 0	0 0 + + 0	0 0 0 0 0	+ + 0 0 0	0 0 0 + 0	0 0 0 0 0
01111100	0 0 0 0 0	+ + 0 − 0	0 0 0 0 0	0 − 0 0 0	0 0 + 0 0	0 0 0 + +
01011110	+ 0 0 0 0	0 0 0 0 0	0 0 + − 0	0 + 0 0 0	+ 0 0 0 0	0 + 0 0 0
00111110	0 0 0 0 0	0 + − 0 0	0 + 0 0 −	0 0 0 0 0	0 − 0 − 0	0 0 0 0 −
00011111	0 0 − 0 0	0 0 − 0 0	+ 0 0 − 0	+ 0 0 0 0	+ 0 0 0 0	0 + − 0 0
11121000	0 0 − 0 −	0 0 0 0 0	− 0 − 0 0	0 0 0 0 −	0 0 0 0 0	0 0 0 0 0
11111100	0 − 0 0 +	+ 0 0 0 0	0 0 + 0 0	0 0 0 − 0	0 0 0 0 −	− 0 0 0 0
10111110	0 − + 0 0	0 0 0 0 +	0 + 0 0 0	0 0 + 0 +	0 0 0 0 0	+ 0 0 0 0
01121100	0 0 0 0 0	0 0 0 − 0	0 0 0 0 0	− − 0 0 0	0 0 0 − 0	0 0 0 + 0
01111110	+ 0 0 0 0	− + 0 0 0	0 0 0 0 −	0 + 0 0 0	0 − 0 0 0	0 0 0 0 −
01011111	+ 0 0 0 −	0 0 0 − 0	0 0 0 − 0	0 0 0 0 0	+ 0 0 0 0	0 + − 0 0
00111111	0 0 0 − 0	0 + − 0 0	0 0 0 0 −	+ 0 0 0 0	0 − + 0 0	0 0 0 + 0
11221000	0 0 0 + 0	+ 0 0 0 0	0 + 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
11121100	0 0 0 0 +	0 0 0 0 0	+ 0 + 0 0	0 0 0 0 +	0 0 0 0 −	0 0 0 0 0
11111110	0 − 0 0 0	− 0 0 0 +	0 0 − 0 0	0 0 + 0 0	0 0 0 0 0	+ 0 0 0 0
10111111	0 − + − 0	0 0 0 0 +	− 0 0 0 0	0 0 + − 0	0 0 0 0 −	0 0 0 0 0
01122100	0 0 0 0 0	0 0 0 − 0	0 0 0 0 0	− − 0 0 0	0 0 − − 0	0 0 − 0 −
01121110	+ 0 0 0 0	0 0 + 0 0	0 0 0 0 0	0 + 0 0 0	0 − 0 + 0	0 − 0 0 0
01111111	+ 0 0 0 0	0 + 0 − 0	0 0 0 0 −	0 0 0 0 0	0 − + 0 0	0 0 0 + 0
11221100	0 0 0 − 0	− 0 0 0 0	0 − 0 0 0	0 0 0 0 0	0 0 0 0 −	0 0 0 − 0
11122100	0 0 0 0 +	0 0 0 0 0	+ 0 + 0 0	0 0 0 + +	0 0 0 0 0	+ 0 − 0 0
11121110	0 0 − 0 0	0 0 0 0 0	0 0 − 0 0	0 0 + 0 −	0 0 0 0 0	0 − 0 0 0
11111111	0 − 0 0 +	0 0 0 0 +	0 0 0 0 0	0 0 + − 0	0 0 0 0 −	0 0 0 0 0
01122110	+ 0 0 0 0	0 0 + 0 0	0 0 0 0 +	0 + 0 0 0	+ 0 0 + 0	0 0 0 0 +
01121111	+ 0 0 0 0	0 0 + − 0	0 0 0 0 0	− 0 0 0 0	0 − 0 0 0	0 − 0 + 0
11222100	0 0 0 − 0	− 0 0 0 0	0 − 0 0 0	0 0 0 + 0	0 0 + 0 0	+ 0 0 0 +
11221110	0 0 0 0 0	+ 0 0 0 0	0 + 0 0 0	0 0 + 0 0	0 + 0 0 0	0 0 0 0 0
11122110	0 0 − 0 0	0 0 0 0 −	0 0 − 0 0	0 0 0 0 −	+ 0 0 0 0	− 0 0 0 0
11121111	0 0 − 0 +	0 0 0 0 0	+ 0 0 0 0	0 0 + 0 0	0 0 0 0 −	0 − 0 0 0

	0 0 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 3 2 2 2 2 2 2 1 2 1 1 1 1 1 1 0 1 1 0 1 0 0 1	1 1 0 1 1 1 1 1 2 1 1 1 1 2 2 2 2 2 3 3 2 2 2 2 2 2 1 2 1 1 1 1 1 0 1 0 1 1 0 0	1 1 1 0 1 1 1 1 1 2 2 2 1 1 2 2 2 2 2 3 2 2 2 2 2 2 1 2 2 1 1 1 1 2 1 0 1 1 1 0	1 1 1 1 1 1 1 1 1 2 2 2 2 1 2 3 3 2 2 3 2 2 2 2 2 2 1 2 2 2 1 1 1 2 1 0 1 1 1 0	1 1 1 1 1 2 1 1 1 2 2 2 2 2 2 3 3 3 2 3 2 3 2 2 3 1 2 2 2 2 1 1 1 2 1 1 0 1 1 0	1 1 1 1 1 2 1 1 2 2 2 2 2 2 2 3 3 3 4 3 2 3 2 3 3 2 2 2 2 2 1 1 2 1 1 1 1 1 0 1
10111100	0 − 0 0 0	0 0 0 0 0	0 0 0 + +	0 0 0 0 0	+ 0 0 0 0	0 0 0 0 0
01121000	0 0 0 0 0	+ 0 0 0 0	+ 0 + 0 0	0 0 + + 0	0 0 0 + 0	0 0 0 0 0
01111100	0 0 0 0 0	0 + 0 0 −	0 0 0 0 0	0 − 0 − 0	0 0 0 0 0	0 0 + 0 0
01011110	0 0 + 0 +	0 0 0 0 0	0 + 0 0 0	0 + + 0 0	0 0 + 0 0	0 + 0 0 0
00111110	0 0 0 0 0	0 − 0 − 0	0 0 − 0 0	0 0 0 0 0	− 0 0 0 0	− 0 0 0 −
00011111	0 0 0 − 0	0 0 0 − 0	− 0 0 0 +	0 0 0 0 +	0 0 0 0 +	0 0 0 0 0
11121000	− 0 0 0 0	0 0 − 0 0	+ 0 0 − 0	0 0 + 0 0	0 0 0 + 0	0 0 0 0 0
11111100	0 − 0 0 0	0 0 0 0 −	0 0 0 + 0	0 − 0 0 0	0 0 0 0 0	0 0 + 0 0
10111110	0 + 0 0 0	0 0 + − 0	0 0 0 0 0	0 0 0 0 0	− 0 0 0 0	− 0 0 0 −
01121100	0 0 0 + 0	0 + 0 0 0	0 + 0 0 0	0 0 0 − 0	0 0 0 − 0	0 0 0 0 0
01111110	0 0 + 0 0	0 − 0 0 0	0 0 − 0 0	0 + 0 0 0	0 0 + 0 0	0 + 0 0 0
01011111	0 0 + − 0	0 0 0 0 −	− 0 0 0 0	− 0 0 0 0	0 − 0 0 0	0 0 0 0 0
00111111	0 0 0 0 0	+ 0 0 − 0	0 0 0 0 +	0 0 0 0 +	0 0 0 0 +	0 0 0 + 0
11221000	− 0 0 0 0	− 0 − 0 0	0 0 − − 0	0 0 0 − 0	0 0 0 0 0	0 0 0 0 0
11121100	0 − 0 + 0	0 0 0 0 0	0 + 0 + 0	0 0 0 0 0	0 0 0 − 0	0 0 0 0 0
11111110	0 + + 0 0	0 0 + 0 0	0 0 0 0 0	0 + 0 0 0	0 0 + 0 0	0 + 0 0 0
10111111	− 0 0 0 0	0 0 0 − 0	0 0 0 0 +	0 0 0 0 +	0 0 0 0 +	0 0 0 + 0
01122100	0 0 0 0 −	0 0 0 0 0	0 0 0 0 0	0 0 0 − 0	0 0 0 − 0	0 0 − 0 0
01121110	0 0 0 0 0	0 − 0 0 0	0 − − 0 0	0 0 − 0 0	0 0 0 0 0	0 + 0 0 +
01111111	0 0 + 0 0	+ 0 0 0 −	0 0 0 0 0	− 0 0 0 0	0 − 0 0 0	0 0 0 + 0
11221100	0 − 0 0 0	0 − 0 0 0	0 0 0 + 0	0 0 0 + 0	0 0 0 0 0	0 0 0 0 0
11122100	0 0 0 0 −	0 0 0 0 0	0 0 0 + 0	0 0 0 0 0	0 0 0 − 0	0 0 − 0 0
11121110	0 + 0 0 0	0 0 + 0 0	0 − 0 0 0	0 0 − 0 0	0 0 0 0 0	0 + 0 0 +
11111111	− 0 + 0 0	0 0 0 0 −	0 0 0 0 0	− 0 0 0 0	0 − 0 0 0	0 0 0 + 0
01122110	0 0 0 0 +	0 0 0 0 0	0 0 − 0 0	0 0 − 0 0	0 0 − 0 0	− 0 0 0 0
01121111	0 0 0 + 0	+ 0 0 0 0	+ 0 0 0 0	0 0 0 0 0	0 − 0 0 −	0 0 0 0 0
11222100	0 0 0 0 0	0 0 0 0 0	0 0 0 + 0	0 0 0 + 0	0 0 0 0 0	0 0 − 0 0
11221110	0 + 0 0 0	0 + + 0 0	0 0 + 0 0	0 0 0 0 0	0 0 0 0 0	0 + 0 0 +
11122110	0 0 0 0 +	0 0 + 0 0	0 0 0 0 0	0 0 − 0 0	0 0 − 0 0	− 0 0 0 0
11121111	− 0 0 + 0	0 0 0 0 0	+ 0 0 0 0	0 0 0 0 0	0 − 0 0 −	0 0 0 0 0

	11111 21222 22322 33443 23333 22222 22112 11011	12111 12222 23322 34443 33333 32223 21122 10111	21121 22222 33233 44444 33333 22323 12222 11111	12121 22222 23333 44445 43444 33333 22222 11111	21222 23323 33344 55555 44444 33333 22222 11111	22222 33333 44444 66666 45555 33444 22233 11112
10111100	- 0 0 0 -	0 0 0 - 0	0 - 0 0 0	0 0 0 0 0	0 + 0 0 0	00000
01121000	0 0 0 0 0	+ 0 0 0 +	0 0 0 0 0	0 + 0 + 0	0 0 0 0 0	00000
01111100	0 + 0 0 0	0 0 0 - 0	0 0 0 + 0	0 0 0 0 0	+ 0 0 0 0	00000
01011110	0 0 0 0 0	0 0 + 0 0	+ 0 0 0 0	0 0 0 0 0	0 0 0 + 0	00000
00111110	0 0 0 - 0	0 0 0 0 0	+ 0 0 0 0	0 0 0 0 0	0 0 + 0 0	00000
00011111	0 0 - 0 0	0 - 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 +	00000
11121000	0 0 0 0 0	+ 0 0 0 +	0 0 0 0 -	0 0 - 0 0	0 0 0 0 0	00000
11111100	0 + 0 0 0	0 0 0 - 0	0 - 0 0 0	0 0 0 0 -	0 0 0 0 0	00000
10111110	0 0 0 - 0	0 0 - 0 0	0 0 0 0 0	0 0 0 0 0	0 - 0 0 0	00000
01121100	0 + 0 0 +	0 0 0 0 0	0 0 0 + 0	0 0 0 - 0	0 0 0 0 0	00000
01111110	0 0 0 - 0	0 0 0 0 0	+ 0 0 0 0	0 0 0 0 0	- 0 0 0 0	00000
01011111	0 0 - 0 0	0 - 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 - 0	00000
00111111	0 0 0 0 0	0 - 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 - 0 0	00000
11221000	0 0 0 0 0	+ 0 0 0 +	0 0 + 0 0	+ 0 0 0 0	0 0 0 0 0	00000
11121100	0 + 0 0 +	0 0 0 0 0	0 - 0 0 0	0 0 + 0 0	0 0 0 0 0	00000
11111110	0 0 0 - 0	0 0 - 0 0	0 0 0 0 0	0 0 0 0 +	0 0 0 0 0	00000
10111111	0 0 + 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 + 0 0 0	00000
01122100	- 0 0 0 0	0 0 0 0 0	0 0 0 + 0	0 + 0 0 0	0 0 0 0 0	00000
01121110	0 0 0 0 0	0 0 0 0 0	+ 0 0 0 0	0 0 0 + 0	0 0 0 0 0	00000
01111111	0 0 0 0 0	0 - 0 0 0	0 0 0 0 0	0 0 0 0 0	+ 0 0 0 0	00000
11221100	0 + 0 0 +	0 0 0 + 0	0 0 0 0 0	- 0 0 0 0	0 0 0 0 0	00000
11122100	- 0 0 0 0	0 0 0 0 0	0 - 0 0 -	0 0 0 0 0	0 0 0 0 0	00000
11121110	0 0 0 0 0	0 0 - 0 0	0 0 0 0 0	0 0 - 0 0	0 0 0 0 0	00000
11111111	0 0 + 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 -	0 0 0 0 0	00000
01122110	0 0 0 0 0	0 0 0 0 0	+ 0 0 0 0	0 - 0 0 0	0 0 0 0 0	00000
01121111	0 0 0 0 0	0 - 0 0 0	0 0 0 0 0	0 0 0 - 0	0 0 0 0 0	00000
11222100	- 0 0 0 0	0 0 0 + 0	0 0 + 0 0	0 0 0 0 0	0 0 0 0 0	00000
11221110	0 0 0 + 0	0 0 0 0 0	0 0 0 0 0	+ 0 0 0 0	0 0 0 0 0	00000
11122110	0 0 0 0 0	0 0 - 0 0	0 0 0 0 +	0 0 0 0 0	0 0 0 0 0	00000
11121111	0 0 + 0 0	0 0 0 0 0	0 0 0 0 0	0 0 + 0 0	0 0 0 0 0	00000

	10000 01000 00100 00010 00001 00000 00000 00000	00010 00001 00010 00001 00000 10000 01000 00100	00000 00000 10000 11000 01100 00110 00011 00001	10000 01100 11010 11111 00111 00001 00000 00000	00110 00101 00111 00111 10011 11000 11000 01000	00001 10001 01001 11101 11111 11110 00110 00010
01122210	-0000	00++0	0000+	-0000	00-+0	0000+
01122111	-0000	-00+0	000+0	-0000	00-00	00000
11232100	0-000	0+000	000++	00000	++000	000+0
11222110	000-0	-0+0-	00000	0000+	0-000	+0000
11221111	0000-	00000	0-+00	00-0+	-0000	+0-00
11122210	00-00	00+00	+000+	0+0-0	0000-	00000
11122111	00-00	-0000	+00+0	0+000	00000	0-000
01122211	-0000	0-0+0	00000	-0000	00-+0	0000+
12232100	00000	0+000	000++	00000	++000	00++0
11232110	0-000	-0+00	00-00	00000	0-000	+00-0
11222210	000-0	00+0-	0+00+	00+00	00000	00000
11222111	000-0	-000-	000+0	0000+	00000	+0-00
11122211	00-00	0-000	+0000	0+0-0	0000-	00000
01122221	-0000	000+0	00000	-0000	00-+0	0000+
12232110	00000	-0+00	00-00	0000-	0-000	0-0-0
11232210	0-00-	00+00	0000+	00+00	0000+	000-+
11232111	0-000	-0000	00-+0	00000	+0000	+0000
11222211	000-0	0-00-	0+000	00+00	00000	00-00
11122221	00-00	00000	+0000	0+0-0	0000-	0+000
12232210	0000-	00+00	0-00+	000-0	000-0	000-0
12232111	00000	-0000	00-+0	0000-	+0000	0-+00
11233210	0-000	00+0+	0000+	0+000	0++00	00000
11232211	0-00-	0-000	00000	00+00	+000+	0000+
11222221	000-0	0000-	0+000	00+0-	00000	-0000
12233210	000-0	00+00	-000+	-0000	0+000	00000
12232211	0000-	0-000	0-000	000-0	+00-0	00+00
11233211	0-000	0-00+	000-0	0+000	00+00	00000
11232221	0-00-	00000	00+00	00+00	0000+	-000+
12243210	00-00	00+-0	0000+	00000	0+000	000+0
12233211	000-0	0-000	-00-0	-0000	00000	00+00

	1 0 0 0 0 0 1 1 1 0 1 1 1 0 1 1 2 1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 0 0 0	0 1 1 1 0 0 1 1 0 1 0 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 0 1 1 1 1 0 0 1 0 1 0 0 0 0	0 0 0 1 1 1 1 0 1 1 1 0 1 2 1 1 1 1 2 2 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 0 1 1 0 0	1 1 0 0 0 1 0 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 2 1 1 1 1 1 1 1 1 1 0 1 1 0 1 0 0 1	1 1 1 1 0 1 1 1 1 1 2 1 1 1 1 2 2 2 1 2 1 2 1 1 2 1 1 1 1 1 0 0 1 1 1 0 0 0 1 0	0 1 1 1 1 1 1 1 1 1 1 2 2 1 1 2 2 2 2 2 1 2 1 2 1 1 1 1 1 1 1 0 1 1 1 1 0 0 0 1
01122210	0 0 0 0 0	0 + 0 0 0	0 0 0 + 0	0 + 0 0 0	0 0 0 + 0	0 0 0 0 +
01122111	+ 0 0 0 0	0 0 + − 0	0 0 0 0 +	− 0 0 0 0	+ 0 − 0 0	0 0 − 0 0
11232100	0 0 0 − 0	0 0 0 0 0	− − 0 0 0	− 0 0 0 −	0 0 0 − 0	0 0 0 0 0
11222110	0 0 0 0 0	+ 0 0 0 −	0 + 0 0 −	0 0 0 0 0	0 0 0 0 0	− 0 0 0 −
11221111	0 0 0 − 0	0 0 0 0 0	0 0 0 0 0	0 0 + 0 0	0 + 0 0 −	0 0 0 − 0
11122210	0 − 0 0 0	0 0 0 0 0	0 0 − + 0	0 0 0 0 −	0 0 0 0 0	− 0 0 0 0
11122111	0 0 − 0 +	0 0 0 0 −	+ 0 0 0 0	0 0 0 + 0	+ 0 0 0 0	0 0 − 0 0
01122211	0 0 0 0 0	0 + 0 − 0	0 0 0 + 0	− 0 0 0 0	0 0 − 0 0	0 0 − − 0
12232100	0 0 0 0 +	+ 0 0 + 0	0 0 + 0 0	0 + 0 0 0	0 0 0 0 0	0 0 0 0 0
11232110	0 0 + 0 0	0 0 + 0 0	0 + 0 0 0	0 0 0 0 +	0 0 0 + 0	0 0 0 0 0
11222210	0 − 0 0 0	+ − 0 0 0	0 + 0 0 0	0 0 0 0 0	0 0 0 0 0	− 0 0 0 −
11222111	0 0 0 − 0	0 0 0 0 −	0 0 0 0 −	0 0 0 + 0	0 0 + 0 0	0 0 0 0 0
11122211	0 − 0 0 +	0 0 0 0 0	+ 0 0 + 0	0 0 0 + 0	0 0 0 0 +	0 0 − 0 0
01122221	− 0 0 0 0	0 + − 0 0	0 0 0 + −	0 0 0 0 0	− − 0 0 0	0 − 0 0 0
12232110	− 0 0 0 0	− 0 0 0 0	0 0 − 0 0	0 − 0 0 0	0 0 0 0 0	0 0 0 0 0
11232210	0 0 0 0 0	0 0 0 0 0	0 + 0 0 0	0 0 0 0 +	0 0 0 + 0	0 0 0 0 0
11232111	0 0 + − 0	0 0 + 0 0	− 0 0 0 0	− 0 0 0 0	0 0 0 0 0	0 0 0 0 0
11222211	0 − 0 − 0	0 − 0 0 0	0 0 0 0 0	0 0 0 + 0	0 0 + 0 +	0 0 0 + 0
11122221	0 − + 0 0	0 0 0 0 +	0 0 0 + 0	0 0 + 0 0	− 0 0 0 0	0 − 0 0 0
12232210	0 0 0 0 0	− 0 0 0 0	0 0 − 0 0	0 − 0 0 0	0 0 0 0 0	0 0 0 0 0
12232111	− 0 0 0 +	0 0 0 + 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0
11233210	0 0 0 0 0	0 0 0 0 0	0 + 0 0 0	0 0 0 0 +	0 0 0 + 0	+ 0 0 0 +
11232211	0 0 0 − 0	0 0 0 0 0	− 0 0 0 0	− 0 0 0 0	0 0 0 0 +	0 0 0 + 0
11222221	0 − 0 0 0	0 − 0 0 +	0 0 0 0 +	0 0 + 0 0	0 + 0 0 0	0 0 0 0 0
12233210	0 0 0 0 0	− 0 0 0 0	0 0 − 0 0	0 − 0 0 0	0 0 0 0 0	+ 0 0 0 +
12232211	0 0 0 0 +	0 0 0 + 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 +	0 0 0 + 0
11233211	0 0 0 − 0	0 0 0 0 0	− 0 0 0 0	− 0 0 − 0	0 0 − 0 0	0 0 − 0 0
11232221	0 0 − 0 0	0 0 − 0 0	0 0 0 0 0	0 0 + 0 0	0 + 0 0 0	0 + 0 0 0
12243210	0 0 0 0 0	0 0 0 0 0	0 0 − 0 0	0 − 0 0 −	0 0 0 − 0	0 0 0 0 0
12233211	0 0 0 0 +	0 0 0 + 0	0 0 0 0 0	0 0 0 − 0	0 0 − 0 0	0 0 − 0 0

	00111	11011	11101	11111	11111	11111
	11111	11121	11112	11112	21112	21122
	11222	11122	22112	22212	22222	22222
	22322	22233	22223	33223	33323	33343
	22221	22222	22222	22222	23223	23233
	21111	21211	21221	21222	12222	22222
	11011	11101	11121	11121	11121	11211
	01001	01100	01110	01110	10110	11101
01122210	0000+	0+000	0+000	0+000	+0000	00000
01122111	00000	+0000	+0000	+000+	00000	00000
11232100	00000	00000	000+0	000+0	000+0	00000
11222110	00000	00+00	00+00	00000	00−00	−0000
11221111	−0000	−0000	00000	00000	0−00−	000−0
11122210	0−00+	00000	0+000	0+000	+0000	00000
11122111	−0000	00000	+0000	+000+	00000	00000
01122211	000−0	0000−	0000−	00000	00000	00000
12232100	00000	00000	000+0	000+0	000+0	00+00
11232110	00000	00+00	00+00	00+00	00000	−000−
11222210	0−000	0−000	00000	0+000	+0000	00000
11222111	−0000	−0000	00000	+000+	00000	000−0
11122211	000−0	0000−	0000−	00000	00000	00000
01122221	00−00	000−0	00000	00000	00000	00000
12232110	00000	00+00	00+00	00+00	00+00	0+000
11232210	0−000	0−000	0−000	00000	+0000	0000−
11232111	−0000	−0000	−0000	0000+	0000+	00000
11222211	00000	0000−	0000−	00000	00000	000−0
11122221	00−00	000−0	00000	00000	00000	00000
12232210	0−000	0−000	0−000	0−000	00000	0+000
12232111	−0000	−0000	−0000	−0000	0−000	00000
11233210	0000+	00000	00000	00000	+0000	+0000
11232211	000+0	00000	0000−	00000	0000+	00000
11222221	00−00	000−0	00000	00000	00000	000−0
12233210	0000+	00000	00000	0−000	00−00	00−00
12232211	000+0	0000+	00000	00000	0−000	00000
11233211	00000	00000	0000−	0000−	00000	00000
11232221	00000	000−0	00000	00000	0000+	0000−
12243210	0000+	00000	0+000	00+00	000+0	00000
12233211	00000	0000+	00000	+0000	00000	00+00

	12232221	11233221	12343210	12243211	12233221	11233321	22343210	12343211	12243221	12233321	22343211	12343221	12243321	22343221	12343321	12244321	22343321	12344321	22344321	12354321	22354321	13354321	23354321	22454321	23454321	23464321	23465321	23465421	23465431	23465432
01122210	0	0	0	0	0	0	0	0	0	0	+	0	0	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01122111	0	0	0	0	0	0	−	0	0	0	0	0	0	0	0	0	+	0	0	0	0	0	0	0	0	0	0	0	0	0
11232100	−	0	0	0	−	0	0	0	0	−	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11222110	0	0	0	+	0	0	0	0	0	0	0	0	−	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11221111	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	−	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11122210	0	0	0	0	0	0	0	−	0	0	0	−	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11122111	0	0	+	0	0	0	0	0	0	0	0	0	0	0	−	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01122211	0	0	0	0	0	0	−	0	0	0	0	0	0	−	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12232100	0	+	0	0	0	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11232110	0	0	0	0	0	0	0	0	0	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11222210	0	0	0	+	0	0	0	0	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11222111	0	0	0	0	0	0	0	0	0	0	0	0	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11122211	0	0	+	0	0	0	0	0	0	0	0	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01122221	0	0	0	0	0	0	−	0	0	0	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12232110	0	0	0	0	0	−	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11232210	0	0	0	0	−	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11232111	0	0	0	0	0	0	0	0	0	−	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11222211	0	0	0	0	0	0	0	0	−	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11122221	0	0	+	0	0	0	0	−	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12232210	0	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12232111	0	0	0	0	0	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11233210	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11232211	0	0	0	0	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11222221	0	0	0	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12233210	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12232211	0	−	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11233211	−	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11232221	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12243210	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12233211	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	10000	00010	00000	10000	00110	00001
	01000	00001	00000	01100	00101	10001
	00100	00010	10000	11010	00111	01001
	00010	00001	11000	11111	00111	11101
	00001	00000	01100	00111	10011	11111
	00000	10000	00110	00001	11000	11110
	00000	01000	00011	00000	11000	00110
	00000	00100	00001	00000	01000	00010
12232221	0000-	00000	0-+00	000-+	000-0	0+000
11233221	0-000	-000+	00000	0+000	00+00	-0000
12343210	-0000	00+00	0000+	00000	0+000	000+0
12243211	00-00	0-0-0	000-0	00000	-0000	00000
12233221	000-0	-0000	-0000	-000+	00000	0+000
11233321	0-000	0000+	00000	0+-00	00+0-	0000-
22343210	00000	00+00	0000+	00000	0+000	000+0
12343211	-0000	0-000	000-0	00000	-0000	00-00
12243221	00-00	-00-0	00-00	00000	00000	0+000
12233321	000-0	00000	-+000	-00+0	000+0	00000
22343211	00000	0-000	000-0	00000	-0000	00-00
12343221	-0000	-0000	00-00	0000-	00000	-0000
12243321	00-0-	000-0	00000	000+0	000++	0000+
22343221	00000	-0000	00-00	0000-	00000	--000
12343321	-000-	00000	0-000	00-00	000+0	0000+
12244321	00-00	000-0	+0000	++000	00+00	00000
22343321	0000-	00000	0-000	00--0	0000-	00000
12344321	-00-0	0000-	00000	+0000	00+00	00000
22344321	000-0	0000-	-0000	0-000	00000	00000
12354321	--000	000+0	00000	00000	00+00	0000-
22354321	0--00	00000	00000	0-000	0000+	00000
13354321	-0000	000+0	00000	-0000	000+0	00000
23354321	00-00	00000	+0000	000-0	00000	0+000
22454321	0-000	0000+	00000	00-00	00000	+0000
23454321	000-0	00000	0+000	0000-	00000	00+00
23464321	0000-	00000	00+00	00000	-0000	000+0
23465321	00000	-0000	000+0	00000	0-000	00000
23465421	00000	0-000	0000+	00000	00000	00000
23465431	00000	00-00	00000	00000	00000	00000
23465432	00000	00000	00000	00000	00000	00000

	10000 01110 11101 12111 11111 10111 00011 00000	01110 01101 01111 12112 11111 10111 10010 10000	00011 11011 10121 11122 11111 11101 11100 01100	11000 10111 11111 11221 11211 11111 11011 01001	11110 11111 21111 22212 12112 11111 00111 00010	01111 11111 12211 22222 12121 11111 10111 10001
12232221	+0000	00000	00000	00+00	0+000	0+000
11233221	00-00	00-0-	0000-	00000	-0000	00000
12343210	00000	+0000	0+000	0-000	000-0	0000-
12243211	0000+	000+0	+0000	+0000	00000	00-00
12233221	+0000	0000-	0000-	00000	-0000	00000
11233321	0-000	0-000	000-0	00000	00000	00000
22343210	00000	+0000	0++00	0000+	00000	+0000
12343211	000-0	000+0	00000	+0000	00+00	000+0
12243221	+0+00	00+00	00000	00000	-0000	0-000
12233321	0-000	0-000	000-0	00000	00000	00000
22343211	000--	00000	-0000	000-0	0000-	00000
12343221	+0000	00+00	0000+	00000	0+000	00000
12243321	00000	00000	000-0	00000	00000	0-000
22343221	00-00	0000-	00000	00-00	00000	00000
12343321	00000	0+000	00000	00000	0+000	000-0
12244321	00000	00000	000-0	00000	+0000	00-00
22343321	0-000	00000	00000	00-00	0000+	00000
12344321	00000	0+000	0000-	00000	00+00	0000-
22344321	0-000	0000+	00000	000-0	00000	+0000
12354321	00000	00+00	00000	-0000	000+0	00000
22354321	00-00	00000	+0000	0000-	00000	00000
13354321	-0000	000+0	00000	0-000	00000	00000
23354321	0000-	00000	00+00	00000	00000	00000
22454321	000-0	00000	0+000	00000	00000	00000
23454321	00000	-0000	00000	00000	00000	00000
23464321	00000	00000	00000	00000	00000	00000
23465321	00000	00000	00000	00000	00000	00000
23465421	00000	00000	00000	00000	00000	00000
23465431	00000	00000	00000	00000	00000	00000
23465432	00000	00000	00000	00000	00000	00000

	00111	11011	11101	11111	11111	11111
	11111	11121	11112	11112	21112	21122
	11222	11122	22112	22212	22222	22222
	22322	22233	22223	33223	33323	33343
	22221	22222	22222	22222	23223	23233
	21111	21211	21221	21222	12222	22222
	11011	11101	11121	11121	11121	11211
	01001	01100	01110	01110	10110	11101
12232221	00+00	00000	00000	00000	0−000	0+000
11233221	00000	000−0	00000	0000−	00000	+0000
12343210	00000	0−000	00−00	000−0	00000	00000
12243211	000−0	00000	−0000	00000	000−0	00000
12233221	00+00	00000	00000	+0000	00−00	00000
11233321	00000	000−0	0000+	00000	−0000	00000
22343210	0+000	00+00	000+0	00000	00000	00000
12343211	00000	+0000	00000	000+0	00000	00000
12243221	00000	00000	−0000	00+00	00000	00000
12233321	00+00	0000−	00000	0+000	00000	00000
22343211	−0000	00000	000−0	00000	00000	00000
12343221	00000	+0000	00−00	00000	00000	00000
12243321	000+0	00000	0−000	00000	00000	00000
22343221	−0000	00+00	00000	00000	00000	00000
12343321	00000	0+000	00000	00000	00000	00000
12244321	0000+	00000	00000	00000	00000	00000
22343321	0−000	00000	00000	00000	00000	00000
12344321	00000	00000	00000	00000	00000	00000
22344321	00000	00000	00000	00000	00000	00000
12354321	00000	00000	00000	00000	00000	00000
22354321	00000	00000	00000	00000	00000	00000
13354321	00000	00000	00000	00000	00000	00000
23354321	00000	00000	00000	00000	00000	00000
22454321	00000	00000	00000	00000	00000	00000
23454321	00000	00000	00000	00000	00000	00000
23464321	00000	00000	00000	00000	00000	00000
23465321	00000	00000	00000	00000	00000	00000
23465421	00000	00000	00000	00000	00000	00000
23465431	00000	00000	00000	00000	00000	00000
23465432	00000	00000	00000	00000	00000	00000

	11111 21222 22322 33443 23333 22222 22112 11011	12111 12222 23322 34443 33333 32223 21122 10111	21121 22222 33233 44444 33333 22323 12222 11111	12121 22222 23333 44445 43444 33333 22222 11111	21222 23323 33344 55555 44444 33333 22222 11111	22222 33333 44444 66666 45555 33444 22233 11112
12232221	00000	00000	00000	00000	00000	00000
11233221	00000	00000	00000	00000	00000	00000
12343210	00000	00000	00000	00000	00000	00000
12243211	00000	00000	00000	00000	00000	00000
12233221	00000	00000	00000	00000	00000	00000
11233321	00000	00000	00000	00000	00000	00000
22343210	00000	00000	00000	00000	00000	00000
12343211	00000	00000	00000	00000	00000	00000
12243221	00000	00000	00000	00000	00000	00000
12233321	00000	00000	00000	00000	00000	00000
22343211	00000	00000	00000	00000	00000	00000
12343221	00000	00000	00000	00000	00000	00000
12243321	00000	00000	00000	00000	00000	00000
22343221	00000	00000	00000	00000	00000	00000
12343321	00000	00000	00000	00000	00000	00000
12244321	00000	00000	00000	00000	00000	00000
22343321	00000	00000	00000	00000	00000	00000
12344321	00000	00000	00000	00000	00000	00000
22344321	00000	00000	00000	00000	00000	00000
12354321	00000	00000	00000	00000	00000	00000
22354321	00000	00000	00000	00000	00000	00000
13354321	00000	00000	00000	00000	00000	00000
23354321	00000	00000	00000	00000	00000	00000
22454321	00000	00000	00000	00000	00000	00000
23454321	00000	00000	00000	00000	00000	00000
23464321	00000	00000	00000	00000	00000	00000
23465321	00000	00000	00000	00000	00000	00000
23465421	00000	00000	00000	00000	00000	00000
23465431	00000	00000	00000	00000	00000	00000
23465432	00000	00000	00000	00000	00000	00000

References

1. H. Azad, Structure constants of algebraic groups. – *J. Algebra* **75**, No. 1, 209–222 (1982).
2. H.Azad, The Jacobi identity. – *Punjab Univ. J. Math.* **16**, 9–29 (1983).
3. N. Bourbaki, *Groupes et algèbres de Lie. Ch. 4–6.* – Hermann: Paris (1968), p. 288.
4. N. Bourbaki, *Groupes et algèbres de Lie. Ch. 7,8.* – Hermann: Paris (1975), p. 271.
5. N. Burgoyne, Williamson C. Some computations involving simple Lie algebras – In: *Proc. 2nd Symp. Symbolic and algebraic manipulation.* New York: Ass. Comp. Mach. (1971).
6. R. W. Carter, *Simple groups of Lie type.* – Wiley: London et al., (1972), p. 331.
7. A. M. Cohen, Point-like spaces related to Buildings. – *Handbook of Incidence Geometry.* North-Holland, Amsterdam, 647–737 (1995).
8. A.M. Cohen, R. L. Griess, B. Lisser, The group $L(2,61)$ embeds in the Lie group of type E_8. – *Comm. Algebra*, **21**, No. 6, 1889–1907 (1993).
9. L. Di Martino, M. Ch. Tamburini, 2–generation of the finite simple groups and related problems. – *Proc. Conf. Generations and Relations in Groups and Geometries (Lucca – 1990)*, Kluwer Publ., 195–233 (1991).
10. L. Di Martino, N. A. Vavilov, (2,3)–generation of $\mathrm{SL}(n,q)$. I. Cases $n = 5,6,7$. – *Comm. Algebra*, **22**, No.4, 1321–1347 (1994); II. Cases $n \geqslant 8$. *ibid.*, **24**, No. 2, 487–515 (1996).
11. L. Di Martino, N. A. Vavilov, (2,3)–generation of $\mathrm{E}E_6(q)$. – *Unpublished.*
12. I. B. Frenkel, V. Kac, Basic representations of affine Lie algebras and dual resonance models. – *Invent. Math.*, **62**, No. 1, 23–66 (1980).
13. I. B. Frenkel, J. Lepowsky, A. Meurman, *Vertex operator algebras and the Monster.* – Academic Press: N.Y. et al. (1988), p. 502.
14. P. Gilkey, G. Seitz, Some representations of exceptional Lie algebras – *Geom. Dedic.*, **25**, Nos. 1–3, 407–416 (1988).
15. J.-Y. Hée, Groupes de Chevalley et groupes classiques. – *Publ. Math. Univ. Paris VII,*, **17**, 1–54 (1984).
16. J. E. Humphreys, *Introduction to Lie algebras and representation theory, 3rd Printing.* – Springer: New York et al. (1980), p. 171.
17. J. E. Hurley, Some normal subgroups of elementary subgroups of Chevalley groups over rings. – *Amer. J. Math.*, **93**, No. 4, 1059–1069 (1971).
18. J. Hurrelbrink, U. Rehmann, Eine endliche Präsentation der Gruppe $\mathrm{G}G_2(\mathbb{Z})$. – *Math. Z.*, **141**, No. 2, 243–251 (1975).
19. V. Kac, *Infinite dimensional Lie algebras. 2nd ed.* – Cambridge Univ. Press (1985).
20. D. E. Knuth, *The TEXbook.* – Addison Wesley, Reading et al. (1990), p. 483.
21. Yu. I. Manin, *Cubic forms: algebra, geometry, arithmetic.* – North Holland: Amsterdam–London (1974), p. 292.
22. H. Matsumoto, Sur les sous-groupes arithmétiques des groupes semi-simples déployés. – *Ann. Sci. Ecole Norm. Sup.*, $4^{\text{ème}}$ *sér.*, **2**, 1–62 (1969).
23. K. Mizuno, The conjugate classes of Chevalley groups of type E_6. – *J. Fac. Sci. Univ. Tokyo*, **24**, No. 3, 525–563 (1977).
24. K. Mizuno, The conjugate classes of unipotent elements of the Chevalley groups $\mathrm{E}E_7$ and $\mathrm{E}E_8$. – *Tokyo J. Math.*, **3**, No. 2, 391–458 (1980).

25. R. Ree, A family of simple groups associated with the simple Lie algebras of type (F_4). – *Amer. J. Math.*, **83**, 401–420 (1961).
26. R. Ree, A family of simple groups associated with the simple Lie algebras of type $\mathrm{E}G_2$). – *Amer. J. Math.*, **83**, 432–462 (1961).
27. G. Segal, Unitary representations of some infinite dimensional groups. – *Comm. Math. Phys.*, **80**, 301–342 (1981).
28. T. Shoji, The conjugacy classes of Chevalley groups of type (F_4) over finite fields of characteristic $p \neq 2$. – *J. Fac. Sci. Univ. Tokyo*, **21**, No. 1, 1–17 (1974).
29. S. Splitthoff, Finite presentability of Steinberg groups and related Chevalley groups. – *Contemp. Math.*, **55**, Part. II, 635–687 (1986).
30. T. A. Springer, *Linear algebraic groups. 2nd ed.* – Birkhäuser: Boston et al. (1981), p. 320.
31. M. R. Stein, Stability theorems for K_1, K_2 and related functors modeled on Chevalley groups. – *Japan J. Math.*, **4**, No. 1, 77–108 (1978).
32. R. Steinberg, *Lectures on Chevalley groups.* – Yale University (1968), p. 277.
33. M. Ch. Tamburini, J. S. Wilson, N. Gavioli, The (2,3)–generation of some classical groups. I, II. – *J. Algebra*, **168**, 353–370 (1994); *ibid.*, **176**, 667–680 (1995).
34. D. M. Testerman, A construction of certain maximal subgroups of the algebraic groups $\mathrm{E}E_6$ and $\mathrm{F}F_4$. – *Comm. Algebra*, **17**, No. 4, 1003–1016 (1989).
35. J. Tits, Sur les constantes de structure et le théorème d'existence des algèbres de Lie semi-simples. – *Publ. Math. Inst. Hautes Et. Sci.*, **31**, 21–58 (1966).
36. N. A. Vavilov, Structure of Chevalley groups over commutative rings. – *Proc. Conf. Non-associative algebras and related topics (Hiroshima – 1990)*, World Sci. Publ.: London et al. (1991), 219–335.
37. N. A. Vavilov, E. B. Plotkin, Structure of Chevalley groups over commutative rings. I. Elementary calculations. – *Acta Applicandae Math.*, **45**, No. 1, 73–113 (1996) (Fortsetzung folgt).
38. N. A. Vavilov, E. B. Plotkin, A. V. Stepanov, Calculations in Chevalley groups over commutative rings. – *Sov. Math. Dokl.*, **40**, No. 1, 145–147 (1990).
39. E. B. Plotkin, A. A. Semenov, N. A. Vavilov, Visual basic representations: an atlas. – *Int. J. Algebra and Computations*, **8**, No. 1, 61–97 (1998).
40. A. V. Stepanov, N. A. Vavilov, Decomposition of transvections: a theme with variations. – K-*theory*, **19**, No. 1, 109–153 (2000).
41. N. A. Vavilov, A third look at weight diagrams. – *Rend. Sem. Math. Univ. Padova*, **104**, No. 1, 201–250 (2000).
42. N. A. Vavilov, Can one see the signs of structure constants? – *submitted to St. Petersburg J. Math.*
43. S. Wolfram, *Mathematica: A system for doing mathematics by computer.* – Addison Wesley, Reading et al. (1988), p. 749.

Department of Mathematics and Mechanics
University of Sanct Petersburg
Petrodvorets, 198904, Russia

Поступило 5 июня 2001 г.