Symmetric Hadamard matrices of order 36

Citation for published version (APA):
Bussemaker, F. C., & Seidel, J. J. (1970). *Symmetric Hadamard matrices of order 36*. (EUT report. WSK, Dept. of Mathematics and Computing Science; Vol. 70-WSK-02). Technische Hogeschool Eindhoven.

Document status and date:
Published: 01/01/1970

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 23. Aug. 2022

TECHNISCHE HOGESCHOOL EINDHOVEN
NEDERLAND
ONDERAFDELING DER WISKUNDE

TECHNOLOGICAL UNIVERSITY EINDHOVEN
THE NETHERLANDS
DEPARTMENT OF MATHEMATICS

BIBLIOTHEEK
8 606379
T.H.EINDHOVEN

SYMMETRIC HADAMARD MATRICES OF ORDER 36

by

F.C. Bussemaker and J.J. Seidel

T.H.-Report 70-WSK-02

July 1970

C O N T E N T S

CHAPTER I

SYMMETRIC HADAMARD MATRICES OF ORDER 36, THEORETICAL PART

Symmetric Hadamard matrices H of order 36 with constant diagonal I are obtained from the 12 Latin squares on 6 symbols, and from the 80 Steiner triple systems on 15 symbols. By use of the switching operation (which is generated by multiplying the i-th row and i-th column by -1) these matrices are compared to one another, and new ones are obtained.

All H, except for one pair of Latin square type, appear to be pairwise nonequivalent with respect to switching. Among the 80 Hadamard matrices of Steiner type exactly 23 can be switched into regular Hadamard matrices with HJ = - 6J. All 80 can be switched into regular H with HJ = 6J, some of them in several ways. Among the equivalent regular symmetric Hadamard matrices thus obtained from the lines of PG(3,2), there are two which are closely related to rank 3 graphs. The geometry of PG(3,2) is represented in terms of the Steiner system (24,8,5).

CHAPTER II

GENERAL ALGOL PROCEDURES ON THE EQUIVALENCE OF GRAPHS AND ON RANK 3 GRAPHS

Two procedures, which are not restricted to the special graphs of the present report, are described separately. The procedure Cliquelist serves to distinguish between graphs which are not equivalent with respect to switching. The procedure Alphabeta determines certain invariants for rank 3 graphs. These procedures, which essentially consist in counting void and complete subgraphs of a graph, are explained and coded in ALGOL.

CHAPTER III

TABLES OF STEINER TRIPLE SYSTEMS OF ORDER 15 AND OF LATIN SQUARES OF ORDER 6

The 80 Steiner triple systems of order 15 are taken from the tables by White, Cole, and Cummings; some errors are corrected.

The tables of Fisher and Yates list the 17 Latin squares of order 6. We reproduce 12 of these, which correspond to the 12 non-isomorphic Latin square graphs of order 6.

CHAPTER IV

ALGOL PROGRAMS

The program Equivalence serves to investigate the mutual equivalences of the 80 + 12 Steiner graphs and Latin square graphs. The program Subsystem tests the Steiner triple systems on 15 symbols for the existence of sub-systems of 15 triples in which every symbol occurs three times.

CHAPTER V

NUMERICAL RESULTS

The numerical results obtained by carrying out the programs discussed in Chapter IV are collected in six tables.

CHAPTER I

SYMMETRIC HADAMARD MATRICES OF ORDER 36, THEORETICAL PART *)

1. Introduction

A square matrix H is a Hadamard matrix if its elements are ± 1, and if it is orthogonal. If in addition, H is symmetric, has a constant diagonal I, and has order 36, then we have

$$H^2 = 36I \ , \quad H = H^T \ , \quad H = A + I \ , \quad (A - 5I)(A + 7I) = 0 \ .$$

The symmetric matrix A has 0 on the diagonal and ± 1 elsewhere. The matrix A can be interpreted as the adjacency matrix of a graph on 36 vertices, by defining $\{i,j\}$ to be an edge whenever $a_{ij} = -1$. The resulting graph need not be regular. However, sometimes this graph can be made regular (hence strongly regular [2]) by use of "switching". In terms of the matrix A "switching" amounts to multiplication by -1 of certain rows and of the corresponding columns, cf. [10], [12], [13].

In section 2 of this chapter, symmetric Hadamard matrices of order 36 are obtained from the 12 Latin squares of order 6, and from the 80 Steiner triple systems of order 15. In section 3 the graphs obtained in this way are shown to be non-equivalent with respect to switching, with one exceptional pair. The method consists of counting void and complete subgraphs, and is capable of wider application. Section 4 investigates how graphs of the Steiner type can be switched into strongly regular graphs. Among these graphs there are 23 which can be made regular with valency 21, whereas all 80 can be made regular with valency 15. This follows from a computer search by which each of the 80 Steiner triple systems appears to contain a 3-factor, that is, a subsystem of 15 triples in which every symbol occurs exactly three times.

The lines of the projective geometry PG(3,2) are investigated in section 5. To that end a representation of PG(3,2) in terms of the Steiner system (24,8,5), suggested by Hughes [9], is developed. PG(3,2) is shown to contain five non-isomorphic 3-factors. They lead to three non-isomorphic

*) This Chapter is published separately ([4]) in the Proceedings of the International Conference on Combinatorial Mathematics, New York Academy of Sciences, April 1970.

strongly regular graphs of valency 15. In addition, PG(3,2) yields a strongly regular graph of valency 21. In section 6, the latter and one of the former turn out to be equivalent rank 3 graphs.

Some of the present results were announced in [6], which contains a section on symmetric Hadamard matrices of an order not restricted to 36.

2. Latin square graphs and Steiner graphs

A Latin square of order 6 consists of 36 ordered triples selected from 6 symbols such that for each pair of coordinates every pair of symbols occurs exactly once. The *Latin square graph* [2] belonging to a Latin square of order 6 has as its vertices the 36 triples of that Latin square, and any two vertices are adjacent if and only if the corresponding triples have one symbol in common. This graph is a strongly regular graph whose adjacency matrix A satisfies

$$v = 36 \ , \quad (A - 5I)(A + 7I) = 0 \ , \quad AJ = 5J \ .$$

Therefore, the matrix $H = A + I$ is a regular symmetric Hadamard matrix with a constant diagonal I. Since there are 12 non-isomorphic Latin squares of order 6 (see [5]), we have 12 non-isomorphic Hadamard matrices of the Latin square type.

A Steiner triple system of order 15 consists of 35 unordered triples selected from 15 symbols such that every unordered pair of symbols occurs in exactly one triple. The 35 triples are taken as the vertices of a graph, any two vertices being adjacent if and only if the corresponding triples have one symbol in common. This graph is a strongly regular graph whose adjacency matrix S satisfies

$$v = 35 \ , \quad (S - 5I)(S + 7I) = - J \ , \quad SJ = - 2J \ .$$

We add to this graph an isolated vertex, thus obtaining a graph on 36 vertices which is called a *Steiner graph*. The adjacency matrix A of a Steiner graph satisfies

$$v = 36 \ , \quad (A - 5I)(A + 7I) = 0 \ .$$

The matrix $H = A + I$ is a symmetric Hadamard matrix with a constant diagonal I. Since there are 80 non-isomorphic Steiner triple systems of order 15, see [15], we have 80 non-isomorphic Hadamard matrices of the Steiner type. Steiner graphs are strong graphs (defined in [12]), but they are not regular. Indeed, the isolated vertex has no adjacencies, whereas it follows from $SJ = - 2J$ that each other vertex has 18 adjacencies.

3. Equivalence under switching

Let x be any vertex of any graph on v vertices. *Switching with respect to* x (also called complementation with respect to x, cf. [10], [12], [13]) is defined to be the following operation: cancel all existing adjacencies to x and add all non-existing adjacencies to x. In the adjacency matrix of the graph the effect of switching with respect to x is that the row and the column corresponding to x are multiplied by -1. The operations of switching with respect to any number of vertices generate an equivalence relation on the set of all graphs on v vertices.

The 92 Latin square graphs and Steiner graphs of order 36 defined above, are non-isomorphic in pairs. Now we turn to the question of whether these graphs are pairwise non-equivalent under switching. The answer to this question is affirmative, with only one exception which is the following pair of equivalent Latin square graphs.

Theorem 3.1. The Latin square graphs which correspond to the Latin squares

a	b	c	d	e	f
b	a	d	c	f	e
c	d	e	f	a	b
d	c	f	e	b	a
e	f	a	b	c	d
f	e	b	a	d	c

and

b	a	c	d	e	f
a	b	d	c	f	e
c	d	e	f	a	b
d	c	f	e	b	a
e	f	a	b	c	d
f	e	b	a	d	c

are equivalent with respect to switching.

Proof. The Latin square graph which corresponds to the first Latin square is indicated in the first of the following three squares:

a_{11}	b_{12}	c_{13}	d_{14}	e_{15}	f_{16}
b_{21}	a_{22}	d_{23}	c_{24}	f_{25}	e_{26}
c_{31}	d_{32}	e_{33}	f_{34}	a_{35}	b_{36}
d_{41}	c_{42}	f_{43}	e_{44}	b_{45}	a_{46}
e_{51}	f_{52}	a_{53}	b_{54}	c_{55}	d_{56}
f_{61}	e_{62}	b_{63}	a_{64}	d_{65}	c_{66}

a_{22}	b_{21}	c_{13}	d_{14}	e_{15}	f_{16}
b_{12}	a_{11}	d_{23}	c_{24}	f_{25}	e_{26}
c_{31}	d_{32}	e_{44}	f_{43}	a_{35}	b_{36}
d_{41}	c_{42}	f_{34}	e_{33}	b_{45}	a_{46}
e_{51}	f_{52}	a_{53}	b_{54}	c_{66}	d_{65}
f_{61}	e_{62}	b_{63}	a_{64}	d_{56}	c_{55}

a	b	c	d	e	f
b	a	d	c	f	e
c	d	f	e	b	a
d	c	e	f	a	b
e	f	b	a	d	c
f	e	a	b	c	d

In the first square we switch with respect to the following twelve vertices:

$$a_{11}, b_{12}, b_{21}, a_{22}, e_{33}, f_{34}, f_{43}, e_{44}, c_{55}, d_{56}, d_{65}, c_{66} .$$

What we obtain is the third of these squares. In the middle one, only the horizontal and the vertical adjacencies have been taken care of. In the third square also the letter adjacencies have been made correct by interchanging e and f, a and b, c and d at 16 entries, as indicated. Finally, if in the third square the rows 1 and 2, the columns 3 and 4, and the columns 5 and 6 are interchanged, then the desired Latin square is obtained.

In order to investigate the equivalence of the other pairs of graphs, we use the following general method. Let G be any graph with $|G|$ vertices. Any vertex x of G is made an isolated vertex by switching with respect to the vertices which are adjacent to x. Then x is deleted in order to obtain the graph G_x. In G_x we calculate the number $m(x,\nu)$ of void subgraphs of order ν, $\nu \geq 3$, and the number $n(x,\nu)$ of complete subgraphs of order ν. Finally, given any integers m and n, we determine the number $M(m,\nu)$ of the vertices x such that $m(x,\nu) = m$, and the number $N(n,\nu)$ of the vertices x such that $n(x,\nu) = n$. Clearly, for each ν these numbers satisfy

$$\sum_{m=0}^{\infty} M(m,\nu) = \sum_{n=0}^{\infty} N(n,\nu) = |G| .$$

Now the following necessary conditions for the equivalence of graphs hold:

Theorem 3.2. If two graphs G and $\bar{G}$ are equivalent under switching, then, for each m, for each n, and for each ν, they have the same numbers $M(m,\nu)$ and $N(n,\nu)$.

Proof. Let $\mathcal{D}$ be the class of all diagonal matrices with elements ± 1. For the adjacency matrices A and $\bar{A}$ of the equivalent graphs G and $\bar{G}$ we have

$$\bar{A} = DAD \; ; \quad D \in \mathcal{D} .$$

By switching the elements of the x-th row and of the x-th column of A into +1, and by doing the same for $\bar{A}$, we obtain

$$A_x = EAE \; , \quad \bar{A}_x = F\bar{A}F \; ; \quad E,F \in \mathcal{D} .$$

Therefore, we have

$$\bar{A}_x = FDEA_xEDF = A_x \; ,$$

so G_x and $\bar{G}_x$ are isomorphic. This implies the equality for G and $\bar{G}$ of the numbers mentioned in the theorem.

Remark. It would be interesting to know whether the converse of this theorem is true. This question is related to Ulam's problem, cf. [14], p. 29.

Theorem 3.3. The 12 + 80 Latin square graphs and Steiner graphs of section 2 are pairwise non-equivalent with respect to switching, with the exception of the pair of Latin square graphs of Theorem 3.1.

Proof. For each graph we compute the series of numbers $M(m,5)$ and, if necessary, the series $M(m,4)$, $m = 0,1,2,3,\ldots$. It turns out that all graphs have different series, except for the graphs mentioned in Theorem 3.1. Therefore, by application of Theorem 3.2, they are all non-equivalent.

Remark. Full details of the calculations may be found in Chapters II-V.

4. Regular Steiner graphs

All Latin square graphs are regular, but none of the Steiner graphs are. We now investigate the question of whether any Steiner graph can be switched into a regular graph, that is, whether the equivalence class of any Steiner graph contains strongly regular graphs. Suppose this is possible by switching with respect to any x vertices. The vertices of the Steiner graph are rearranged so as to obtain the following adjacency matrices A before, and $\bar{A}$ after switching:

$$A = \begin{bmatrix} 0 & j^T & j^T \\ j & S_{11} & S_{12} \\ j & S_{21} & S_{22} \end{bmatrix}, \quad \bar{A} = \begin{bmatrix} 0 & -j^T & j^T \\ -j & S_{11} & -S_{12} \\ j & -S_{21} & S_{22} \end{bmatrix}.$$

For regular $\bar{A}$ there are two possibilities, namely $\bar{A}J = 5J$ and $\bar{A}J = -7J$; that is, $x = 15$ and $x = 21$.

The point-block (1,0) incidence matrix N of any Steiner system satisfies:

$$NN^T = 6I + J\,, \quad JN = 3J\,, \quad NJ = 7J\,, \quad N^TN = 3I + \tfrac{1}{2}(J - I - S)\,,$$

$$(S - 5I)(S + 7I) = -J\,, \quad SJ = -2J\,.$$

In either case we write accordingly:

$$N = [N_1 \quad N_2]\,,$$

then

$N_1^TN_1 = 3I + \frac{1}{2}(J - I - S_{11})$, $N_1^TN_2 = \frac{1}{2}(J - S_{12})$, $N_2^TN_2 = 3I + \frac{1}{2}(J - I - S_{22})$,

$j^TN_1 = 3j^T$, $j^TN_2 = 3j^T$, $N_1j + N_2j = 7j$, $S_{11}j + S_{12}j = -2j$, $S_{21}j + S_{22}j = -2j$.

The switching requirements $\bar{A}j = 5j$ and $\bar{A}j = -7j$ yield

$$S_{11}j - S_{12}j = 6j, \quad \text{and} \quad S_{11}j - S_{12}j = -6j,$$
$$-S_{21}j + S_{22}j = 4j, \qquad -S_{21}j + S_{22}j = -8j,$$

respectively. Therefore, in the two cases $x = 15$ and $x = 21$ we have

$$S_{11}j = 2j, \quad S_{12}j = -4j, \quad \text{and} \quad S_{11}j = -4j, \quad S_{12}j = 2j,$$
$$S_{21}j = -3j, \quad S_{22}j = j, \qquad S_{21}j = 3j, \quad S_{22}j = -5j,$$
$$N_1^TN_1j = 9j, \quad N_1^TN_2j = 12j, \qquad N_1^TN_1j = 15j, \quad N_1^TN_2j = 6j,$$
$$N_2^TN_1j = 9j, \quad N_2^TN_2j = 12j, \qquad N_2^TN_1j = 9j, \quad N_2^TN_2j = 12j,$$

respectively. Since N^T has rank 15 it follows from the last equations that in either case, given N, there exists a unique N_1j. In addition, this vector satisfies

$$(N_1j)^T(N_1j) = 135, \quad j^TN_1j = 45 \quad \text{and} \quad (N_1j)^T(N_1j) = 315, \quad j^TN_1j = 63,$$

respectively. Therefore, we have

$$N_1j = (3\ 3\ 3\ 3\ 3\ 3\ 3\ 3\ 3\ 3\ 3\ 3\ 3\ 3\ 3)^T \text{ and } N_1j = (0\ 3\ 3\ 3\ 3\ 3\ 3\ 3\ 6\ 6\ 6\ 6\ 6\ 6\ 6)^T,$$

respectively. The first case leads to the following theorem.

Theorem 4.1. Each Steiner graph can be switched into a regular graph of valency 15.

Proof. The investigations above show that any Steiner graph may be switched into a regular graph of valency 15 if and only if the corresponding Steiner system contains a subsystem of 15 triples, in which every symbol occurs exactly 3 times. Each of the 80 Steiner systems indeed contains such a subsystem; this was shown by a computer search, cf. Chapter V where for each Steiner system a subsystem is given.

Remark. It would be interesting to prove this statement without using a computer.

Theorem 4.2. There are 23 Steiner graphs which can be switched into a regular graph of valency 21.

Proof. It follows from the tables of White, Cole, Cummings ([15], and Chapter III) that there are 23 Steiner systems which contain a projective plane PG(2,2); that is, a subsystem of 7 triples in which 7 symbols occur exactly 3 times. We shall prove that any Steiner graph can be switched into a regular graph of valency 21 if and only if the corresponding Steiner system contains a PG(2,2).

Suppose the 15×21 matrix N_1 and the 15×14 matrix N_2 satisfy

$$N_1 j = (0\ 3\ 3\ 3\ 3\ 3\ 3\ 3\ 6\ 6\ 6\ 6\ 6\ 6\ 6)^T,\ N_2 j = (7\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 1\ 1\ 1\ 1\ 1\ 1\ 1)^T .$$

Accordingly, we write

$$N = [N_1 \mid N_2] = \begin{bmatrix} 0^T & j^T & 0^T \\ N_{12} & N_{21} & N_{22} \\ N_{13} & N_{23} & N_{24} \end{bmatrix},$$

with 7×21 submatrices N_{12}, N_{13}, and 7×7 submatrices N_{21}, N_{22}, N_{23}, N_{24}. From

$$N_{21}j + N_{22}j = 4j,\ N_{23}j + N_{24}j = j,\ N_2^T N_2 j = 12j,\ N_2^T j = 3j$$

we deduce that

$$N_{21}^T j = j\ ,\quad N_{22}^T j = 3j\ ,\quad N_{23}^T j = j\ ,\quad N_{24}^T j = 0\ .$$

Hence, after suitable rearrangement of the rows and the columns of N, we have

$$N_{24} = 0\ ,\quad N_{21} = I\ ,\quad N_{23} = I\ .$$

We now concentrate on N_{22}. It follows from the equations for N that

$$N_{13}N_{13}^T = 5I + J\ ,\quad N_{13}j = 6j\ ,\quad (j^T N_{13} - 2j^T)(N_{13}^T j - 2j) = 0\ ,$$

whence

$$j^T N_{13} = 2j^T\ ,\quad j^T N_{12} = j^T\ ,\quad N_{12}N_{12}^T = 3I\ ,$$

and

$$N_{22}N_{22}^T = 2I + J \ , \quad j^T N_{22} = 3j^T \ .$$

So N_{22} is the incidence matrix of a PG(2,2), which proves the first part of the assertion.

Conversely, suppose any Steiner system containing PG(2,2) as a subsystem is given. Consider the 7 triples of PG(2,2), and the 7 triples through any point not belonging to PG(2,2). If the corresponding Steiner graph is switched with respect to the remaining 21 vertices, then a regular graph of valency 21 is obtained. Now the theorem is proved.

Remark. From the Theorems 4.1 and 4.2 we have at least 80 + 23 regular symmetric Hadamard matrices of order 36 which are not of Latin square type. Probably the number of such matrices is much higher. Indeed, many Steiner systems contain several subsystems of the same kind which yield non-isomorphic Hadamard matrices. This is illustrated for a special Steiner system in section 5.

Remark. In a different terminology, we have found 80 pseudo Latin square graphs [2] by Theorem 4.1, and 23 negative Latin square graphs [11] by Theorem 4.2, all of order 36.

Remark. Instead of dealing only with Steiner graphs, we can also switch Latin square graphs so as to obtain different strongly regular graphs. Without going into details we remark that each Latin square graph can be switched into a negative Latin square graph.

Remark. The case of block designs with

$$v = k(2k-1) \ , \quad b = 4k^2 - 1 \ , \quad k = k \ , \quad r = 2k+1 \ , \quad \lambda = 1 \ ,$$

of which the present Steiner systems are the specializations for k = 3, may be treated in an analogous way. This will be worked out elsewhere. There are relations to the paper [3] by Bose and Shrikhande.

5. The lines of PG(3,2)

One of the Steiner systems on 15 symbols is provided by the points and the lines of the projective space PG(3,2) over GF(2). Pursuing the investigations of section 4 for this special system, we are interested in finding the subsystems of 15 lines in which every point occurs exactly 3 times. To that end we shall use a representation of PG(3,2) in terms of the Steiner system (24,8,5), which was suggested to us by Hughes [9]; see also Berlekamp [1].

A *Steiner system* S(24,8,5) is a set of 24 elements and a collection of 8-subsets, called blocks, such that every 5-subset is contained in one block. It is well-known that there exists a unique S(24,8,5), that its automorphism group is the Mathieu group M_{24}, that it can be viewed as the set of vectors of weight 8 in the Golay code (24,12), and that any pair of its blocks intersects in 4, in 2, or in no elements. In fact (cf. Lemma 5.1 of [6]), any block intersects 280 blocks in 4 elements, 448 blocks in 2 elements, and 30 blocks in no elements.

We fix one element, called the *origin* 0, and one block, called *infinity* ∞, where 0 and ∞ are non-incident. The 24 - 1 - 8 = 15 remaining elements of S(24,8,5) are called the *points*. The blocks on 0 which have void intersection with ∞ are called the *planes*; each plane contains 7 points. The blocks on 0 which intersect ∞ in 4 elements are called the *halflines*; each halfline contains 3 points. On 3 such points and the origin there is another halfline, which intersects ∞ in the complementary quadruple. The pair of two such halflines is called a *line*. By a check on the axioms we have:

Theorem 5.1. The points, lines and planes defined above constitute the projective geometry PG(3,2).

Any two lines having exactly 2 elements of ∞ in common intersect in a point. Indeed, the pair of blocks containing these 2 elements and the origin, must have one further point in common. Any two lines which have 1 or 3 elements of ∞ in common, are skew. A *spread* of lines, that is, a set of 5 mutually skew lines, is represented by a triple of elements of ∞. Indeed, the 5 quadruples of ∞ containing any given triple yield 5 skew lines. Conversely, normalizing the 5 lines of any spread so as to have an element of ∞ in common, we observe that they all must have 3 elements of ∞ in common. So we have:

Theorem 5.2. The 56 spreads of PG(3,2) are represented by the unordered triples out of the 8 elements of ∞.

The system of the 10 lines which are on $a \in \infty$, on $b \in \infty$, but not on $c \in \infty$, is denoted by ab(c), and is called a *Petersen system*. Indeed, the lines form a Petersen graph, if adjacency is defined by intersection. In addition, each point of PG(3,2) is contained in exactly 2 lines of the system. We now have Singer's representation:

Theorem 5.3. The 35 lines of PG(3,2) are given by abc, ab(c), a(b)c, (a)bc, with $a,b,c \in \infty$.

In fact, the lines of PG(3,2) may be arranged in such a way that their adjacency matrix S reads as follows:

$$S = \begin{bmatrix} J-I & C-I & -C-I & C-I & -C-I & C-I & -C-I \\ C-I & -C & J-2I & C+I & -J+2I & C+I & -J+2I \\ -C-I & J-2I & C & -J+2I & -C+I & -J+2I & -C+I \\ C-I & C+I & -J+2I & -C & J-2I & C+I & -J+2I \\ -C-I & -J+2I & -C+I & J-2I & C & -J+2I & -C+I \\ C-I & C+I & -J+2I & C+I & -J+2I & -C & J-2I \\ -C-I & -J+2I & -C+I & -J+2I & -C+I & J-2I & C \end{bmatrix}$$

where

$$C = \begin{bmatrix} 0 & - & + & + & - \\ - & 0 & - & + & + \\ + & - & 0 & - & + \\ + & + & - & 0 & - \\ - & + & + & - & 0 \end{bmatrix}.$$

From this 35-matrix S regular Hadamard matrices of order 36 are obtained by application of Theorems 4.2 and 4.1.

Let P en PG(2,2) denote any non-incident pair of a point and a plane of PG(3,2). Switching with respect to all lines, except for the 7 on P and the 7 of PG(2,2), yields a strongly regular graph of valency 21. Obviously, only one such graph is obtained in this way. This graph has certain special properties. It is the graph III which appears in section 6.

Strongly regular graphs of valency 15 may be obtained in more than one way. According to the proof of Theorem 4.1 we have to find systems of 15 lines in PG(3,2) such that each of the 15 points is on 3 lines. Such a system is called a 3-*factor*. Here are five different types of 3-factors of PG(3,2), presented in terms of the elements of ∞ = {a,b,c,d,e,f,g,h}, by use of the notation introduced after Theorem 5.2.

Type 1: ab(c) and abc. This 3-factor is composed of a Petersen system and a spread. Its graph is the complement of the triangular graph on 6 symbols. It contains 6 spreads altogether.

Type 2: ab(c) and bcd. Again, this 3-factor consists of a Petersen system and a spread; however, there is only one additional spread abd.

Type 3: abc, ade, afg. In this case there are three spreads.

Type 4: bcf, acg, abh. In this case we have a different combination of three spreads.

Type 5: xyde, xyef, xyfg, xygh, xyhd, with x,y $\in$ {a,b,c}. Here we have a combination of three 5-cycles. This 3-factor contains no spreads. Its graph represents a triangular tessellation of the torus into 30 triangles.

For the graph of the 3-factor of each type we compute the total number of triangles. These numbers are 15, 27, 23, 28, 30, respectively. This implies that the five 3-factors are non-isomorphic.

Now we are in a position to apply the switching process with respect to any of these five subgraphs of the Steiner graph belonging to PG(3,2), thus obtaining strongly regular graphs of order 36 and valency 15. For each of these strongly regular graphs we compute the number $m(\nu)$ of void subgraphs of order ν, and the number $n(\nu)$ of complete subgraphs of order ν, $\nu = 3,4,5,6$.

	type 1	type 2	type 3	type 4	type 5
m(3)	1200	1200	1200	1200	1200
m(4)	1080	1176	1176	1200	1200
m(5)	216	312	312	336	336
m(6)	0	32	32	41	41
m(7)	0	0	0	0	0
n(3)	540	540	540	540	540
n(4)	135	231	231	255	255
n(5)	0	0	0	0	0

From these data we observe that the automorphism group of the strongly regular graph obtained from any subgraph of type 2, 3, 4, or 5, cannot be transitive. Indeed, suppose such a graph is transitive, then each vertex would be contained in equally many void subgraphs of order 6. Since there are 36 vertices, the number m(6) would be divisible by 6, which is not the case. This explains why we only obtain three, and not five non-isomorphic strongly regular graphs of order 36 and valency 15 from the five types of non-isomorphic subgraphs of order 15. More precisely, we announce the following theorem (whose proof will appear elsewhere) and conjecture:

Theorem 5.4. The 3-factors of types 1, 2, 3, 4, 5 described above, are the only non-isomorphic 3-factors of PG(3,2).

Conjecture. There are exactly three non-isomorphic strongly regular graphs of order 36 and valency 15 which are equivalent to the Steiner graph belonging to PG(3,2).

The graph obtained from the 3-factor of type 1 has special properties. It is the graph II which appears in section 6.

6. Rank 3 graphs of order 36

A transitive permutation group G on a set Ω is said to be *of rank* 3 if the number of orbits of any G_x equals 3 (see [8]); here G_x denotes the stabilizer of $x \in \Omega$. Let G be a rank 3 group of even order. For any $x \in \Omega$ let G_x have the orbits $\{x\}$, $\Delta(x)$, $\Gamma(x)$. The notation is chosen so that

$$\Delta(g(x)) = g(\Delta(x)) \ , \quad \Gamma(g(x)) = g(\Gamma(x)) \quad \text{for all } x \in \Omega,\ g \in G \ .$$

Then we have $y \in \Delta(x)$ if and only if $x \in \Delta(y)$. Two *rank* 3 *graphs* belonging to G are defined as follows. Both graphs have the vertex set Ω. For the first graph adjacency of any $x,y \in \Omega$ is defined iff $y \in \Delta(x)$, for the other graph iff $y \in \Gamma(x)$; so the graphs are complementary.

It is well known that rank 3 graphs are strongly regular (see [8]). They have additional properties which general strongly regular graphs fail to satisfy. For any non-adjacent $x,y \in \Omega$ let $\alpha(\nu)$ be the number of distinct void subgraphs of order ν which contain the vertices x and y. For any adjacent $u,v \in \Omega$ let $\beta(\nu)$ be the number of distinct complete subgraphs of order ν which contain u and v. For rank 3 graphs the numbers $\alpha(\nu)$ and $\beta(\nu)$ are independent of the choice of $x,y \in \Omega$, and $u,v \in \Omega$, respectively, for all integer $\nu \geq 2$ (see [7]). For strongly regular graphs this property is guarantied only for $\alpha(3) = p^2_{22}$ and $\beta(3) = p^1_{11}$.

Hestenes, Higman, and Sims [7] have characterized several rank 3 graphs. It follows from their work that there is a unique rank 3 graph belonging to each of the following sets of order v and eigenvalues ρ_0, ρ_1, ρ_2:

Graph I: $v = 35$, $\rho_0 = -2$, $\rho_1 = 5$, $\rho_2 = -7$.

The group is the linear fractional group L(4,2) of order $\frac{1}{2}8!$, acting on the lines of PG(3,2). The parameters are as follows: the valency is 18, and

$\alpha(3) = 6$, $\alpha(4) = 6$, $\alpha(5) = 2$, $\alpha(6) = 0$, $\beta(3) = 9$, $\beta(4) = 20$,

$\beta(5) = 20$, $\beta(6) = 10$, $\beta(7) = 2$, $\beta(8) = 0$.

Graph II: $v = 36$, $\rho_0 = 5$, $\rho_1 = 5$, $\rho_2 = -7$.

The group is the orthogonal group O(6,−1,2) of order $2^6(2^3 + 1)(2^4 - 1)(2^2 - 1)$, acting on the non-singular points of PG(5,2). The parameters are as follows: the valency is 15, and

$\alpha(3) = 10$, $\alpha(4) = 18$, $\alpha(5) = 6$, $\alpha(6) = 0$, $\beta(3) = 6$, $\beta(4) = 3$, $\beta(5) = 0$.

Graph III: $v = 36$, $\rho_0 = -7$, $\rho_1 = 5$, $\rho_2 = -7$.

The group is the exceptional Lie group E(2,2) of order $2^6(2^6 - 1)(2^2 - 1)$. The parameters are as follows: the valency is 21, and

$\alpha(3) = 4$, $\alpha(4) = 0$, $\beta(3) = 12$, $\beta(4) = 38$, $\beta(5) = 40$, $\beta(6) = 20$,

$\beta(7) = 4$, $\beta(8) = 0$.

Theorem 6.1. The rank 3 graphs II and III are equivalent, under switching, to the Steiner graph which belongs to the rank 3 graph I.

Proof. The lines of PG(3,2) constitute the unique rank 3 graph I. The corresponding Steiner graph is switched into a strongly regular graph in two ways. At first, switching is performed with respect to the vertices of any subgraph of type 1, as defined in section 5. The resulting graph has the same parameters as the unique rank 3 graph II, hence is isomorphic to graph II. Secondly, switching is performed with respect to the vertices which correspond to all lines except for the 7 lines of any plane and the 7 lines on any point non-incident with the plane. The resulting graph has the same parameters as the unique rank 3 graph III, hence is isomorphic to graph III. These observations prove the theorem. The calculations were performed by use of the procedure Alphabeta which is described in Chapter II.

References

[1] Berlekamp, E.R., Coding theory and the Mathieu groups, to be published.

[2] Bose, R.C., Strongly regular graphs, partial geometries, and partially balanced designs, Pacific J. Math. 13 (1963), 389-419.

[3] Bose, R.C., and S.S. Shrikhande, Graphs in which each pair of vertices is adjacent to the same number d of other vertices, University of North Carolina, Instit. Statist. Mimeo Series 600.6 (1969).

[4] Bussemaker, F.C., and J.J. Seidel, Symmetric Hadamard matrices of order 36, Proc. Intern. Conf. Combin. Math., N.Y. Acad. Sci. (1970)

[5] Fisher, R.A., and F. Yates, Statistical tables, 6th edit., London (1963).

[6] Goethals, J.M., and J.J. Seidel, Strongly regular graphs derived from combinatorial designs, Canad. J. Math., to appear.

[7] Hestenes, M., D.G. Higman, and C.C. Sims, Rank 3 permutation groups of small degree, to be published.

[8] Higman, D.G., Finite permutation groups of rank 3, Math. Zeitschr. 86 (1964), 145-156.

[9] Hughes, D.R., private communication.

[10] Lint, J.H. van, and J.J. Seidel, Equilateral point sets in elliptic geometry, Kon. Ned. Akad. Wetensch. Amst. Proc. A, 69 (= Indag. Math. 28) (1966), 335-348.

[11] Mesner, D.M., A new family of partially balanced incomplete block designs with some Latin square design properties, Ann. Math. Statist. 38 (1967), 571-581.

[12] Seidel, J.J., Strongly regular graphs with (-1,1,0) adjacency matrix having eigenvalue 3, Lin. Algebra and Appl. 1 (1968), 281-298.

[13] Seidel, J.J., Strongly regular graphs, in W.T. Tutte, Recent progress in combinatorics, New York (1969), 185-198.

[14] Ulam, S.M., A collection of mathematical problems, New York (1960).

[15] White, H.S., F.N. Cole, and L.D. Cummings, Complete classification of the triad systems on fifteen elements, Memoir Nat. Acad. Sci. 14 (1925), Second memoir, 1-89.

CHAPTER II

GENERAL ALGOL PROCEDURES ON THE EQUIVALENCE OF GRAPHS AND ON RANK 3 GRAPHS

1. Introduction

The notions of switching and equivalence are defined in Chapter I, section 3. Theorem 3.2 gives necessary conditions for the equivalence of any two graphs in terms of their lists of numbers $M(m,\nu)$ and $N(n,\nu)$. The present chapter deals first with the procedure Cliquelist by which these numbers may be determined. The authors do not know of any example of non-equivalent graphs with the same cliquelist.

In Chapter I, section 6, the notion of a rank 3 graph is defined. The constancy of the numbers $\alpha(\nu)$ and $\beta(\nu)$ provides necessary conditions for a graph to be of rank 3. These numbers may be determined by use of the procedure Alphabeta.

The procedure Cliquelist uses two subroutines which are coded in the procedures Cliquenumber and Isolation. The procedure Alphabeta uses the procedure Betan. For each procedure a description of the parameters, an explanation, and an ALGOL text are given. The ALGOL text of the subprocedures is not given separately.

From now on we write v for ν.

2. integer procedure Cliquenumber (dim, A, clique, v)

The number of complete (void) subgraphs of order v of the graph G of order dim with (-1,1) adjacency matrix A is determined if clique ≡ true (false).

2.1. Formal parameters

integer dim:	< expression >; denotes the order of the symmetric matrix A; is called by value.
integer array A:	array of dimensions [1 : dim, 1 : dim]; contains the elements of the right upper half of the (-1,1) adjacency matrix A of the graph G.
Boolean clique:	< Boolean expression >; if clique ≡ true (false) then the number of complete (void) subgraphs of order v in G is determined; is called by value.

integer v: < expression >; denotes the order of the subgraphs of G, the number of which has to be determined; satisfies $3 \leq v \leq dim$; is called by value.

Cliquenumber: on exit of the procedure the value of the function designator Cliquenumber is the number of complete (void) subgraphs of order v in G.

2.2. Explanation

The procedure is based on the back-tracking algorithm. A general step in this algorithm is as follows (clique ≡ true). Let the vertices of the graph G be enumerated by 1,2,...,dim. Let

$$V_i = \{p_1, p_2, \ldots, p_i\}\ , \quad \text{with} \quad 1 \leq p_1 < p_2 < \ldots < p_i \leq \dim - (v-i)\ ,$$

be the vertex set of a complete subgraph in G of order $i < v$. We try to extend V_i by a vertex k out of the candidate set

$$C_{i+1} = \{p_i+1, p_i+2, \ldots, \dim -v+i+1\}\ .$$

To that end the consecutive vertices of C_{i+1} are tested on the condition $A[p_j,k] = -1$, for all $j \in \{1,2,\ldots,i\}$.

3. procedure Isolation (dim, A, clique, x)

The graph G with (-1,1) adjacency matrix A is transformed into the graph Gxx by switching with respect to all vertices of G which are adjacent to x, and, if clique ≡ false, by switching with respect to x.

3.1. Formal parameters

integer dim: < expression >; denotes the order of A; is called by value.

integer array A: array of dimensions [1 : dim, 1 : dim]; contains the elements of the right upper half of the symmetric matrix A; at call, A is the (-1,1) adjacency matrix of the graph G; on exit of the procedure, A is the (-1,1) adjacency matrix of the graph Gxx.

<u>Boolean</u> clique: < Boolean expression >; if clique ≡ <u>true</u> (<u>false</u>) then, by switching, vertex x is made nonadjacent (adjacent) to all vertices of G; is called by value.

<u>integer</u> x: < expression >; $1 \le x \le dim$; denotes the vertex x of G which is isolated from (joined to) all other vertices of G, so as to form the graph Gxx; is called by value.

3.2. <u>Explanation</u>

Let G be any graph of order dim, let x be any vertex of G, and let clique ≡ <u>true</u>. G is transformed into Gxx by isolation of x, which is performed by one of the following methods:

a. by switching with respect to all vertices of G which are adjacent to x,
b. by switching with respect to x and to all vertices of G which are nonadjacent to x.

The procedure chooses between these methods, taking into account the number of adjacencies of x in G. If clique ≡ <u>false</u> then x has to be joined to all other vertices of G. Again the procedure chooses the method of the minimum number of operations.

4. <u>procedure</u> Cliquelist (dim, A, clique, v, number, n, N)

If clique ≡ <u>true</u> (<u>false</u>) then the number of elements ≠ 0 of N(n,v) (M(m,v)) is determined. These elements, and their indices, are placed in the arrays N, and n, respectively.

4.1. <u>Formal parameters</u>

<u>integer</u> dim: < expression >; denotes the order of the symmetric matrix A; is called by value.

<u>integer</u> <u>array</u> A: array of dimensions [1 : dim, 1 : dim]; contains the elements of the right upper half of the symmetric matrix A; at call, A is the (-1,1) adjacency matrix of any graph G; on exit of the procedure, A is the (-1,1) adjacency matrix of a graph equivalent to G.

Boolean clique: < Boolean expression >; if clique $\equiv$ true (false) then the integer array N(n,v) (M(m,v)) is determined; is called by value.

integer v: < expression >; $3 \leq v \leq dim - 1$; denotes the order of the subgraphs of G; is called by value.

integer number: < variable >; on exit of the procedure, denotes the number of elements $\neq 0$ of the integer array N(n,v) (if clique $\equiv$ true), or of the integer array M(m,v) (if clique $\equiv$ false).

integer array n: array of dimension [1 : number]; on exit of the procedure, contains the indices n of the non-zero elements of the integer array N(n,v) (if clique $\equiv$ true), or the indices m of the non-zero elements of the integer array M(m,v) (if clique $\equiv$ false).

integer array N: array of dimension [1 : number]; on exit of the procedure, contains the elements N(n,v) $\neq 0$ (if clique $\equiv$ true), or the elements M(m,v) $\neq 0$ (if clique $\equiv$ false).

4.2. Explanation

The procedure constructs the graphs Gxx, of order dim, for x = 1,2,...,dim. For clique $\equiv$ true (false), in each Gxx the number n(x,v) of complete (m(x,v) of void) subgraphs of order v is determined. In the set of integers ≥ 0 thus obtained, the maximum and the minimum are determined. The frequency of occurrence of each integer between these bounds is determined. Finally, the integers and their frequencies (if $\neq 0$) are placed in the arrays n and N.

4.3. The ALGOL text

```
procedure Cliquelist(dim,A,clique,v,number,n,N); value dim,clique,v;
integer dim,v,number; Boolean clique; integer array A,n,N;
begin comment Definitions:
      G:        a graph of finite order dim, undirected, without loops,
                without multiple edges, with (-1,1) adjacency matrix A.
      x:        any vertex of G.
      Gx:       the graph obtained from G by switching with respect to
                all vertices of G which are adjacent to x, and then
                deleting x.
      v:        integer ≥ 3.
      m(x,v):   the number of void subgraphs of order v in Gx.
      n(x,v):   the number of complete subgraphs of order v in Gx.
      m,n:      integers ≥ 0.
      M(m,v):   the number of vertices x of G with m(x,v) = m.
      N(n,v):   the number of vertices x of G with n(x,v) = n.
         If clique ≡ true (false) then the number of elements ≠ 0 of
      N(n,v) (M(m,v)) is determined. These elements, and their indices, are
      placed in the arrays N, and n, respectively;
      integer a,i,max,min,x;

      integer procedure Cliquenumber(dim,A,clique,v); value dim,clique,v;
      integer dim,v; Boolean clique; integer array A;
      begin comment The number of complete (void) subgraphs of order v of
            the graph G of order dim with (-1,1) adjacency matrix A is
            determined if clique ≡ true (false);
            integer elt,i,index,j,number,ub; Boolean equal;
            integer array p[1 : v];
            elt := if clique then -1 else 1; number := 0;
            i := 1; p[1] := index := 1; ub := dim - v + 1;
      nexti: i := i + 1; ub := ub + 1;
      nextind: index := index + 1;
            if index > ub then
            begin i := i - 1; if i = 0 then goto ready;
                  index := p[i]; ub := ub - 1; goto nextind
            end;
```

```
        j := 0; equal := true;
        for j := j + 1 while j < i ∧ equal do equal := A[p[j],index] = elt;
        if equal then
        begin if i = v then begin number := number + 1; goto nextind end;
              p[i] := index; goto nexti
        end;
        goto nextind;
ready: Cliquenumber := number
end Cliquenumber;

procedure Isolation(dim,A,clique,x); value dim,clique,x;
integer dim,x; Boolean clique; integer array A;
begin comment The graph G with (-1,1) adjacency matrix A is transformed
        into the graph Gxx by switching with respect to all vertices of
        G which are adjacent (nonadjacent) to x if clique ≡ true (false);
        integer Axi,elt,i,i1,j,sum,x1;
        x1 := x - 1; sum := 0;
        for i := 1 step 1 until x1 do sum := sum + A[i,x];
        for i := x + 1 step 1 until dim do sum := sum + A[x,i];
        elt := -sign(sum); if elt = 0 then elt := if clique then -1 else 1;
        for i := 1 step 1 until x1, x + 1 step 1 until dim do
        begin Axi := if i < x then A[i,x] else A[x,i];
              if Axi = elt then
              begin i1 := i - 1;
                    for j := 1 step 1 until i1 do A[j,i] := -A[j,i];
                    for j := i + 1 step 1 until dim do A[i,j] := -A[i,j]
              end
        end;
        if clique ≡ elt = 1 then
        begin for i := 1 step 1 until x1 do A[i,x] := -A[i,x];
              for i := x + 1 step 1 until dim do A[x,i] := -A[x,i]
        end
end Isolation;
```

```
for x := 1 step 1 until dim do
begin Isolation(dim,A,clique,x);
      a := N[x] := Cliquenumber(dim,A,clique,v);
      if x = 1 then min := max := a else
      if a < min then min := a else if a > max then max := a
end;
begin integer array frequency[min : max];
      for i := min step 1 until max do frequency[i] := 0;
      for i := 1 step 1 until dim do
      begin a := N[i]; frequency[a] := frequency[a] + 1 end;
      number := 0;
      for i := min step 1 until max do
      begin a := frequency[i];
            if a ≠ 0 then
            begin number := number + 1;
                  n[number] := i; N[number] := a
            end
      end
end
end Cliquelist;
```

5. procedure Betan (dim, A, n, condition, betan)

For the graph G of order dim with adjacency matrix A and for any given n, it is checked whether beta(n,u,v), that is, the number of complete subgraphs of order n in G which contain any pair of adjacent vertices u and v of G, is independent of u and v. If this is the case then condition is set true and betan becomes this number, beta(n); otherwise condition is set false.

5.1. Formal parameters

integer dim:	< expression >; denotes the order of the symmetric matrix A; is called by value.
integer array A:	array of dimensions [1 : dim, 1 : dim]; contains the elements of the right upper half of the adjacency matrix A of the graph G.
integer n:	< expression >; n ≥ 3; denotes the order of the subgraphs on which the determination of the numbers beta(n,u,v) depends; is called by value.

Boolean condition: < variable >; on exit of the procedure, denotes whether beta(n,u,v) is independent of the choices of u and v of G or not. If so, condition is set true; otherwise, false.

integer betan: < variable >; on exit of the procedure, denotes the value of beta(n) if condition is true.

5.2. Explanation

The procedure works in principle as follows. Suppose the vertices of the graph are enumerated 1,2,...,dim. To each pair of adjacent vertices, u and v (v > u) of G, there is assigned a number beta(u,v), which denotes the number of complete subgraphs of order n containing u and v. Initially, beta(u,v) = 0 for every pair of adjacent vertices u,v. Each complete subgraph of order n in G is searched for in the same way as in the procedure Cliquenumber. Whenever a complete subgraph is found, the numbers beta(u,v) are increased by one for each pair (u,v) in that subgraph. The procedure always examines collections of vertices $p_1,p_2,\dots,p_i$, with $1 \le p_1 < p_2 < \dots < p_i$ and $i \le n$, for mutual adjacencies; p_1 runs through the numbers 1,2,...,dim − v + 1. When $p_1 > 1$, the numbers beta(u,v) for the pairs (u,v) with $u \in \{1,2,\dots,p_1-1\}$ are independent of further results of the procedure. Each time, just before p_1 is increased by one, the numbers beta(u,v) for $u = p_1$ are checked to see whether they are constant. If so, then call this constant beta(n). Otherwise, condition is set false, and the procedure halts. At the end of the procedure the numbers beta(u,v) with u = dim − v + 2,dim − v + 3,...,dim − 1, respectively, are checked to see whether they are equal to beta(n).

6. procedure Alphabeta (dim, A, condition, alpha, beta, label, w)

The procedure checks in the first place whether the (−1,1) adjacency matrix A of order dim of the graph G is regular. If not, then the program exits to the label label in the main program. If G is regular, the procedure checks the numbers alpha(n,x,y) and beta(m,u,v) whether they are independent of the choices of x,y and u,v, respectively. If so, then Boolean condition is set true, and for n = 3,4,... the arrays alpha and beta are filled up to and including the first index n for which alpha(n) and beta(n), respectively, equal 0. Otherwise condition is set false, and w becomes the smallest integer ≥ 3 for which either alpha(w,x1,y1) ≠ alpha(w,x2,y2) or beta(w,u1,v1) ≠ beta(w,u2,v2).

6.1. Formal parameters

integer dim: < expression >; denotes the order of the symmetric matrix A; is called by value.

integer array A: array of dimensions [1 : dim, 1 : dim]; contains the elements of the right upper half of the (-1,1) adjacency matrix A of the graph G.

Boolean condition: < variable >; on exit of the procedure, denotes whether the numbers alpha and beta are constant.

integer array alpha: array of dimension [3 : dim]; as long as condition is true this array contains on exit of the procedure the numbers alpha(3),alpha(4),..., alpha(m), where m is the smallest integer for which alpha(m) = 0.

integer array beta: array of dimension [3 : dim]; as long as condition is true this array contains on exit of the procedure the numbers beta(3),beta(4),...,beta(m), where m is the smallest integer for which beta(m) = 0.

label label: label (in the main program) to which the procedure branches if G is not regular.

integer w: < variable >; as long as condition is false, w indicates on exit of the procedure the smallest integer for which either alpha(w,x1,y1) $\neq$ $\neq$ alpha(w,x2,y2) or beta(w,u1,v1) $\neq$ beta(w,u2,v2).

6.2. Explanation

This procedure first checks whether G is regular. If not, the procedure branches to the label label in the main program. If G is regular, the numbers beta(n,u,v) are checked whether they are independent of the choices of u and v from G for $n \geq 3$. To that purpose the procedure Betan is used, first for $n = 3$, and after that, as long as condition is true, and as long as beta(n − 1) $\geq$ n − 2, for n = 4,5,... . When beta(n − 1) < n − 2, then beta(n,u,v) = 0 for all adjacent pairs (u,v) of G. (Indeed, a complete subgraph on u and v of order n contains n-2 complete subgraphs on u and v of order n-1. After beta(n,u,v), the numbers alpha(n,x,y) are checked in the same way. For this it suffices to check beta(n,x,y) in the complement of graph G.

6.3. The ALGOL text

```
procedure Alphabeta(dim,A,condition,alpha,beta,label,w); value dim;
integer dim,w; Boolean condition; integer array A,alpha,beta; label label;
begin comment Definitions:
      G:             a graph of finite order dim, undirected, without loops,
                     without multiple edges, with (-1,1) adjacency matrix A.
      x,y:           any pair of nonadjacent vertices of G.
      u,v:           any pair of adjacent vertices of G.
      n:             integer ≥ 2.
      alpha(n,x,y):  the number of void subgraphs of order n in G which
                     contain the vertices x and y.
      beta(n,u,v):   the number of complete subgraphs of order n in G which
                     contain the vertices u and v.
        For rank 3 graphs the numbers alpha(n,x,y) and beta(n,u,v) are
      independent of the choices of x,y respectively u,v of G for every
      n ≥ 2, and equal to alpha(n)  respectively beta(n).
        The procedure checks in the first place whether the (-1,1)
      adjacency matrix A of order dim of the graph G is regular. If this
      is not the case, the program exits to the label label in the main
      program. If G is regular, the procedure checks the numbers alpha(n,x,y)
      and beta(n,u,v) to see whether they are independent of the choices
      of x,y and u,v, respectively. If so, the Boolean condition is set
      true, and for n = 3,4,... the arrays alpha and beta are filled up
      to and including the first index n for which alpha(n) and beta(n).
      respectively, equal 0. Otherwise condition is set false, and w becomes
      the smallest integer ≥ 3 for which either alpha(w,x1,y1) ≠
      alpha(w,x2,y2) or beta(w,u1,v1) ≠ beta(w,u2,v2);
      integer betan,dim1,i,i1,j,n,s,sum;
```

```
procedure Betan(dim,A,n,condition,betan); value dim,n;
integer dim,n,betan; Boolean condition; integer array A;
begin comment For the graph G of order dim with (-1,1) adjacency
      matrix A and for given n, beta(n,u,v) is checked to see whether
      it is independent of the choices of u and v of G. If this is
      the case then condition is set true and betan becomes beta(n),
      otherwise condition is set false;
      integer a,b,dim1,i,index,j,k,n1,ub; Boolean equal,first;
      integer array p[1 : dim],beta[1 : dim,1 : dim];
      dim1 := dim - 1; first := true;
      for i := 1 step 1 until dim1 do
      for j := i + 1 step 1 until dim do beta[i,j] := 0;
      i := 1; p[1] := index := 1; ub := dim - n + 1; n1 := n - 1;
nexti: i := i + 1; ub := ub + 1;
nextind: index := index + 1;
      if index > ub then
      begin i := i - 1; if i = 0 then goto ready;
            index := p[i];
            if i = 1 then
            for j := index + 1 step 1 until dim do
            if A[index,j] = -1 then
            begin if first then
                  begin betan := beta[index,j]; first := false end else
                  if beta[index,j] ≠ betan then
                  begin condition := false; goto End end
            end;
            ub := ub - 1; goto nextind
      end;
      j := 0; equal := true;
      for j := j + 1 while j < i ∧ equal do equal := A[p[j],index] = -1;
      if equal then
      begin p[i] := index;
            if i = n then
            begin for j := 1 step 1 until n1 do
                  begin a := p[j]; for k := j + 1 step 1 until n do
                        begin b := p[k]; beta[a,b] := beta[a,b] + 1 end
                  end;
```

```
                    goto nextind
                end;
                goto nexti
            end;
            goto nextind;
    ready: for i := index step 1 until dim1 do
            for j := i + 1 step 1 until dim do
            if A[i,j] = -1 then
            begin if first then begin betan := beta[i,j]; first := false end else
                if beta[i,j] ≠ betan then
                begin condition := false; goto End end
            end;
            condition := true;
    End:
    end Betan;

    for i := 1 step 1 until dim do
    begin i1 := i - 1; sum := 0;
            for j := 1 step 1 until i1 do sum := sum + A[j,i];
            for j := i + 1 step 1 until dim do sum := sum + A[i,j];
            if i = 1 then s := sum else if s ≠ sum then goto label
    end;
    for n := 3, 4 step 1 until betan + 2 do
    begin Betan(dim,A,n,condition,betan);
            if condition then
            begin a := n + 1; beta[n] := betan end else
            begin w := n; goto Ready end
    end; beta[a] := 0;
    dim1 := dim - 1;
    for i := 1 step 1 until dim1 do
    for j := i + 1 step 1 until dim do A[i,j] := -A[i,j];
    for n := 3, 4 step 1 until betan + 2 do
    begin Betan(dim,A,n,condition,betan);
            if condition then
            begin a := n + 1; alpha[n] := betan end else
            begin w := n; goto Ready end
    end; alpha[a] := 0;
Ready:
end Alphabeta;
```

CHAPTER III

TABLES OF STEINER TRIPLE SYSTEMS OF ORDER 15 AND OF LATIN SQUARES OF ORDER 6

1. Steiner triple systems of order 15

The original source for the 80 Steiner triple systems of order 15 is White, Cole, and Cummings, [15] pp. 77-80. Afterwards, R.A. Fisher refound 79 of them, whereas Hall and Swift confirmed Cole's list by a computer search. The present chapter contains the list of Cole, in extended form, in its original order, and numbered from 1 to 80. Four errors have been corrected, viz.

nr. 1:	for	3	14	14	read	3	13	14 ,
nr. 25:	for	5	8	15	read	5	6	15 ,
nr. 60:	for	84	3	11	read	8	13	14 ,
nr. 66:	for	8	11	15	read	9	11	15 .

In view of Theorem 4.2 it is noted that only the systems nr. 1 through 22, 61 contain a subsystem of 7 triples in which 7 symbols occur three times.

nr. 1

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	7
3	5	6
3	8	11
3	9	10
3	12	15
3	13	14
4	8	12
4	9	13
4	10	14
4	11	15
5	8	13
5	9	12
5	10	15
5	11	14
6	8	14
6	9	15
6	10	12
6	11	13
7	8	15
7	9	14
7	10	13
7	11	12

nr. 2

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	7
3	5	6
3	8	11
3	9	10
3	12	15
3	13	14
4	8	12
4	9	13
4	10	14
4	11	15
5	8	13
5	9	12
5	10	15
5	11	14
6	8	15
6	9	14
6	10	13
6	11	12
7	8	14
7	9	15
7	10	12
7	11	13

nr. 3

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	7
3	5	6
3	8	11
3	9	10
3	12	15
3	13	14
4	8	12
4	9	13
4	10	14
4	11	15
5	8	14
5	9	15
5	10	12
5	11	13
6	8	15
6	9	14
6	10	13
6	11	12
7	8	13
7	9	12
7	10	15
7	11	14

nr. 4

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	7
3	5	6
3	8	11
3	9	10
3	12	15
3	13	14
4	8	12
4	9	13
4	10	14
4	11	15
5	8	14
5	9	12
5	10	15
5	11	13
6	8	15
6	9	14
6	10	13
6	11	12
7	8	13
7	9	15
7	10	12
7	11	14

nr. 5

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	7
3	5	6
3	8	11
3	9	10
3	12	15
3	13	14
4	8	12
4	9	13
4	10	15
4	11	14
5	8	14
5	9	15
5	10	13
5	11	12
6	8	13
6	9	12
6	10	14
6	11	15
7	8	15
7	9	14
7	10	12
7	11	13

nr. 6			nr. 7			nr. 8			nr. 9			nr. 10		
1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
1	4	5	1	4	5	1	4	5	1	4	5	1	4	5
1	6	7	1	6	7	1	6	7	1	6	7	1	6	7
1	8	9	1	8	9	1	8	9	1	8	9	1	8	9
1	10	11	1	10	11	1	10	11	1	10	11	1	10	11
1	12	13	1	12	13	1	12	13	1	12	13	1	12	13
1	14	15	1	14	15	1	14	15	1	14	15	1	14	15
2	4	6	2	4	6	2	4	6	2	4	6	2	4	6
2	5	7	2	5	7	2	5	7	2	5	7	2	5	7
2	8	10	2	8	10	2	8	10	2	8	10	2	8	10
2	9	11	2	9	11	2	9	11	2	9	11	2	9	11
2	12	14	2	12	14	2	12	14	2	12	14	2	12	14
2	13	15	2	13	15	2	13	15	2	13	15	2	13	15
3	4	7	3	4	7	3	4	7	3	4	7	3	4	7
3	5	6	3	5	6	3	5	6	3	5	6	3	5	6
3	8	11	3	8	11	3	8	11	3	8	11	3	8	11
3	9	10	3	9	10	3	9	12	3	9	12	3	9	12
3	12	15	3	12	15	3	10	15	3	10	15	3	10	15
3	13	14	3	13	14	3	13	14	3	13	14	3	13	14
4	8	12	4	8	12	4	8	13	4	8	13	4	8	13
4	9	13	4	9	14	4	9	10	4	9	10	4	9	10
4	10	15	4	10	15	4	11	14	4	11	14	4	11	14
4	11	14	4	11	13	4	12	15	4	12	15	4	12	15
5	8	14	5	8	15	5	8	15	5	8	15	5	8	15
5	9	15	5	9	13	5	9	14	5	9	14	5	9	14
5	10	13	5	10	12	5	10	13	5	10	12	5	10	13
5	11	12	5	11	14	5	11	12	5	11	13	5	11	12
6	8	13	6	8	13	6	8	14	6	8	14	6	8	14
6	9	14	6	9	15	6	9	15	6	9	15	6	9	13
6	10	12	6	10	14	6	10	12	6	10	13	6	10	12
6	11	15	6	11	12	6	11	13	6	11	12	6	11	15
7	8	15	7	8	14	7	8	12	7	8	12	7	8	12
7	9	12	7	9	12	7	9	13	7	9	13	7	9	15
7	10	14	7	10	13	7	10	14	7	10	14	7	10	14
7	11	13	7	11	15	7	11	15	7	11	15	7	11	13

nr. 11

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	7
3	5	6
3	8	11
3	9	12
3	10	15
3	13	14
4	8	13
4	9	14
4	10	12
4	11	15
5	8	15
5	9	13
5	10	14
5	11	12
6	8	14
6	9	10
6	11	13
6	12	15
7	8	12
7	9	15
7	10	13
7	11	14

nr. 12

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	7
3	5	6
3	8	11
3	9	12
3	10	15
3	13	14
4	8	13
4	9	14
4	10	12
4	11	15
5	8	15
5	9	13
5	10	14
5	11	12
6	8	12
6	9	15
6	10	13
6	11	14
7	8	14
7	9	10
7	11	13
7	12	15

nr. 13

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	7
3	5	6
3	8	11
3	9	12
3	10	15
3	13	14
4	8	13
4	9	15
4	10	12
4	11	14
5	8	15
5	9	14
5	10	13
5	11	12
6	8	14
6	9	10
6	11	13
6	12	15
7	8	12
7	9	13
7	10	14
7	11	15

nr. 14

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	7
3	5	6
3	8	11
3	9	12
3	10	15
3	13	14
4	8	13
4	9	15
4	10	12
4	11	14
5	8	14
5	9	10
5	11	13
5	12	15
6	8	15
6	9	14
6	10	13
6	11	12
7	8	12
7	9	13
7	10	14
7	11	15

nr. 15

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	7
3	5	6
3	8	12
3	9	13
3	10	14
3	11	15
4	8	15
4	9	10
4	11	12
4	13	14
5	8	14
5	9	12
5	10	15
5	11	13
6	8	11
6	9	14
6	10	13
6	12	15
7	8	13
7	9	15
7	10	12
7	11	14

nr. 16

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	7
3	5	6
3	8	12
3	9	13
3	10	14
3	11	15
4	8	15
4	9	14
4	10	13
4	11	12
5	8	11
5	9	10
5	12	15
5	13	14
6	8	14
6	9	15
6	10	12
6	11	13
7	8	13
7	9	12
7	10	15
7	11	14

nr. 17

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	7
3	5	6
3	8	12
3	9	13
3	10	14
3	11	15
4	8	15
4	9	14
4	10	13
4	11	12
5	8	11
5	9	12
5	10	15
5	13	14
6	8	14
6	9	10
6	11	13
6	12	15
7	8	13
7	9	15
7	10	12
7	11	14

nr. 18

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	7
3	5	6
3	8	12
3	9	13
3	10	15
3	11	14
4	8	15
4	9	10
4	11	12
4	13	14
5	8	11
5	9	14
5	10	13
5	12	15
6	8	14
6	9	15
6	10	12
6	11	13
7	8	13
7	9	12
7	10	14
7	11	15

nr. 19

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	7
3	5	6
3	8	12
3	9	14
3	10	13
3	11	15
4	8	15
4	9	12
4	10	14
4	11	13
5	8	13
5	9	10
5	11	14
5	12	15
6	8	11
6	9	15
6	10	12
6	13	14
7	8	14
7	9	13
7	10	15
7	11	12

nr. 20

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	7
3	5	6
3	8	12
3	9	14
3	10	15
3	11	13
4	8	15
4	9	10
4	11	12
4	13	14
5	8	11
5	9	13
5	10	14
5	12	15
6	8	14
6	9	12
6	10	13
6	11	15
7	8	13
7	9	15
7	10	12
7	11	14

nr. 21

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	12
2	11	14
2	13	15
3	4	7
3	5	6
3	8	11
3	9	13
3	10	15
3	12	14
4	8	12
4	9	14
4	10	13
4	11	15
5	8	13
5	9	15
5	10	14
5	11	12
6	8	14
6	9	10
6	11	13
6	12	15
7	8	15
7	9	11
7	10	12
7	13	14

nr. 22

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	12
2	11	14
2	13	15
3	4	7
3	5	6
3	8	11
3	9	13
3	10	15
3	12	14
4	8	12
4	9	14
4	10	13
4	11	15
5	8	15
5	9	11
5	10	12
5	13	14
6	8	13
6	9	15
6	10	14
6	11	12
7	8	14
7	9	10
7	11	13
7	12	15

nr. 23

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	7
3	5	8
3	6	11
3	9	12
3	10	15
3	13	14
4	8	13
4	9	14
4	10	12
4	11	15
5	6	14
5	9	10
5	11	13
5	12	15
6	8	12
6	9	15
6	10	13
7	8	15
7	9	13
7	10	14
7	11	12
8	11	14

nr. 24

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	7
3	5	8
3	6	11
3	9	12
3	10	15
3	13	14
4	8	13
4	9	14
4	10	12
4	11	15
5	6	12
5	9	15
5	10	13
5	11	14
6	8	15
6	9	13
6	10	14
7	8	14
7	9	10
7	11	13
7	12	15
8	11	12

nr. 25

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	7
3	5	8
3	6	11
3	9	12
3	10	15
3	13	14
4	8	13
4	9	15
4	10	14
4	11	12
5	6	15
5	9	13
5	10	12
5	11	14
6	8	12
6	9	14
6	10	13
7	8	14
7	9	10
7	11	13
7	12	15
8	11	15

nr. 26

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	7
3	5	8
3	6	12
3	9	10
3	11	15
3	13	14
4	8	13
4	9	12
4	10	15
4	11	14
5	6	15
5	9	14
5	10	12
5	11	13
6	8	11
6	9	13
6	10	14
7	8	14
7	9	15
7	10	13
7	11	12
8	12	15

nr. 27

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	7
3	5	8
3	6	12
3	9	13
3	10	14
3	11	15
4	8	14
4	9	10
4	11	13
4	12	15
5	6	11
5	9	15
5	10	12
5	13	14
6	8	13
6	9	14
6	10	15
7	8	15
7	9	12
7	10	13
7	11	14
8	11	12

nr. 28

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	7
3	5	8
3	6	12
3	9	13
3	10	15
3	11	14
4	8	14
4	9	10
4	11	13
4	12	15
5	6	11
5	9	15
5	10	12
5	13	14
6	8	15
6	9	14
6	10	13
7	8	13
7	9	12
7	10	14
7	11	15
8	11	12

nr. 29

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	7
3	5	8
3	6	12
3	9	13
3	10	15
3	11	14
4	8	15
4	9	12
4	10	14
4	11	13
5	6	11
5	9	10
5	12	15
5	13	14
6	8	14
6	9	15
6	10	13
7	8	13
7	9	14
7	10	12
7	11	15
8	11	12

nr. 30

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	7
3	5	8
3	6	12
3	9	14
3	10	13
3	11	15
4	8	14
4	9	10
4	11	13
4	12	15
5	6	11
5	9	12
5	10	15
5	13	14
6	8	13
6	9	15
6	10	14
7	8	15
7	9	13
7	10	12
7	11	14
8	11	12

nr. 31

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	7
3	5	8
3	6	12
3	9	14
3	10	13
3	11	15
4	8	14
4	9	12
4	10	15
4	11	13
5	6	11
5	9	10
5	12	15
5	13	14
6	8	13
6	9	15
6	10	14
7	8	15
7	9	13
7	10	12
7	11	14
8	11	12

nr. 32

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	7
3	5	8
3	6	12
3	9	14
3	10	13
3	11	15
4	8	13
4	9	10
4	11	14
4	12	15
5	6	11
5	9	15
5	10	12
5	13	14
6	8	14
6	9	13
6	10	15
7	8	15
7	9	12
7	10	14
7	11	13
8	11	12

nr. 33

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	7
3	5	8
3	6	12
3	9	14
3	10	15
3	11	13
4	8	13
4	9	10
4	11	14
4	12	15
5	6	11
5	9	15
5	10	12
5	13	14
6	8	15
6	9	13
6	10	14
7	8	14
7	9	12
7	10	13
7	11	15
8	11	12

nr. 34

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	7
3	5	8
3	6	12
3	9	14
3	10	15
3	11	13
4	8	14
4	9	12
4	10	13
4	11	15
5	6	11
5	9	15
5	10	12
5	13	14
6	8	15
6	9	13
6	10	14
7	8	13
7	9	10
7	12	15
7	11	14
8	11	12

nr. 35

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	8
3	5	9
3	6	12
3	7	13
3	10	14
3	11	15
4	7	14
4	9	12
4	10	15
4	11	13
5	6	15
5	8	14
5	10	13
5	11	12
6	8	13
6	9	10
6	11	14
7	8	11
7	9	15
7	10	12
8	12	15
9	13	14

nr. 36

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	8
3	5	9
3	6	12
3	7	13
3	10	15
3	11	14
4	7	14
4	9	12
4	10	13
4	11	15
5	6	15
5	8	13
5	10	14
5	11	12
6	8	14
6	9	10
6	11	13
7	8	11
7	9	15
7	10	12
8	12	15
9	13	14

nr. 37

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	8
3	5	9
3	6	12
3	7	13
3	10	15
3	11	14
4	7	10
4	9	14
4	11	13
4	12	15
5	6	11
5	8	15
5	10	12
5	13	14
6	8	13
6	9	15
6	10	14
7	8	14
7	9	12
7	11	15
8	11	12
9	10	13

nr. 38

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	8
3	5	9
3	6	12
3	7	14
3	10	13
3	11	15
4	7	10
4	9	13
4	11	14
4	12	15
5	6	13
5	8	15
5	10	14
5	11	12
6	8	11
6	9	14
6	10	15
7	8	12
7	9	15
7	11	13
8	13	14
9	10	12

nr. 39

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	8
3	5	9
3	6	12
3	7	14
3	10	13
3	11	15
4	7	10
4	9	14
4	11	13
4	12	15
5	6	11
5	8	12
5	10	15
5	13	14
6	8	15
6	9	13
6	10	14
7	8	13
7	9	15
7	11	12
8	11	14
9	10	12

nr. 40

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	8
3	5	9
3	6	12
3	7	14
3	10	13
3	11	15
4	7	13
4	9	12
4	10	15
4	11	14
5	6	15
5	8	13
5	10	14
5	11	12
6	8	14
6	9	10
6	11	13
7	8	11
7	9	15
7	10	12
8	12	15
9	13	14

nr. 41

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	8
3	5	9
3	6	12
3	7	14
3	10	13
3	11	15
4	7	11
4	9	12
4	10	15
4	13	14
5	6	10
5	8	14
5	11	13
5	12	15
6	8	15
6	9	13
6	11	14
7	8	13
7	9	15
7	10	12
8	11	12
9	10	14

nr. 42

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	8
3	5	9
3	6	12
3	7	14
3	10	15
3	11	13
4	7	10
4	9	13
4	11	14
4	12	15
5	6	11
5	8	15
5	10	12
5	13	14
6	8	14
6	9	15
6	10	13
7	8	13
7	9	12
7	11	15
8	11	12
9	10	14

nr. 43

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	8
3	5	9
3	6	12
3	7	14
3	10	15
3	11	13
4	7	10
4	9	15
4	11	12
4	13	14
5	6	11
5	8	13
5	10	14
5	12	15
6	8	15
6	9	14
6	10	13
7	8	12
7	9	13
7	11	15
8	11	14
9	10	12

nr. 44

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	8
3	5	9
3	6	12
3	7	14
3	10	15
3	11	13
4	7	13
4	9	12
4	10	14
4	11	15
5	6	15
5	8	14
5	10	13
5	11	12
6	8	13
6	9	10
6	11	14
7	8	11
7	9	15
7	10	12
8	12	15
9	13	14

nr. 45

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	8
3	5	9
3	6	12
3	7	15
3	10	13
3	11	14
4	7	10
4	9	12
4	11	15
4	13	14
5	6	15
5	8	14
5	10	12
5	11	13
6	8	11
6	9	13
6	10	14
7	8	13
7	9	14
7	11	12
8	12	15
9	10	15

nr. 46

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	8
3	5	9
3	6	12
3	7	15
3	10	13
3	11	14
4	7	10
4	9	14
4	11	13
4	12	15
5	6	11
5	8	12
5	10	15
5	13	14
6	8	13
6	9	15
6	10	14
7	8	14
7	9	13
7	11	12
8	11	15
9	10	12

nr. 47

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	8
3	5	9
3	6	12
3	7	15
3	10	14
3	11	13
4	7	10
4	9	13
4	11	14
4	12	15
5	6	11
5	8	12
5	10	15
5	13	14
6	8	14
6	9	15
6	10	13
7	8	13
7	9	14
7	11	12
8	11	15
9	10	12

nr. 48

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	8
3	5	10
3	6	12
3	7	15
3	9	13
3	11	14
4	7	9
4	10	14
4	11	13
4	12	15
5	6	11
5	8	12
5	9	15
5	13	14
6	8	13
6	9	14
6	10	15
7	8	14
7	10	13
7	11	12
8	11	15
9	10	12

nr. 49

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	8
3	5	10
3	6	12
3	7	15
3	9	13
3	11	14
4	7	14
4	9	15
4	10	12
4	11	13
5	6	13
5	8	14
5	9	12
5	11	15
6	8	11
6	9	14
6	10	15
7	8	13
7	9	10
7	11	12
8	12	15
10	13	14

nr. 50

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	8
3	5	10
3	6	12
3	7	15
3	9	13
3	11	14
4	7	9
4	10	12
4	11	15
4	13	14
5	6	15
5	8	14
5	9	12
5	11	13
6	8	11
6	9	14
6	10	13
7	8	13
7	10	14
7	11	12
8	12	15
9	10	15

nr. 51

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	8
3	5	11
3	6	12
3	7	15
3	9	13
3	10	14
4	7	10
4	9	14
4	11	13
4	12	15
5	6	9
5	8	15
5	10	12
5	13	14
6	8	14
6	10	13
6	11	15
7	8	13
7	9	12
7	11	14
8	11	12
9	10	15

nr. 52

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	8
3	5	11
3	6	12
3	7	15
3	9	13
3	10	14
4	7	13
4	9	14
4	10	15
4	11	12
5	6	9
5	8	15
5	10	12
5	13	14
6	8	14
6	10	13
6	11	15
7	8	12
7	9	10
7	11	14
8	11	13
9	12	15

nr. 53

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	8
3	5	11
3	6	12
3	7	15
3	9	13
3	10	14
4	7	12
4	9	14
4	10	15
4	11	13
5	6	9
5	10	12
5	8	15
5	13	14
6	8	14
6	10	13
6	11	15
7	8	13
7	9	10
7	11	14
8	11	12
9	12	15

nr. 54

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	8
3	5	11
3	6	12
3	7	15
3	9	13
3	10	14
4	7	12
4	9	15
4	10	13
4	11	14
5	6	10
5	8	15
5	9	12
5	13	14
6	8	13
6	9	14
6	11	15
7	8	14
7	9	10
7	11	13
8	11	12
10	12	15

nr. 55

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	11
2	12	14
2	13	15
3	4	8
3	5	11
3	6	12
3	7	15
3	9	13
3	10	14
4	7	9
4	10	13
4	11	14
4	12	15
5	6	10
5	8	15
5	9	12
5	13	14
6	8	13
6	9	14
6	11	15
7	8	14
7	10	12
7	11	13
8	11	12
9	10	15

nr. 56			nr. 57			nr. 58			nr. 59			nr. 60		
1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
1	4	5	1	4	5	1	4	5	1	4	5	1	4	5
1	6	7	1	6	7	1	6	7	1	6	7	1	6	7
1	8	9	1	8	9	1	8	9	1	8	9	1	8	9
1	10	11	1	10	11	1	10	11	1	10	11	1	10	11
1	12	13	1	12	13	1	12	13	1	12	13	1	12	13
1	14	15	1	14	15	1	14	15	1	14	15	1	14	15
2	4	6	2	4	6	2	4	6	2	4	6	2	4	6
2	5	7	2	5	7	2	5	7	2	5	7	2	5	7
2	8	10	2	8	10	2	8	10	2	8	10	2	8	10
2	9	11	2	9	11	2	9	11	2	9	11	2	9	11
2	12	14	2	12	14	2	12	14	2	12	14	2	12	14
2	13	15	2	13	15	2	13	15	2	13	15	2	13	15
3	4	8	3	4	8	3	4	8	3	4	8	3	4	8
3	5	11	3	5	11	3	5	11	3	5	15	3	5	15
3	6	12	3	6	12	3	6	12	3	6	12	3	6	12
3	7	15	3	7	15	3	7	15	3	7	11	3	7	11
3	9	14	3	9	14	3	9	14	3	9	13	3	9	13
3	10	13	3	10	13	3	10	13	3	10	14	3	10	14
4	7	14	4	7	10	4	7	12	4	7	13	4	7	14
4	9	13	4	9	13	4	9	13	4	9	12	4	9	15
4	10	15	4	11	14	4	10	15	4	10	15	4	11	13
4	11	12	4	12	15	4	11	14	4	11	14	4	10	12
5	6	9	5	6	9	5	6	9	5	6	14	5	6	13
5	8	15	5	8	15	5	8	15	5	8	12	5	8	12
5	10	12	5	10	12	5	10	12	5	9	10	5	9	10
5	13	14	5	13	14	5	13	14	5	11	13	5	11	14
6	8	13	6	8	13	6	8	13	6	8	11	6	8	11
6	10	14	6	10	14	6	10	14	6	9	15	6	9	14
6	11	15	6	11	15	6	11	15	6	10	13	6	10	15
7	8	12	7	8	14	7	8	14	7	8	15	7	8	15
7	9	10	7	9	12	7	9	10	7	9	14	7	9	12
7	11	13	7	11	13	7	11	13	7	10	12	7	10	13
8	11	14	8	11	12	8	11	12	8	13	14	8	13	14
9	12	15	9	10	15	9	12	15	11	12	15	11	12	15

nr. 61

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	12
2	11	14
2	13	15
3	4	7
3	5	6
3	8	11
3	9	15
3	10	13
3	12	14
4	8	12
4	9	11
4	10	15
4	13	14
5	8	15
5	9	13
5	10	14
5	11	12
6	8	14
6	9	10
6	11	13
6	12	15
7	8	13
7	9	14
7	10	12
7	11	15

nr. 62

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	12
2	11	14
2	13	15
3	4	7
3	5	8
3	6	11
3	9	14
3	10	13
3	12	15
4	9	10
4	8	13
4	11	15
4	12	14
5	6	12
5	9	15
5	10	14
5	11	13
6	8	14
6	9	13
6	10	15
7	8	15
7	9	11
7	10	12
7	13	14
8	11	12

nr. 63

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	12
2	11	14
2	13	15
3	4	7
3	5	8
3	6	11
3	9	15
3	10	13
3	12	14
4	8	14
4	9	10
4	11	13
4	12	15
5	6	12
5	9	13
5	10	14
5	11	15
6	8	13
6	9	14
6	10	15
7	8	15
7	9	11
7	10	12
7	13	14
8	11	12

nr. 64

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	12
2	11	14
2	13	15
3	4	7
3	5	8
3	6	13
3	9	14
3	10	12
3	11	15
4	8	15
4	9	13
4	10	14
4	11	12
5	6	11
5	9	15
5	10	13
5	12	14
6	8	14
6	9	10
6	12	15
7	8	12
7	9	11
7	10	15
7	13	14
8	11	13

nr. 65

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	12
2	11	14
2	13	15
3	4	8
3	5	9
3	6	14
3	7	11
3	10	13
3	12	15
4	7	15
4	9	10
4	11	12
4	13	14
5	6	12
5	8	15
5	10	14
5	11	13
6	8	11
6	9	13
6	10	15
7	8	13
7	9	14
7	10	12.
8	12	14
9	11	15

nr. 66

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	12
2	11	14
2	13	15
3	4	8
3	5	12
3	6	14
3	7	11
3	9	13
3	10	15
4	7	15
4	9	10
4	11	12
4	13	14
5	6	9
5	8	15
5	10	14
5	11	13
6	8	11
6	10	13
6	12	15
7	8	13
7	9	14
7	10	12
8	12	14
9	11	15

nr. 67

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	12
2	11	14
2	13	15
3	4	8
3	5	10
3	6	13
3	7	12
3	9	14
3	11	15
4	7	15
4	9	10
4	11	12
4	13	14
5	6	11
5	8	15
5	9	13
5	12	14
6	8	12
6	9	15
6	10	14
7	8	14
7	9	11
7	10	13
8	11	13
10	12	15

nr. 68

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	12
2	11	14
2	13	15
3	4	8
3	5	9
3	6	11
3	7	14
3	10	13
3	12	15
4	7	15
4	9	10
4	11	12
4	13	14
5	6	12
5	8	15
5	10	14
5	11	13
6	8	13
6	9	14
6	10	15
7	8	11
7	9	13
7	10	12
8	12	14
9	11	15

nr. 69

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	12
2	11	14
2	13	15
3	4	8
3	5	10
3	6	12
3	7	13
3	11	15
3	9	14
4	7	15
4	9	10
4	11	12
4	13	14
5	6	11
5	8	15
5	9	13
5	12	14
6	8	14
6	9	15
6	10	13
7	8	12
7	9	11
7	10	14
8	11	13
10	12	15

nr. 70

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	12
2	11	14
2	13	15
3	4	8
3	5	14
3	6	12
3	7	11
3	9	15
3	10	13
4	7	15
4	9	10
4	11	13
4	12	14
5	6	13
5	8	15
5	9	11
5	10	12
6	8	11
6	9	14
6	10	15
7	8	12
7	9	13
7	10	14
8	13	14
11	12	15

nr. 71

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	12
2	11	14
2	13	15
3	4	8
3	5	13
3	6	11
3	7	12
3	9	14
3	10	15
4	7	15
4	9	10
4	11	13
4	12	14
5	6	14
5	8	15
5	9	11
5	10	12
6	8	12
6	9	15
6	10	13
7	8	11
7	9	13
7	10	14
8	13	14
11	12	15

nr. 72

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	12
2	11	14
2	13	15
3	4	8
3	5	14
3	6	13
3	7	12
3	9	11
3	10	15
4	7	15
4	9	10
4	11	13
4	12	14
5	6	11
5	8	15
5	9	13
5	10	12
6	8	12
6	9	15
6	10	14
7	8	11
7	9	14
7	10	13
8	13	14
11	12	15

nr. 73

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	12
2	11	14
2	13	15
3	4	8
3	5	13
3	6	10
3	7	12
3	9	14
3	11	15
4	7	15
4	9	13
4	10	14
4	11	12
5	6	11
5	8	15
5	9	10
5	12	14
6	8	12
6	9	15
6	13	14
7	8	14
7	9	11
7	10	13
8	11	13
10	12	15

nr. 74

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	12
2	11	14
2	13	15
3	4	8
3	5	11
3	6	12
3	7	14
3	10	15
3	9	13
4	7	15
4	9	14
4	10	12
4	11	13
5	6	13
5	8	15
5	9	10
5	12	14
6	8	11
6	9	15
6	10	14
7	8	12
7	9	11
7	10	13
8	13	14
11	12	15

nr. 75

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	12
2	11	14
2	13	15
3	4	8
3	5	13
3	6	14
3	7	12
3	9	11
3	10	15
4	7	15
4	9	14
4	10	12
4	11	13
5	6	11
5	8	15
5	9	10
5	12	14
6	8	12
6	9	15
6	10	13
7	8	11
7	9	13
7	10	14
8	13	14
11	12	15

nr. 76

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	12
2	11	14
2	13	15
3	4	8
3	5	14
3	6	11
3	7	9
3	10	13
3	12	15
4	7	15
4	9	10
4	11	13
4	12	14
5	6	13
5	8	12
5	9	11
5	10	15
6	8	14
6	9	15
6	10	12
7	8	13
7	10	14
7	11	12
8	11	15
9	13	14

nr. 77

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	12
2	11	14
2	13	15
3	4	8
3	5	14
3	6	9
3	7	13
3	10	15
3	11	12
4	7	15
4	9	10
4	11	13
4	12	14
5	6	11
5	8	13
5	9	15
5	10	12
6	8	14
6	10	13
6	12	15
7	8	12
7	9	11
7	10	14
8	11	15
9	13	14

nr. 78

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	12
2	11	14
2	13	15
3	4	8
3	5	11
3	6	13
3	7	9
3	10	15
3	12	14
4	7	15
4	9	11
4	10	12
4	13	14
5	6	10
5	8	13
5	9	14
5	12	15
6	8	14
6	9	15
6	11	12
7	8	12
7	10	14
7	11	13
8	11	15
9	10	13

nr. 79

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	7
2	8	10
2	9	12
2	11	14
2	13	15
3	4	8
3	5	10
3	6	15
3	7	13
3	9	11
3	12	14
4	7	11
4	9	15
4	10	12
4	13	14
5	6	9
5	8	14
5	11	13
5	12	15
6	8	13
6	10	14
6	11	12
7	8	12
7	9	14
7	10	15
8	11	15
9	10	13

nr. 80

1	2	3
1	4	5
1	6	7
1	8	9
1	10	11
1	12	13
1	14	15
2	4	6
2	5	8
2	7	9
2	10	12
2	11	14
2	13	15
3	4	10
3	5	7
3	6	11
3	8	15
3	9	13
3	12	14
4	7	12
4	8	13
4	9	14
4	11	15
5	6	14
5	9	10
5	11	13
5	12	15
6	8	12
6	9	15
6	10	13
7	8	11
7	10	15
7	13	14
8	10	14
9	11	12

2. Latin squares of order 6

The 17 Latin squares of order 6 occur in [5], p. 87. In the terminology of Chapter I, section 2, only 12 of these yield pairwise non-isomorphic Latin square graphs on 36 vertices. The corresponding Latin squares are listed below. Theorem 3.1 of Chapter I deals with nr. 87 and nr. 89.

nr. 81

1	2	3	4	5	6
2	3	6	1	4	5
3	6	2	5	1	4
4	5	1	2	6	3
5	1	4	6	3	2
6	4	5	3	2	1

nr. 82

1	2	3	4	5	6
2	1	6	5	3	4
3	6	2	1	4	5
4	3	5	2	6	1
5	4	1	6	2	3
6	5	4	3	1	2

nr. 83

1	2	3	4	5	6
2	1	5	6	3	4
3	6	2	1	4	5
4	5	1	2	6	3
5	4	6	3	2	1
6	3	4	5	1	2

nr. 84

1	2	3	4	5	6
2	1	5	3	6	4
3	6	2	1	4	5
4	5	6	2	3	1
5	4	1	6	2	3
6	3	4	5	1	2

nr. 85

1	2	3	4	5	6
2	3	4	5	6	1
3	5	1	6	2	4
4	6	2	1	3	5
5	4	6	2	1	3
6	1	5	3	4	2

nr. 86

1	2	3	4	5	6
2	1	5	6	3	4
3	6	1	5	4	2
4	3	2	1	6	5
5	4	6	3	2	1
6	5	4	2	1	3

nr. 87

1	2	3	4	5	6
2	3	6	1	4	5
3	6	2	5	1	4
4	1	5	2	6	3
5	4	1	6	3	2
6	5	4	3	2	1

nr. 88

1	2	3	4	5	6
2	3	1	6	4	5
3	1	2	5	6	4
4	6	5	2	1	3
5	4	6	3	2	1
6	5	4	1	3	2

nr. 89

1	2	3	4	5	6
2	3	1	6	4	5
3	1	2	5	6	4
4	6	5	2	1	3
5	4	6	1	3	2
6	5	4	3	2	1

nr. 90

1	2	3	4	5	6
2	3	1	5	6	4
3	1	2	6	4	5
4	6	5	2	1	3
5	4	6	3	2	1
6	5	4	1	3	2

nr. 91

1	2	3	4	5	6
2	1	6	5	4	3
3	4	1	2	6	5
4	6	5	1	3	2
5	3	2	6	1	4
6	5	4	3	2	1

nr. 92

1	2	3	4	5	6
2	3	1	6	4	5
3	1	2	5	6	4
4	5	6	1	2	3
5	6	4	3	1	2
6	4	5	2	3	1

CHAPTER IV

ALGOL PROGRAMS

1. Introduction

This chapter deals with the programs Equivalence and Subsystem. The program Equivalence serves to investigate the mutual equivalences of the Steiner graphs and the Latin square graphs of order 36. It provides a tool for the proof of Theorem 3.2 of Chapter I. The program uses the procedure Cliquelist of Chapter II, and the procedures Steinergraph, Latsqgraph, and Control, which are described below.

The program Subsystem examines the Steiner triple systems of order 15 on subsystems of 15 triples in which each of the 15 symbols occurs three times. It provides a tool for the proof of Theorem 4.1 of Chapter I.

The numerical results are listed in Chapter V.

2. procedure Steinergraph (dim, order, A)

From a Steiner triple system of a certain order, consisting of dim - 1 triples, the (-1,1) adjacency matrix A of the corresponding Steiner graph of order dim is constructed.

2.1. Formal parameters

integer dim: < expression >; denotes the order of the symmetric matrix A; is called by value.

integer order: < expression >; denotes the order of the Steiner triple system; is called by value.

integer array A: array of dimensions [1 : dim, 1 : dim]; on exit of the procedure, contains the elements of the right upper half of the (-1,1) adjacency matrix of the Steiner graph.

2.2. Global parameters

AND: the assumption is made that the library of subroutines contains the procedure AND(a,b), which calculates the logical product of the binary numbers a and b: **integer procedure** AND(a,b); **value** a,b; **integer** a,b.

2.3. Explanation

The Steiner graph is constructed from the dim - 1 triples on the paper tape as follows. The first vertex is an isolated vertex. The (i+1)-th vertex of the graph corresponds to the i-th triple on the paper tape. To the triple (p,q,r) there is attached the number $g = 2^p + 2^q + 2^r$, $0 < p,q,r \leq$ order. Two vertices are adjacent iff the corresponding triples have one symbol in common, that is, iff the logical product (in binary notation) of their attached numbers g is non-zero.

3. procedure Latsqgraph (dim, A)

From a Latin square of order sqrt(dim) the (-1,1) adjacency matrix A of the corresponding Latin square graph is constructed.

3.1. Formal parameters

integer dim:	< expression >; denotes the order of the symmetric matrix A; is called by value.
integer array A:	array of dimensions [1 : dim, 1 : dim]; on exit of the procedure, contains the elements of the right upper half of the adjacency matrix of the Latin square graph.

3.2. Explanation

The Latin square graph is constructed from the given Latin square on the paper tape as follows. The i-th vertex of the graph corresponds to the i-th element on the paper tape. Two vertices are adjacent iff the corresponding elements are the same, or belong to the same row, or belong to the same column.

4. procedure Control (dim, A, label)

This procedure checks whether the Latin square graph of order dim, and the subgraph of order dim - 1 of the Steiner graph, are regular. If not, then there is an error in the Latin square, or in the Steiner triple system, and the procedure branches to the label label in the main program. The meaning of the formal parameters dim, A, label is obvious.

5. The ALGOL program Equivalence

```
begin comment Investigation of Steiner graphs and Latin square graphs of
      order 36 for mutual equivalence with the help of the procedure
      Cliquelist, of which heading and body can be found elsewhere. Input of
      data must come in this sequence:
      1. v. This is the order (4 or 5) of the void subgraphs with which this
         investigation is done.
      2. next, per graph to be examined,
         (i)  nr. This is the serial number of the graph. The numbers 1
                through 80 correspond to the Steiner graphs, the numbers 81
                through 92 to the Latin square graphs.
         (ii) The 35 triples from which the Steiner graph is to be construct-
                ed (if 1 ≤ nr ≤ 80) or the elements of the Latin square of
                order 6 (if 81 ≤ nr ≤ 92).
      For each graph that is examined, the program determines the numbers
      (≠ 0) of subgraphs of order 35 that contain a specified number of void
      subgraphs of the given order;
      integer i,nr,number,v; integer array n,N[1 : 36],A[1 : 36,1 : 36];
      < procedure Cliquelist; cf. pp. 21,22,23 >

      procedure Steinergraph(dim,order,A); value dim,order;
      integer dim,order; integer array A;
      begin comment From a Steiner triple system of a certain order consist-
            ing of dim - 1 triples, the program constructs the right upper
            half of the (-1,1) adjacency matrix A of the Steiner graph of
            order dim;
            integer a,i,j; integer array bin[1 : order],bitrow[2 : dim];
            a := bin[1] := 1;
            for i := 2 step 1 until order do begin a := a × 2; bin[i] := a end;
            for i := 2 step 1 until dim do
            begin a := bitrow[i] := bin[read] + bin[read] + bin[read];
                  for j := i - 1 step -1 until 2 do
                  A[j,i] := if AND(a,bitrow[j]) > 0 then -1 else 1
            end;
            for j := 2 step 1 until dim do A[1,j] := 1
      end Steinergraph;
```

```
procedure Latsqgraph(dim,A); value dim;
integer dim; integer array A;
begin comment From a Latin square of order sqrt(dim) the right upper
      half of the (-1,1) adjacency matrix A of the Latin square graph
      of order dim is constructed;
      integer a,dim1,i,i1,j,j1,k,k1,n,n1; integer array Lat[1 : dim];
      dim1 := dim - 1; n := sqrt(dim); n1 := n - 1;
      for i := 1 step 1 until dim1 do
      for j := i + 1 step 1 until dim do A[i,j] := 1;
      for i := 1 step 1 until n1 do
      begin i1 := (i - 1) × n;
            for j := i + 1 step 1 until n do
            begin j1 := (j - 1) × n;
                  for k := 1 step 1 until n do
                  begin k1 := (k - 1) × n;
                        A[i1 + k,j1 + k] := A[i + k1,j + k1] := -1
                  end
            end
      end;
      for i := 1 step 1 until dim do
      begin a := Lat[i] := read;
            for j := i - 1 step -1 until 1 do
            if Lat[j] = a then A[j,i] := -1
      end
end Latsqgraph;

procedure Control(dim,A,label); value dim;
integer dim; integer array A; label label;
begin comment The procedure checks whether the Latin square graph of
      order dim or the subgraph S (of order dim - 1) of the Steiner
      graph is regular. If this is not the case, there is an error in
      the Latin square, or in the Steiner triple system from which the
      graph with (-1,1) adjacency matrix A has been constructed, and
      the procedure branches to the label label in the main program;
      integer dim1,i,i1,j,s,sum; Boolean first;
      first := true; dim1 := dim - 1;
      for i := 1 step 1 until dim do
      begin i1 := i - 1; sum := 0;
```

```
                    for j := 1 step 1 until i1 do sum := sum + A[j,i];
                    for j := i + 1 step 1 until dim do sum := sum + A[i,j];
                    if sum ≠ dim1 then
                    begin if first then begin s := sum; first := false end else
                        if sum ≠ s then
                        begin NLCR; PRINTTEXT(⟨the graph is not regular,⟩);
                            PRINTTEXT(⟨ error in vertex number⟩);
                            ABSFIXT(2,0,i); goto label
                        end
                    end
                end
            end Control;

            v := read;
            for nr := read while nr > 0 do
            begin CARRIAGE(4); PRINTTEXT(⟨nr.⟩); ABSFIXT(2,0,nr);
                if nr > 80 then Latsqgraph(36,A) else Steinergraph(36,15,A);
                Control(36,A,End); Cliquelist(36,A,false,v,number,n,N);
                for i := 1 step 1 until number do
                begin NLCR; PRINTTEXT(⟨number of subgraphs of the order 35 with⟩);
                    ABSFIXT(3,0,n[i]); PRINTTEXT(⟨void subgraphs of the order⟩);
                    ABSFIXT(1,0,v); PRINTTEXT(⟨is ⟩); ABSFIXT(2,0,N[i])
                end;
End:    end
end
```

6. The ALGOL program Subsystem

The program examines the Steiner triple system S of order 15 for a subsystem T of 15 triples in which each of the 15 symbols of S occurs three times.

6.1. Explanation

The 35 triples of S are given as data on the paper tape. They are denoted by

$$\{triple1[n],triple2[n],triple3[n]\}, \qquad n = 1,2,\dots,35 ,$$

with elements from $\{1,2,\dots,15\}$. Each triple is ordered according to triple1[n] < triple2[n] < triple3[n], and the set of the 35 triples of S is ordered lexicographically.

The program uses the arrays lb, range, ub.

lb[i], 1 ≤ i ≤ 15, denotes the sequence number nr of the first triple in S for which triple1[nr] = i, at least if such a triple exists. If not, then lb[i] is not filled.

range[i], 1 ≤ i ≤ 15, denotes the frequency of the triples in S which begin with the number i.

ub[i], 1 ≤ i ≤ 15, equals lb[i] + range[i], for range[i] ≠ 0; otherwise ub[i] is not filled.

The program is based on the back-tracking algorithm, and operates according to the following outline. Suppose that in the subsystem T_{i-1} each of the numbers 1,2,...,i-1 occurs three times, i less than, and i+1,...,15 at most three times. The program searches for a set U_i of triples, each containing i, which together with T_{i-1} forms a subsystem T_i in which i occurs three times, and each of i+1,...,15 occurs not more than three times. There may exist several sets U_i. Since the triples of U_i have i as their first element, the search for U_i may be restricted to the subset S_i of S which contains all triples having sequence number nr with lb[i] ≤ nr < < ub[i]. (The process of searching for triples within S_i again is based on the back-tracking algorithm.) If a set U_i is found, then i is increased by one and the program proceeds as before. If no U_i exists, then the program steps back one, searches for the next set U_{i-1} from S_{i-1}, and continues forward again.

Finally, some variables used in the program are described.

i: keeps track of the numbers 1,2,...,i-1, which already occur three times in the subsystem;

index: denotes how many triples are already in the subsystem;

j: denotes how many triples in the subsystem have i as their first element;

nr: the sequence number of the triples in S;

freq[k], 1 ≤ k ≤ 15: denotes how many more triples containing the number k > i are needed for the subsystem;

subsyst[index], 1 ≤ index ≤ 15: denotes the sequence number nr in S corresponding to the triple with sequence number index in the subsystem;

bound := ub[i] - freq[i] + j: the sequence number in S of the j-th triple, having i as its first element, is smaller than bound;

a,a1,a2,a3,b2,b3,k,l,lasta,lastlb: auxiliary variables.

6.2. The ALGOL text of the program Subsystem

```
begin comment The program examines a Steiner triple system of order 15 on
      subsystems of 15 triples, in which each of the 15 symbols of the triple
      system occurs three times. The program lists the first subsystem found,
      if it exists.
        Input of data per triple system which should be examined sequentially:
      (i)  nr. This is the serial number of the triple system to be examined.
      (ii) The 35 Steiner triples. The elements of the triples are
           triple1[n],triple2[n],triple3[n] ∈ {1,2,...,15}, n = 1,2,...,35.
           Each triple is ordered such that triple1[n] < triple2[n] <
           < triple3[n] and the set of triples is ordered lexicographically.
      integer a,a1,a2,a3,b2,b3,bound,i,index,j,k,l,lasta,lastlb,nr;
      integer array triple1,triple2,triple3[1:35],freq,lb,range,subsyst,ub[1:15]

      comment Initializing;
Begin: nr := read; if nr ≤ 0 then goto End;
      CARRIAGE(4); PRINTTEXT(⊰nr.⊱); ABSFIXT(2,0,nr); NLCR;
      for k := 1 step 1 until 35 do
      begin triple1[k] := read; triple2[k] := read; triple3[k] := read end;
      for k := 1 step 1 until 15 do freq[k] := 3;
      lb[1] := 1; lb[2] := 8; lb[3] := 14; range[1] := 7; range[2] := 6;
      lasta := 3; lastlb := 14;
      for k := 19 step 1 until 35 do
      begin a := triple1[k];
            if a ≠ lasta then
            begin lb[a] := k; range[lasta] := k - lastlb; a1 := a - 1;
                  for l := lasta + 1 step 1 until a1 do range[l] := 0;
                  lasta := a; lastlb := k
            end
      end; range[a] := 36 - lastlb;
      for k := a + 1 step 1 until 15 do range[k] := 0;
      for k := 1 step 1 until 15 do
      if range[k] ≠ 0 then ub[k] := lb[k] + range[k];
```

```
      comment Test for subsystems;
      i := 0; index := 0;
nexti: i := i + 1;
      if i = 16 then goto print;
      a := freq[i]; if a = 0 then goto nexti;
      if a > range[i] then goto lasti;
      bound := ub[i] - a; j := 0; nr := lb[i] - 1;
nextj: j := j + 1;
      if j > a then goto nexti;
      bound := bound + 1;
nextnr: nr := nr + 1;
      if nr ≥ bound then
      begin j := j - 1;
            if j ≠ 0 then
            begin nr := subsyst[index];
                  a2 := triple2[nr]; a3 := triple3[nr];
                  freq[a2] := freq[a2] + 1; freq[a3] := freq[a3] + 1;
                  index := index - 1; bound := bound - 1; goto nextnr
            end else if i = 1 then
            begin NLCR; PRINTTEXT(⊀no subsystems of order 15⊁);
                  goto Begin
            end else goto lasti
      end;
      a2 := triple2[nr]; a3 := triple3[nr];
      b2 := freq[a2]; b3 := freq[a3];
      if b2 = 0 ∨ b3 = 0 then goto nextnr;
      freq[a2] := b2 - 1; freq[a3] := b3 - 1;
      index := index + 1; subsyst[index] := nr; goto  nextj;
lasti: nr := subsyst[index]; i := triple1[nr];
      a2 := triple2[nr]; a3 := triple3[nr];
      freq[a2] := freq[a2] + 1; freq[a3] := freq[a3] + 1;
      index := index - 1; j := a  := freq[i]; bound := ub[i];
      goto nextnr;
 print:  for k := 1 step 1 until 15 do
      begin NLCR; a := subsyst[k]; ABSFIXT(2,0,a);
            ABSFIXT(6,0,triple1[a]); ABSFIXT(3,0,triple2[a]);
            ABSFIXT(3,0,triple3[a])
      end; goto Begin;
End:
end
```

CHAPTER V

NUMERICAL RESULTS

Introduction

Table 1 contains the non-zero numbers M(m,5) for all Steiner graphs of order 36, indicated by their serial number, and for all m (= the number of void subgraphs of order 5) which have $M(m,5) \neq 0$ for at least one graph. Table 2 contains the non-zero numbers M(m,4) for the Steiner graphs which are not distinguished by their M(m,5). Tables 3 and 4 contain these numbers for the Latin square graphs of order 36. The tables are computed from the data listed in Chapter III, by use of the program Equivalence which is described in Chapter IV. The time needed by the computer EL-X8 for the computation for one graph of the sequence of numbers M(m,5), or M(m,4), is from 4 to 6 minutes.

Table 5 contains, for each Steiner triple system on 15 symbols, the number of one-factors, that is, the number of subsystems of 5 triples in which every symbol occurs once. It is a by-product of the present investigation.

Table 6 lists, for each Steiner triple system on 15 symbols, a subsystem of 15 triples in which each symbol occurs three times. The time needed by EL-X8 for the computation of one subsystem is from 25 to 35 seconds.

Further numerical data occur in the text of Chapter I, namely, in Sections 5 and 6. The computation in Section 6 of a sequence $\alpha(\nu)$ and $\beta(\nu)$, for $\nu = 3,4,\ldots$, by use of the procedure Alphabeta, takes from 10 to 150 seconds.

Table 1. M(m,5) for all Steiner graphs

nr. \ m	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	22	24	26	28	32	40	56
1																												36
2																							12				24	
3									8														12			16		
4									4								8				8				16			
5																	18									18		
6													8				12						16					
7																	18									18		
8					8								4				6				8		10					
9			2				6				6				4		8		2		8							
10							6				6				12							12						
11							8				12		2		4		10											
12		1						9		3				12						11								
13					6								14						4			12						
14	2								12								6		4			12						
15					4				8				4				12				8							
16	8																						28					
17					6								14						4			12						
18					8				4				8		4				12									
19					4						12						20											
20		1		3		6		7		3		6		10														
21		2		6	3					6	1	18																
22					5		3	4	9		12			3														
23		2				10	2		4	6		4			2	6												
24	1		2		1	10	3		2	2	4	4	2		5													
25		2		2	2	4	3	4	2		5	2	2		2	6												
26	2				6		5	4			5		2	4			8											
27	1			4	3		8	4		2	8			6														
28		1	1	4	4	5	1	6	2	2	2	2	6															
29	1		3		3	9	1			3		6	4	6														
30				4		6	3	6		10			7															
31						9				18								9										
32		1	4		1	10	1	2	5	6	3	3																
33		1	1	6	6	4	3	5	6	2	2																	
34		2	4	3	7	4	5	2	3	3	3																	
35	2	3		6	6	3	4		12																			
36		2	4			8	4			8	4	6																

nr. \ m	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	22	24	26	28	32	40	56
37	12					2		18				4																
38		3		8	10		7	5	3																			
39		2		6	7	2		6	9	2	2																	
40	2	1	3	5	1	1	4		7	3	9																	
41		3	4	7	1	2	2	5	3	9																		
42		4	4	5	4	4	3	2	5	5																		
43				4		6	3	6		10			7															
44		2	6		10	4	2	2	10																			
45			4	8	8		5	6	1	4																		
46			5	13	4	6	5	1	1		1																	
47		5	5	4	1	1	4	3	7	5	1																	
48		2	4	5	6	4	3	7	3		2																	
49			6	6	10	3	7	2			1	1																
50				10	7	1		11	5	2																		
51	1	1	5	6	6	5	8	2	2																			
52	3	1	3	6	5	16	1	1																				
53		2	3	3	6	7	9	5		1																		
54		1	6	1	6	4	6	4	6		2																	
55	1	2	7	1	6	6	7	3	2		1																	
56		3	3	8	7	9	4	2																				
57			2	6	4	13	5	2		2	1	1																
58			2	2	6	6	6	5	3	4	1	1																
59	1	3	3	5		6	6	6	6																			
60	1	7	4	1	2	3	9	2	4	3																		
61								15		21																		
62	3	6		3	3	12	6	3																				
63	1				9		8		6		12																	
64				1	9	6	11	9																				
65		1	7	4	8	8	5	3																				
66			5	8	10	7	4		1	1																		
67		1	6	7	11	5	3	2		1																		
68		4	5	7	7	7	4	1		1																		
69		3	8	5	10	5	3	1	1																			
70		3	4	5	3	9	7	3	2																			
71	1	5	8	4	6	5	6		1																			
72		2	5	8	9	8	2	2																				

nr. \ m	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	22	24	26	28	32	40	56
73		4	4	4	4	6		2	4	7				1														
74	4			8			8	4		8		4																
75			9		3		12		6		6																	
76		1	5	5	5	5		10		5																		
77	3	6	21		6																							
78				8	4	8	4		4	4		4																
79						9				18								9										
80						30						6																

Table 2. M(m,4) for some Steiner graphs

nr. \ m	180	188	196	200	204	206	208	210	214	220	224	228
5							8				12	16
7							20			16		
13		1	11		4		4	4	12			
17	1	3	12	4		4		12				

nr. \ m	181	183	185	187	189	191	193	195	197	199
30		2	7	9	4	4	2	3	5	
43	6	12	1	6	3	1	6	1		
31				4	4	4	10		12	2
79	10			6	18		2			

Table 3. M(m,5) for all Latin square graphs

nr. \ m	0	1	2	3	4	5	6	7	8	9	10	11	12	14	20
81							4	8	8		4	4	8		
82			12	4			8			12					
83	3	8			1	12	12								
84					6				30						
85	12	24													
86	6	12	18												
87															36
88									18					18	
89															36
90			36												
91					6				30						
92			36												

Table 4. M(m,4) for some Latin square graphs

nr. \ m	130	138	140	142	145	146
84		10		24		2
87			36			
89			36			
90					36	
91	6	30				
92	36					

Table 5. The number of one-factors in each Steiner system

nr.	number of 1-factors	nr.	number of 1-factors	nr.	number of 1-factors
1	56	28	2	55	2
2	24	29	0	56	1
3	8	30	3	57	4
4	8	31	5	58	3
5	16	32	2	59	0
6	12	33	1	60	6
7	32	34	1	61	7
8	4	35	0	62	0
9	2	36	1	63	6
10	6	37	5	64	3
11	6	38	4	65	2
12	1	39	1	66	3
13	4	40	0	67	4
14	0	41	1	68	1
15	8	42	5	69	4
16	0	43	3	70	2
17	12	44	1	71	2
18	4	45	2	72	4
19	16	46	2	73	9
20	1	47	1	74	3
21	1	48	1	75	6
22	4	49	2	76	1
23	1	50	7	77	1
24	0	51	2	78	9
25	1	52	0	79	17
26	0	53	1	80	11
27	3	54	2		

nr. 11

1	2	3
1	4	5
1	6	7
2	8	10
2	9	11
3	9	12
3	13	14
4	8	13
4	10	12
5	8	15
5	10	14
6	11	13
6	12	15
7	9	15
7	11	14

nr. 12

1	2	3
1	4	5
1	6	7
2	8	10
2	9	11
3	9	12
3	13	14
4	8	13
4	10	12
5	8	15
5	10	14
6	9	15
6	11	14
7	11	13
7	12	15

nr. 13

1	2	3
1	4	5
1	6	7
2	8	10
2	9	11
3	9	12
3	13	14
4	8	13
4	9	15
5	10	13
5	11	12
6	8	14
6	12	15
7	10	14
7	11	15

nr. 14

1	2	3
1	4	5
1	6	7
2	8	10
2	9	11
3	9	12
3	13	14
4	8	13
4	9	15
5	8	14
5	12	15
6	10	13
6	11	12
7	10	14
7	11	15

nr. 15

1	2	3
1	4	5
1	6	7
2	8	10
2	9	11
3	8	12
3	9	13
4	8	15
4	13	14
5	10	15
5	11	13
6	9	14
6	12	15
7	10	12
7	11	14

nr. 16

1	2	3
1	4	5
1	6	7
2	8	10
2	9	11
3	8	12
3	9	13
4	8	15
4	9	14
5	12	15
5	13	14
6	10	12
6	11	13
7	10	15
7	11	14

nr. 17

1	2	3
1	4	5
1	6	7
2	8	10
2	9	11
3	8	12
3	9	13
4	8	15
4	9	14
5	10	15
5	13	14
6	11	13
6	12	15
7	10	12
7	11	14

nr. 18

1	2	3
1	4	5
1	6	7
2	8	10
2	9	11
3	8	12
3	9	13
4	8	15
4	13	14
5	9	14
5	12	15
6	10	12
6	11	13
7	10	14
7	11	15

nr. 19

1	2	3
1	4	5
1	6	7
2	8	10
2	9	11
3	8	12
3	9	14
4	8	15
4	11	13
5	11	14
5	12	15
6	10	12
6	13	14
7	9	13
7	10	15

nr. 20

1	2	3
1	4	5
1	6	7
2	8	10
2	9	11
3	8	12
3	9	14
4	8	15
4	13	14
5	9	13
5	12	15
6	10	13
6	11	15
7	10	12
7	11	14

nr. 21

1	2	3
1	4	5
1	6	7
2	8	10
2	9	12
3	10	15
3	12	14
4	9	14
4	11	15
5	8	13
5	11	12
6	9	10
6	11	13
7	8	15
7	13	14

nr. 22

1	2	3
1	4	5
1	6	7
2	8	10
2	9	12
3	10	15
3	12	14
4	9	14
4	11	15
5	8	15
5	13	14
6	8	13
6	11	12
7	9	10
7	11	13

nr. 23

1	2	3
1	4	5
1	6	7
2	8	10
2	9	11
3	9	12
3	13	14
4	9	14
4	11	15
5	11	13
5	12	15
6	8	12
6	10	13
7	8	15
7	10	14

nr. 24

1	2	3
1	4	5
1	6	7
2	8	10
2	9	11
3	9	12
3	13	14
4	8	13
4	10	12
5	9	15
5	11	14
6	8	15
6	10	14
7	11	13
7	12	15

nr. 25

1	2	3
1	4	5
1	6	7
2	8	10
2	9	11
3	9	12
3	13	14
4	8	13
4	10	14
5	6	15
5	10	12
6	9	14
7	11	13
7	12	15
8	11	15

nr. 26

1	2	3
1	4	5
1	6	7
2	8	10
2	9	11
3	6	12
3	13	14
4	8	13
4	10	15
5	9	14
5	11	13
6	10	14
7	9	15
7	11	12
8	12	15

nr. 27

1	2	3
1	4	5
1	6	7
2	8	10
2	9	11
3	6	12
3	9	13
4	8	14
4	12	15
5	9	15
5	13	14
6	10	15
7	10	13
7	11	14
8	11	12

nr. 28

1	2	3
1	4	5
1	6	7
2	8	10
2	9	11
3	6	12
3	9	13
4	8	14
4	12	15
5	9	15
5	13	14
6	10	13
7	10	14
7	11	15
8	11	12

nr. 29

1	2	3
1	4	5
1	6	7
2	8	10
2	9	11
3	6	12
3	9	13
4	8	15
4	10	14
5	12	15
5	13	14
6	10	13
7	9	14
7	11	15
8	11	12

nr. 30

1	2	3
1	4	5
1	6	7
2	8	10
2	9	11
3	6	12
3	9	14
4	11	13
4	12	15
5	10	15
5	13	14
6	10	14
7	8	15
7	9	13
8	11	12

nr. 31

1	2	3
1	4	5
1	6	7
2	8	10
2	9	11
3	6	12
3	9	14
4	10	15
4	11	13
5	12	15
5	13	14
6	10	14
7	8	15
7	9	13
8	11	12

nr. 32

1	2	3
1	4	5
1	6	7
2	8	10
2	9	11
3	6	12
3	9	14
4	8	13
4	12	15
5	9	15
5	13	14
6	10	15
7	10	14
7	11	13
8	11	12

nr. 33

1	2	3
1	4	5
1	6	7
2	8	10
2	9	11
3	6	12
3	9	14
4	8	13
4	12	15
5	9	15
5	13	14
6	10	14
7	10	13
7	11	15
8	11	12

nr. 34

1	2	3
1	4	5
1	6	7
2	8	10
2	9	11
3	6	12
3	9	14
4	10	13
4	11	15
5	9	15
5	13	14
6	10	14
7	8	13
7	12	15
8	11	12

nr. 35

1	2	3
1	4	5
1	6	7
2	8	10
2	12	14
3	5	9
3	6	12
4	10	15
4	11	13
5	10	13
6	11	14
7	8	11
7	9	15
8	12	15
9	13	14

nr. 36

1	2	3
1	4	5
1	6	7
2	4	6
2	9	11
3	10	15
3	11	14
4	10	13
5	8	13
5	11	12
6	8	14
7	9	15
7	10	12
8	12	15
9	13	14

nr. 37

1	2	3
1	4	5
1	6	7
2	8	10
2	9	11
3	6	12
3	7	13
4	9	14
4	12	15
5	8	15
5	13	14
6	10	14
7	11	15
8	11	12
9	10	13

nr. 38

1	2	3
1	4	5
1	6	7
2	4	6
2	9	11
3	10	13
3	11	15
4	12	15
5	8	15
5	10	14
6	9	14
7	8	12
7	11	13
8	13	14
9	10	12

nr. 39

1	2	3
1	4	5
1	6	7
2	4	6
2	9	11
3	10	13
3	11	15
4	12	15
5	8	12
5	13	14
6	10	14
7	8	13
7	9	15
8	11	14
9	10	12

nr. 40

1	2	3
1	4	5
1	6	7
2	8	10
2	12	14
3	4	8
3	11	15
4	7	13
5	10	14
5	11	12
6	9	10
6	11	13
7	9	15
8	12	15
9	13	14

nr. 41

1	2	3
1	4	5
1	8	9
2	4	6
2	8	10
3	6	12
3	11	15
4	13	14
5	11	13
5	12	15
6	11	14
7	8	13
7	9	15
7	10	12
9	10	14

nr. 42

1	2	3
1	4	5
1	6	7
2	8	10
2	9	11
3	6	12
3	7	14
4	9	13
4	12	15
5	8	15
5	13	14
6	10	13
7	11	15
8	11	12
9	10	14

nr. 43

1	2	3
1	4	5
1	6	7
2	4	6
2	9	11
3	10	15
3	11	13
4	13	14
5	10	14
5	12	15
6	8	15
7	8	12
7	9	13
8	11	14
9	10	12

nr. 44

1	2	3
1	4	5
1	6	7
2	4	6
2	9	11
3	10	15
3	11	13
4	10	14
5	8	14
5	11	12
6	8	13
7	9	15
7	10	12
8	12	15
9	13	14

nr. 45

1	2	3
1	4	5
1	6	7
2	8	10
2	12	14
3	10	13
3	11	14
4	9	12
4	11	15
5	6	15
5	8	14
6	9	13
7	8	13
7	11	12
9	10	15

nr. 46

1	2	3
1	4	5
1	6	7
2	4	6
2	12	14
3	10	13
3	11	14
4	11	13
5	8	12
5	10	15
6	9	15
7	8	14
7	9	13
8	11	15
9	10	12

nr. 47

1	2	3
1	4	5
1	6	7
2	4	6
2	13	15
3	10	14
3	11	13
4	9	13
5	8	12
5	10	15
6	8	14
7	9	14
7	11	12
8	11	15
9	10	12

nr. 48

1	2	3
1	4	5
1	6	7
2	4	6
2	9	11
3	9	13
3	11	14
4	12	15
5	8	12
5	13	14
6	10	15
7	8	14
7	10	13
8	11	15
9	10	12

nr. 49

1	2	3
1	4	5
1	6	7
2	8	10
2	12	14
3	7	15
3	9	13
4	9	15
4	11	13
5	8	14
5	9	12
6	8	11
6	10	15
7	11	12
10	13	14

nr. 50

1	2	3
1	4	5
1	6	7
2	8	10
2	12	14
3	7	15
3	9	13
4	11	15
4	13	14
5	8	14
5	9	12
6	8	11
6	10	13
7	11	12
9	10	15

nr. 51

1	2	3
1	4	5
1	6	7
2	4	6
2	13	15
3	9	13
3	10	14
4	11	13
5	8	15
5	10	12
6	8	14
7	9	12
7	11	14
8	11	12
9	10	15

nr. 52

1	2	3
1	4	5
1	6	7
2	4	6
2	13	15
3	9	13
3	10	14
4	11	12
5	8	15
5	10	12
6	8	14
7	9	10
7	11	14
8	11	13
9	12	15

nr. 53

1	2	3
1	4	5
1	6	7
2	4	6
2	13	15
3	9	13
3	10	14
4	11	13
5	10	12
5	8	15
6	8	14
7	9	10
7	11	14
8	11	12
9	12	15

nr. 54

1	2	3
1	4	5
1	6	7
2	4	6
2	8	10
3	9	13
3	10	14
4	9	15
5	9	12
5	13	14
6	11	15
7	8	14
7	11	13
8	11	12
10	12	15

nr. 55

1	2	3
1	4	5
1	6	7
2	4	6
2	8	10
3	9	13
3	10	14
4	12	15
5	9	12
5	13	14
6	11	15
7	8	14
7	11	13
8	11	12
9	10	15

nr. 56

1	2	3
1	4	5
1	6	7
2	8	10
2	12	14
3	7	15
3	10	13
4	9	13
4	11	12
5	6	9
5	8	15
6	10	14
7	11	13
8	11	14
9	12	15

nr. 57

1	2	3
1	4	5
1	6	7
2	4	6
2	12	14
3	9	14
3	10	13
4	9	13
5	8	15
5	10	12
6	11	15
7	8	14
7	11	13
8	11	12
9	10	15

nr. 58

1	2	3
1	4	5
1	6	7
2	8	10
2	12	14
3	5	11
3	9	14
4	9	13
4	10	15
5	10	12
6	8	13
6	11	15
7	8	14
7	11	13
9	12	15

nr. 59

1	2	3
1	4	5
1	6	7
2	8	10
2	9	11
3	5	15
3	9	13
4	9	12
4	11	14
5	6	14
6	10	13
7	8	15
7	10	12
8	13	14
11	12	15

nr. 60

1	2	3
1	4	5
1	6	7
2	4	6
2	13	15
3	9	13
3	10	14
4	10	12
5	9	10
5	11	14
6	8	11
7	8	15
7	9	12
8	13	14
11	12	15

nr. 61

1	2	3
1	4	5
1	6	7
2	8	10
2	9	12
3	8	11
3	9	15
4	10	15
4	13	14
5	10	14
5	11	12
6	11	13
6	12	15
7	8	13
7	9	14

nr. 62

1	2	3
1	4	5
1	6	7
2	8	10
2	9	12
3	6	11
3	10	13
4	11	15
4	12	14
5	9	15
5	10	14
6	9	13
7	8	15
7	13	14
8	11	12

nr. 63

1	2	3
1	4	5
1	6	7
2	8	10
2	9	12
3	10	13
3	12	14
4	11	13
4	12	15
5	10	14
5	11	15
6	8	13
6	9	14
7	8	15
7	9	11

nr. 64

1	2	3
1	4	5
1	6	7
2	9	12
2	11	14
3	5	8
3	9	14
4	8	15
4	11	12
5	10	13
6	9	10
6	12	15
7	10	15
7	13	14
8	11	13

nr. 65

1	2	3
1	4	5
1	6	7
2	5	7
2	11	14
3	10	13
3	12	15
4	9	10
4	13	14
5	8	15
6	8	11
6	9	13
7	10	12
8	12	14
9	11	15

nr. 66

1	2	3
1	4	5
1	6	7
2	4	6
2	8	10
3	9	13
3	10	15
4	11	12
5	10	14
5	11	13
6	12	15
7	8	13
7	9	14
8	12	14
9	11	15

nr. 67

1	2	3
1	4	5
1	6	7
2	4	6
2	8	10
3	9	14
3	11	15
4	11	12
5	9	13
5	12	14
6	9	15
7	8	14
7	10	13
8	11	13
10	12	15

nr. 68

1	2	3
1	4	5
1	6	7
2	5	7
2	13	15
3	10	13
3	12	15
4	9	10
4	11	12
5	10	14
6	8	13
6	9	14
7	8	11
8	12	14
9	11	15

nr. 69

1	2	3
1	4	5
1	6	7
2	8	10
2	9	12
3	7	13
3	9	14
4	11	12
4	13	14
5	6	11
5	8	15
6	9	15
7	10	14
8	11	13
10	12	15

nr. 70

1	2	3
1	4	5
1	6	7
2	5	7
2	8	10
3	9	15
3	10	13
4	11	13
4	12	14
5	9	11
6	9	14
6	10	15
7	8	12
8	13	14
11	12	15

nr. 71

1	2	3
1	4	5
1	6	7
2	5	7
2	13	15
3	9	14
3	10	15
4	9	10
4	12	14
5	9	11
6	8	12
6	10	13
7	8	11
8	13	14
11	12	15

nr. 72

1	2	3
1	4	5
1	6	7
2	4	6
2	9	12
3	9	11
3	10	15
4	12	14
5	8	15
5	9	13
6	10	14
7	8	11
7	10	13
8	13	14
11	12	15

nr. 73

1	2	3
1	4	5
1	6	7
2	4	6
2	9	12
3	9	14
3	11	15
4	11	12
5	8	15
5	9	10
6	13	14
7	8	14
7	10	13
8	11	13
10	12	15

nr. 74

1	2	3
1	4	5
1	8	9
2	4	6
2	13	15
3	7	14
3	9	13
4	10	12
5	8	15
5	12	14
6	8	11
6	10	14
7	9	11
7	10	13
11	12	15

nr. 75

1	2	3
1	4	5
1	6	7
2	5	7
2	9	12
3	9	11
3	10	15
4	9	14
4	11	13
5	8	15
6	8	12
6	10	13
7	10	14
8	13	14
11	12	15

nr. 76

1	2	3
1	4	5
1	6	7
2	4	6
2	13	15
3	10	13
3	12	15
4	9	10
5	8	12
5	9	11
6	8	14
7	10	14
7	11	12
8	11	15
9	13	14

nr. 77

1	2	3
1	4	5
1	6	7
2	5	7
2	9	12
3	10	15
3	11	12
4	9	10
4	11	13
5	8	13
6	8	14
6	12	15
7	10	14
8	11	15
9	13	14

nr. 78

1	2	3
1	4	5
1	6	7
2	4	6
2	9	12
3	10	15
3	12	14
4	9	11
5	8	13
5	12	15
6	8	14
7	10	14
7	11	13
8	11	15
9	10	13

nr. 79

1	2	3
1	4	5
1	6	7
2	8	10
2	9	12
3	5	10
3	12	14
4	9	15
4	13	14
5	11	13
6	8	13
6	11	12
7	9	14
7	10	15
8	11	15

nr. 80

1	2	3
1	4	5
1	6	7
2	10	12
2	11	14
3	8	15
3	9	13
4	7	12
4	8	13
5	6	14
5	11	13
6	9	15
7	10	15
8	10	14
9	11	12