November 20, 1972

To the Graduate Council:

I am submitting herewith a thesis written by David Hume entitled "Applying the Computer to Binary Quadratic Forms." I recommend that it be accepted for nine quarter hours of credit in partial fulfillment of the requirements for the degree of Master of Arts, with a major in Mathematics.

Robert M McConnell
Major Professor

We have read this thesis
and recommend its acceptance:

Carl G. Wagner

Julius Smith

Accepted for the Council:

Arthur A. Smith
Vice Chancellor for
Graduate Studies and Research

APPLYING THE COMPUTER TO BINARY QUADRATIC FORMS

A Thesis

Presented to

the Graduate Council of

The University of Tennessee

In Partial Fulfillment

of the Requirements for the Degree

Master of Arts

---

by

David Hume

March 1973

ACKNOWLEDGMENTS

This thesis is dedicated to my parents and my algebra teachers, Miss Elizabeth Brixey, Miss Lucille Evans, Mrs. Lucille Lee, and Dr. Robert McConnel. I am indebted to Mrs. Eunice Hinkle for typing the entire manuscript.

1063494

ABSTRACT

Binary quadratic forms over the integers are considered. Many theorems from the book [L. E. Dickson, Introduction to the Theory of Numbers, Chicago, 1929] are stated without proof. Several computer programs based on these theorems are given and explained; sample output is provided. These programs concern 1) representation of integers by definite and indefinite forms, 2) display of definite and indefinite forms by genera and indefinite forms by periods, 3) display as a finite abelian group of the set of all primitive forms of given discriminant $d<0$.

TABLE OF CONTENTS

CHAPTER		PAGE
I.	REPRESENTATION BY BINARY QUADRATIC FORMS	1
	Introduction	1
	Definite Forms	2
	Indefinite Forms	7
II.	CLASSIFICATION OF BINARY QUADRATIC FORMS	17
	Classification of Indefinite Forms into Periods . .	17
	Classification into Genera	20
III.	COMPOSITION OF BINARY QUADRATIC FORMS	26
	Composition Defined on Equivalence Classes	26
	Group Structure Using Prime Power Order Generators .	29
	Group Structure Using Minimal Number of Generators .	37
IV.	OBSERVATIONS	42
BIBLIOGRAPHY		43
APPENDIXES		45
VITA		101

LIST OF TABLES

TABLE		PAGE
I.	Representation Program for Definite Forms	46
II.	Subroutine REDUCE for Definite Forms	48
III.	Representation Program for Indefinite Forms	50
IV.	Subroutine REDUCE for Indefinite Forms	54
V.	Program for Listing Reduced Indefinite Forms by Periods	57
VI.	MAIN PROGRAM for Genera Program	61
VII.	Subroutine DPRFAC for Genera Program	64
VIII.	Subroutine THM13	66
IX.	Subroutine CRTHM	67
X.	Subroutine LEGNDR	69
XI.	Group Structure Program Using Prime Power Order Generators, MAIN PROGRAM	70
XII.	Subroutine DPRFAC for Group Structure Program	74
XIII.	Subroutine SCAN	75
XIV.	Subroutine SWITCH	76
XV.	Subroutine CMPOSE	77
XVI.	Subroutine GCD	79
XVII.	Group Structure Program Using Minimal Number of Generators, MAIN PROGRAM	80
XVIII.	Subroutine BASE	83
XIX.	Sample Results from Representation Program for Definite Forms	85

TABLE PAGE

XX. Sample Results from Representation Program for Indefinite Forms 86

XXI. Sample Output from Periods Program 87

XXII. Sample Output from Genera Program for Definite Forms 90

XXIII. Sample Output from Genera Program for Indefinite Forms 93

XXIV. Sample Output from Group Structures Programs 95

XXV. Program for Listing Reduced Definite Forms 99

CHAPTER I

REPRESENTATION BY BINARY QUADRATIC FORMS

I. INTRODUCTION

Definition 1. A binary quadratic form is any expression of the form $ax^2 + bxy + cy^2$ where a,b, and c are constants. [a,b,c] denotes the form $ax^2 + bxy + cy^2$.

We shall restrict our attention henceforth to integral forms in which a,b, and c are integers.

Definition 2. A form q = [a,b,c] represents an integer m if there are integers α and γ such that $a\alpha^2 + b\alpha\gamma + c\gamma^2 = m$. The representation is proper if α and γ are relatively prime. (α,γ) denotes this representation of m by q.

Definition 3. The divisor of form [a,b,c] is the greatest common divisor of its coefficients a,b, and c. A form is primitive if its divisor is 1.

Definition 4. The discriminant of form [a,b,c] is $d = b^2 - 4ac$.

Definition 5. A form [a,b,c] is definite if $d<0$. In this case, the form represents only positive numbers (positive form) or only negative numbers (negative form). If $d>0$, form [a,b,c] is indefinite and represents positive and negative numbers. (2, p. 67).

Definition 6. Two forms $q = ax^2 + bxy + cy^2$ and $Q = AX^2 + BXY + CY^2$ are equivalent if there is a linear transformation $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix}$ of determinant 1, whose matrix has integral coefficients, and such that if $x = \alpha X + \beta Y$, $y = \gamma X + \delta Y$ are substituted into form q,

the form Q results. The notation

$$[a,b,c] \xrightarrow{\begin{pmatrix}\alpha & \beta\\ \gamma & \delta\end{pmatrix}} [A,B,C]$$

means that [a,b,c] is equivalent to [A,B,C] under the transformation $\begin{pmatrix}\alpha & \beta\\ \gamma & \delta\end{pmatrix}$. In this case, $\begin{pmatrix}\alpha & \beta\\ \gamma & \delta\end{pmatrix}$ takes q to Q, or transforms q to Q.

As defined above, equivalence of forms is an equivalence relation on the set of all binary quadratic forms. Two equivalent forms have the same discriminant and represent the same numbers. (2, pp. 63 - 66).

Definition 7. An automorph A of form q = [a,b,c] is an integral transformation of determinant 1 which leaves q fixed. That is,

$$[a,b,c] \xrightarrow{A} [a,b,c]$$

II. DEFINITE FORMS

We shall restrict our attention to positive definite forms. Properties of negative definite forms follow at once from those of positive forms.

Definition 8. A positive form [a,b,c] is reduced if $-a<b\leq a$ and $a\leq c$, with $b\geq 0$ if $a = c$.

Theorem 1. Every positive form is equivalent to a reduced form. (2, p. 68).

Theorem 2. Equivalent positive reduced forms are identical. (2, p. 71).

Theorem 3. For given discriminant $d<0$, there is only a finite number of reduced integral forms. (2, p. 71).

In light of the preceding three theorems, the set of all integral forms of given discriminant $d<0$ breaks up into a finite number of mutually disjoint equivalence classes, each containing a single reduced form.

Theorem 4. The automorphs of $ax^2 + ay^2$ are

$$T = \begin{pmatrix} \pm 1 & 0 \\ 0 & \pm 1 \end{pmatrix}, \quad S = \begin{pmatrix} 0 & \pm 1 \\ \mp 1 & 0 \end{pmatrix}.$$

The automorphs of $ax^2 + axy + ay^2$ are

$$T \quad , R = \begin{pmatrix} 0 & \mp 1 \\ \pm 1 & \pm 1 \end{pmatrix}, \quad R^{-1} = \begin{pmatrix} \pm 1 & \pm 1 \\ \mp 1 & 0 \end{pmatrix}.$$

For all other positive forms, the only automorphs are T. (2, p.72).

We turn to the following problem: given a positive form $q = [a,b,c]$ and integer m, find all proper representations of m by q, if any. In this representation problem, it is sufficient to consider only primitive forms: if the divisor of $[a,b,c]$ is $g>1$, then $ax^2 + bxy + cy^2 = g(a_1x^2 + b_1xy + c_1y^2)$ where $[a_1,b_1,c_1]$ is primitive; the problem of representing m by $[a,b,c]$ is the same as representing m_1 by $[a_1, b_1, c_1]$ where $m = gm_1$.

Theorem 5. Given form $q = [a,b,c]$ of discriminant d, an integer m, and a proper representation (α,γ) of m by q, then integers β,δ,n can be determined in one and only one way so that $n^2 \equiv d \pmod{4m}$, $0 \le n < 2m$, and

$$[a,b,c] \xrightarrow{\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}} [m,n,\ell]$$

Definition 9. In this case, the representation (α,γ) belongs to the root n of the congruence $n^2 \equiv d \pmod{4m}$.

Proof of Theorem 5. Suppose (α,γ) is a proper representation of m by q. Then there are integers β',δ' such that $\alpha\delta' - \gamma\beta' = 1$. If $x = \delta$, $y = \beta$ is any integral solution to $\alpha x - \gamma y = 1$, then $\alpha(\delta - \delta') = \gamma(\beta - \beta')$. Thus, α divides $\beta - \beta'$, γ divides $\delta - \delta'$, and $\alpha(\delta - \delta') = \alpha\gamma t = \gamma(\beta - \beta')$ where t is an integer; hence

$$(1) \qquad \begin{aligned} \beta &= \beta' + t\alpha \\ \delta &= \delta' + t\gamma \end{aligned} .$$

Now

$$[a,b,c] \xrightarrow{\begin{pmatrix}\alpha & \beta\\ \gamma & \delta\end{pmatrix}} [a\alpha^2 + b\alpha\gamma + c\gamma^2,\ 2a\alpha\beta + b(\alpha\delta + \beta\gamma) + 2c\gamma\delta,\ a\beta^2 + b\beta\delta + c\delta^2]$$

$$= [m,n,\ell].$$

Let $n' = 2a\alpha\beta' + b(\alpha\delta' + \beta'\gamma) + 2c\gamma\delta'$. Then

$$\begin{aligned} n - n' &= 2a\alpha(\beta - \beta') + b(\alpha(\delta - \delta') + \gamma(\beta - \beta')) + 2c\gamma(\delta - \delta') \\ &= 2a\alpha^2 t + b(2\alpha\gamma t) + 2c\gamma^2 t \\ &= 2t(a\alpha^2 + b\alpha\gamma + c\gamma^2) = 2tm. \end{aligned}$$

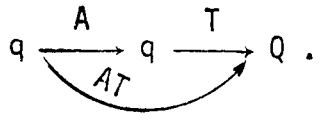
Thus, $n = n' + 2mt$ and there is a single integer t such that $0 \le n < 2m$. By (1), this value of t determines values β and δ. Thus, β,δ,n can be determined in exactly one way such that

$$[a,b,c] \xrightarrow{\begin{pmatrix}\alpha & \beta\\ \gamma & \delta\end{pmatrix}} [m,n,\ell]$$

where $0 \le n < 2m$ and $n^2 - 4m\ell = d$. ##

For each root n of $n^2 \equiv d \pmod{4m}$, compute $Q = [m,n,\ell]$ as in Theorem 5: if q is not equivalent to Q, there are no proper representations of m belonging to that root. Suppose $q \xrightarrow{T} Q$; then all

transformations which take q to Q have form AT where A is an automorph of q:



The first column of each AT gives a proper representation of m by q belonging to root n. In this way, obtain all proper representations. (2, pp. 73, 74).

The components of the representation program for definite forms are:

MAIN PROGRAM (See Table I, Appendix)

SUBROUTINE REDUCE (See Table II, Appendix).

The MAIN PROGRAM employs the method of Theorem 5.

Statements 3-6: Read in all automorphs given by Theorem 4 in the following order: $S,S,T,T,R,R,R^{-1},R^{-1}$.
The I^{th} automorph is stored as $\begin{pmatrix} R(I) & S(I) \\ T(I) & U(I) \end{pmatrix}$.

Statement 9: Read in form [A,B,C]. (This program will accept non-primitive forms.)

Statement 15: Call subroutine REDUCE which transforms [A,B,C] to the unique equivalent reduced form [AR,BR,CR] and returns the 2X2 matrix M1 representing that transformation:

$$[A,B,C] \xrightarrow{M1} [AR,BR,CR].$$

Statements 16-25: Specify which automorphs will be used with the form [A,B,C], using Theorem 4. The variables K1 and K2 reappear in Statement 45.

Statement 26: Read in value of m, a number to be represented properly by [A,B,C].

Statements 30-37: Set up congruence $n^2 \equiv d \pmod{4m}$ and scan for roots n where $0 \leq n < 2m$. When a root N is found, jump to

Statement 38: Compute [M,N,L].

Statement 39: Call REDUCE again:

$$[M,N,L] \xrightarrow{M2} [MR,NR,LR].$$

Statements 40-42: If $[MR,NR,LR] \neq [AR,BR,CR]$, [A,B,C] is not equivalent to [M,N,L] and there is no proper representation of m belonging to the root N; otherwise,

Statements 43, 44: $[A,B,C] \xrightarrow{M1} [AR,BR,CR] = [MR,NR,LR] \xleftarrow{M2} [M,N,L]$. [A,B,C] is equivalent to [M,N,L] under $M1 \cdot M2^{-1}$. Compute the first column of $TR = M1 \cdot M2^{-1}$.

Statements 45-50: Use each appropriate automorph A to compute the first column of $A \cdot TR$ and obtain each proper representation (X,Y) of m belonging to the root N.

Statement 51: Search for next root N.

Subroutine REDUCE (See Table II, Appendix) is based on Gauss's proof of Theorem 1 for integral forms. (2, p.69) This subroutine receives a form and returns the equivalent reduced form with the matrix M of transformation.

Statements 1-9: Start with form [A,B,C] and initialize transformation matrix to $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

Statements 10, 11: Store BR = TEMP1 and CR = TEMP2 in temporary locations.

Statements 13, 14: Divide -BR by 2·CR to obtain integral quotient S and remainder which will be new BR.

Statements 15-21: If it is not already the case, make $|BR| \leq CR$.

Statements 22, 23: Apply transformation $\begin{pmatrix} 0 & 1 \\ -1 & S \end{pmatrix}$ to [AR,TEMP1,TEMP2] to get [AR,BR,CR] in which $|BR| \leq AR$.

Statements 24-30: Compute matrix of transformation that has just occurred.

Statement 31: If $CR \leq AR$, go to statement 10 and repeat process. $CR \geq AR$ will hold after a finite number of passes.

Statements 32-35: If $BR \neq -AR$, go to statement 36; otherwise, apply transformation $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$.

Statement 36, 37: If $AR \neq CR$ or $BR \geq 0$, RETURN.

Statements 38-45: To reach statement 38, a form must look like [a,b,a] with $b<0$; apply transformation $\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$ taking [a,b,a] to [a,-b,a], a reduced form, and update matrix before returning.

Sample results from this program are listed in Table XIX, Appendix. In each case, all proper representations of m by q are given.

III. INDEFINITE FORMS

Definition 10. Given $q = [a,b,c]$ with $d = b^2 - 4ac > 0$, let $R = \sqrt{d}$. The first and second roots of q are $f = \frac{R-b}{2a}$, $s = \frac{-R-b}{2a}$

respectively. We assume that $a \neq 0$ and $\dot{R}$ is irrational (d is not a square).

Definition 11. The form $q = [a,b,c]$ is reduced if $|f|<1$, $|s|>1$, and $fs<0$.

Theorem 6. Every form is equivalent to a form $[a,b,c]$ in which $|b| \leq |a| \leq \sqrt{d/3}$. (2, p. 101).

Theorem 7. Every form is equivalent to a reduced form. (2, p. 101).

Definition 12. Given that $q = [a,b,c] \xrightarrow{\begin{pmatrix} 0 & 1 \\ -1 & \delta \end{pmatrix}} [c,b',c'] = Q$, Q is a right neighboring form to q and q is left neighboring to Q.

Theorem 8. Every reduced form q has unique left and right reduced neighboring forms. If $|\delta|$ is the largest integer $< \frac{1}{|f|}$, where f is the first root of q and δ has the same sign as a, then

$q = [a,b,c] \xrightarrow{\begin{pmatrix} 0 & 1 \\ -1 & \delta \end{pmatrix}} [c,b',c']$ gives the reduced right neighboring form to q. (2, p. 102).

Theorem 9. There is only a finite number of reduced integral forms of given discriminant $d>0$. (2, p. 103).

Thus, any reduced form lies in a finite chain of equivalent reduced forms:

$$\longrightarrow [c_1,b_2,c_2] \xrightarrow{\begin{pmatrix} 0 & 1 \\ -1 & \delta_1 \end{pmatrix}} [c_2,b_3,c_3] \xrightarrow{\begin{pmatrix} 0 & 1 \\ -1 & \delta_2 \end{pmatrix}} [c_3,b_4,c_4] \longrightarrow$$

$$\longrightarrow \begin{pmatrix} 0 & 1 \\ -1 & \delta_{2n} \end{pmatrix} [c_{2n},b_1\ c_1] \xleftarrow{\begin{pmatrix} 0 & 1 \\ -1 & \delta_{2n-1} \end{pmatrix}} [c_{2n-1},b_{2n},c_{2n}] \longleftarrow \cdots \longleftarrow$$

For a reduced form [a,b,c], a and c have opposite signs. (2, p. 101). Since the signs of the leading coefficients of successive forms in a chain alternate, each chain contains an even number of forms.

Definition 13. A finite chain of reduced forms is called a period.

Theorem 10. Two equivalent reduced forms of the same discriminant $d>0$ belong to the same chain. (2, p. 108).

Thus, the set of all forms of given discriminant $d>0$ breaks up into a finite number of mutually disjoint equivalence classes, each containing one chain of reduced forms.

Theorem 11. Given a primitive form $q = [a,b,c]$ of discriminant $d>0$, all automorphs $\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}$ of q are given by $\alpha = \frac{1}{2}(t - bu)$, $\beta = -cu$, $\gamma = au$, $\delta = \frac{1}{2}(t + bu)$ where (t,u) is any integral solution of $t^2 - du^2 = 4$. (2, p. 112).

Theorem 12. The equation $t^2 - du^2 = 4$ has a solution (t,u) with $u \neq 0$. (2, p. 113).

If (t,u) is a solution, then $(t, \pm u)$ and $(-t, \pm u)$ are all solutions of $t^2 - du^2 = 4$. Hence, it suffices to examine positive solutions (t,u) of which there is a minimal positive solution (T,U). If this minimal solution (T,U) yields automorph E of q, then automorphs E^n, $n = 1,2,3 \ldots$ yield infinitely many automorphs of q corresponding to all positive solutions (t,u). (2, p. 113-114).

The method of Theorem 5 may be used intact to find proper representations of an integer m by an indefinite form. The components of the representation program for indefinite forms are:

MAIN PROGRAM (See Table III, Appendix)

SUBROUTINE REDUCE (See Table IV, Appendix).

The MAIN PROGRAM employs the method of Theorem 5; it is necessary to use primitive forms of non-square discriminant d>0. The number to be represented need not be positive.

Statement 7: Read in form [A,B,C].

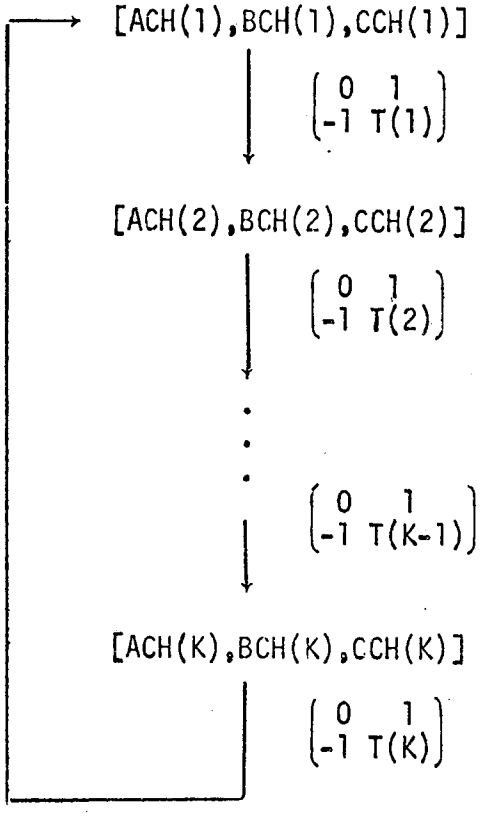
Statement 12: Calculate discriminant $D = B^2 - 4AC$; this value is passed to subroutine REDUCE.

Statement 13: Call REDUCE:

$$[A,B,C] \xrightarrow{M1} [ACH(1),BCH(1),CCH(1)].$$

Statements 14-16: Store 2X2 matrix M1 as M3.

Statements 17-27: Construct chain of K equivalent reduced forms containing [ACH(1),BCH(1),CCH(1)] as first form:



Statement 19: T(I) is the largest integer less than $1/|f|$ where f is the first root of [ACH(I),BCH(I),CCH(I)].

Statements 20-22: Compute right neighboring form by applying transformation $\begin{pmatrix} 0 & 1 \\ -1 & T(I) \end{pmatrix}$.

Statements 24, 25: Test to see if the end of the chain has been reached.

Statement 26: K is the number of distinct forms in the chain.

Statements 28-48: Find a solution (TT,U) to the equation $t^2 - du^2 = 4$ by the method of Theorem 12. This theorem requires a primitive form. For a chain of length 2n, compute the continued fraction expansion $\frac{\gamma}{\alpha} = (g_0, g_1, \cdots, g_{2n-1})$ where $g_i = |T(I)|$.

Statements 28-33: If the first coefficient of the first form in the chain is positive, use forms 1 through 2n-1. If the first coefficient is negative, use forms 2 through 2n. Theorem 12 requires that the first form have positive first coefficient.

Statements 34, 45, 46: If chain has length 2, proceed to short calculation of $\frac{\gamma}{\alpha}$ $\left(= \frac{Z}{W} \text{ in program}\right)$.

Statements 35-44: Calculate continued fraction expansion

$$\frac{\gamma}{\alpha} = g_1 + \cfrac{1}{g_2 + \cfrac{1}{g_3 + \ddots + \cfrac{1}{g_{2n-1}}}}$$

or

$$\frac{\gamma}{\alpha} = g_2 + \cfrac{1}{g_3 + \cfrac{1}{g_4 + \ddots + \cfrac{1}{g_{2n}}}}$$

as the case may be.

Statements 47, 48: Compute (TT,U) according to Theorem 11.

Statement 49: Read value of m to be properly represented by [A,B,C].

Statement 53: Since the number m may be negative, take |m| in computing MM.

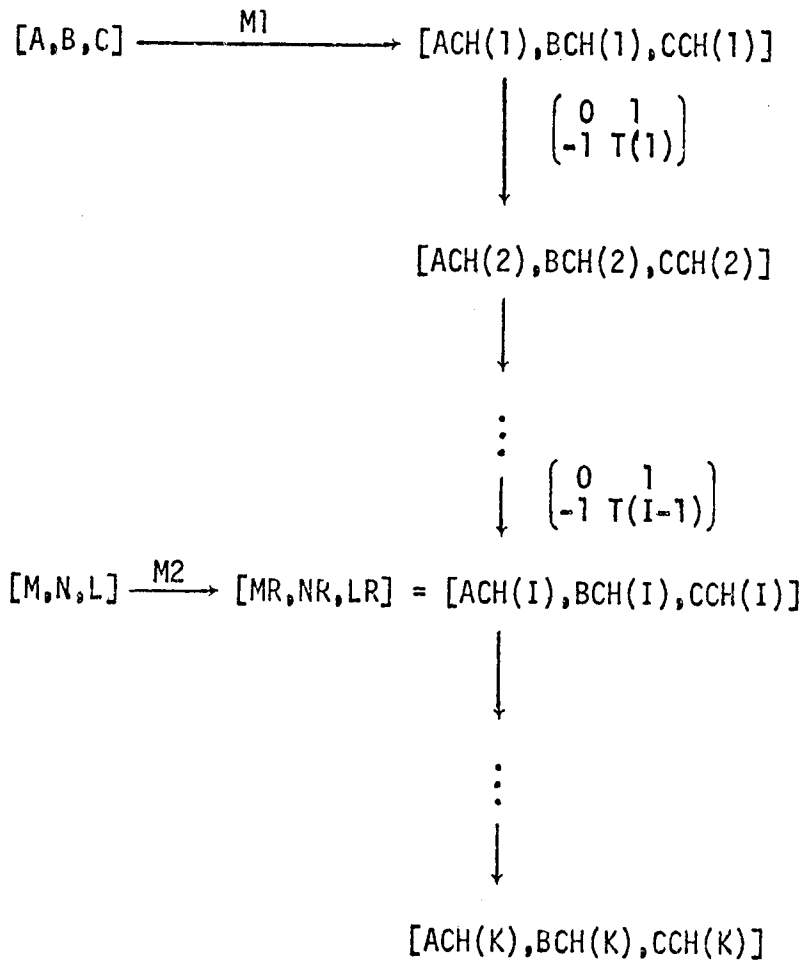
Statements 53-61: Set up congruence $n^2 \equiv d \pmod{4m}$ and search for roots n where $0 \leq n < 2|m| - 1$; for each root N, calculate [M,N,L].

Statement 62: Reduce [M,N,L] to [MR,NR,LR]:

$$[M,N,L] \xrightarrow{M2} [MR,NR,LR].$$

Statements 63-68: [M,N,L] is equivalent to [A,B,C] if and only if [MR,NR,LR] is equal to one of [ACH(I),BCH(I),CCH(I)].

Statement 69: If (M,N,L] is not equivalent to [A,B,C], go search for the next root N. Otherwise, we have the following situation:



Statements 70-81: Compute the matrix of the transformation taking [A,B,C] to [M,N,L].

Statement 70: If I = 1, skip to statement 80 to compute $M1 \cdot M2^{-1}$.

Statements 71-79: Compute matrix product $M1 \cdot \begin{pmatrix} 0 & 1 \\ -1 & T(1) \end{pmatrix} \cdot \ldots \cdot \begin{pmatrix} 0 & 1 \\ -1 & T(I-1) \end{pmatrix}$ and store product as M1.

Statements 80, 81: Compute the first column of $TR = M1 \cdot M2^{-1}$.

Statements 82-84: Restore the original value of matrix M1 (this was stored as matrix M3) to be used for next [M,N,L] computed for next root N of $n^2 \equiv d \pmod{4m}$.

Statements 85-90: The solutions $(\pm 2,0)$ of $t^2 - du^2 = 4$ are not covered by Theorem 12; they yield automorphs

$A = \begin{pmatrix} \pm 1 & 0 \\ 0 & \pm 1 \end{pmatrix}$ and two representations (X,Y) given by the first column

$$A \cdot TR = \begin{pmatrix} \pm 1 & 0 \\ 0 & \pm 1 \end{pmatrix} \begin{pmatrix} TR_{11} & -- \\ TR_{21} & -- \end{pmatrix} = \begin{pmatrix} X & -- \\ Y & -- \end{pmatrix}.$$

Statements 91-103: Compute 4 representations arising from solutions $(\pm TT, \pm U)$ of $t^2 - du^2 = 4$ $(U \neq 0)$.

Statements 91-94: Set up 4 combinations of sign and return TT,U unchanged for future use.

Statements 95-98: Compute $\alpha,\beta,\gamma,\delta$ according to Theorem 11.

Statements 99, 100: $\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \begin{pmatrix} TR_{11} & -- \\ TR_{21} & -- \end{pmatrix} = \begin{pmatrix} X & -- \\ Y & -- \end{pmatrix}$.

Statement 104: Go search for next root N.

Subroutine REDUCE (See Table IV, Appendix) is based on Theorems 6 and 7. This subroutine receives any form and returns an equivalent reduced form with the matrix of transformation.

Statements 3-11: Initialize variables to be used: [A,B,C] is the form being reduced; M is the matrix of the transformation taking [A,B,C] to an equivalent reduced form, which will be one in a chain of reduced forms.

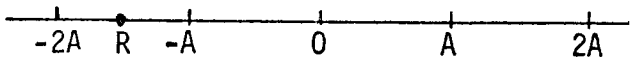
Statement 11: DD is the largest integer $\leq \sqrt{D}$ where D is the discriminant of [A,B,C].

Statement 12: If $|A| \leq DD$, skip to statement 39.

Statements 13, 14: Obtain $R = -B - 2AH$ where $|R| < 2|A|$.

Statement 15: If $|R| \leq |A|$, go to statement 26, otherwise,

Statements 16-25:



A can be positive or negative for an indefinite

form; add or subtract 2A to R and adjust H in any case to obtain $R = -B - 2AH$ with $|R| \leq |A|$.

Statement 26: Statements 15-25 are used by another portion of the subroutine; if CHECK = 1, return to that portion.

Statement 27: For the current value of H, $R = -B - 2AH$; by reversing the sign of H, obtain $R = -B + 2AH$ as required in the transformation $\begin{pmatrix} H & 1 \\ -1 & 0 \end{pmatrix}$.

Statements 28-37: Apply this transformation to [A,B,C] and update transformation matrix.

Statement 38: Go to statement 12 and repeat the process if necessary; after a finite number of loops, $|A| \leq DD$ will hold: skip to

Statement 39: If $|B| \leq |A|$, skip to statement 49; otherwise,

Statement 40: Set CHECK = 1

Statements 41, 42: Obtain $R = B - 2AH$ with $R < |2A|$.

Statement 43: Go to statements 15-25 and then return to

Statement 44: Reverse the sign of H to obtain $R = B + 2AH$ with $|R| \leq |A|$.

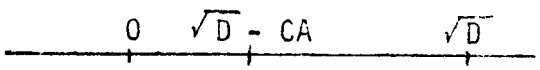
Statements 45-48: Apply transformation $\begin{pmatrix} 1 & H \\ 0 & 1 \end{pmatrix}$ to [A,B,C] to obtain a new [A,B,C] with $|B| \leq |A| \leq DD$; update matrix M.

Statement 49: If $|2C| \leq \sqrt{D}$, go to statement 60. Otherwise,

Statements 50-59: Apply $\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$ and update matrix M.

Statement 60: $CA = 2|C|$.

Statements 61, 62: $BP = B - CA \cdot K$, $|BP| < CA \leq \sqrt{D}$. We have



Statement 63: The desired condition is $\sqrt{D} - CA < BP < \sqrt{D}$ which translates to $DD - CA + 1 \leq BP \leq DD$ (See statement 11).

Statements 64-66: Increase BP and adjust K until $BP = -B - CA \cdot K$ and $BP \geq DD - CA + 1$ both hold.

Statement 67: If $C<0$, $CA = -2C$ and $BP = B - 2C(-K)$.
If $C>0$, $BP = B - 2C(K)$.

Statements 68-71: Apply transformation $\begin{pmatrix} 1 & 0 \\ -K & 1 \end{pmatrix}$ to get reduced form and update matrix M.

Statement 72: Return control to main program.

Sample results from this program are listed in Table XX, Appendix.

CHAPTER II

CLASSIFICATION OF BINARY QUADRATIC FORMS

I. CLASSIFICATION OF INDEFINITE FORMS INTO PERIODS

By Theorem 9 and discussion following, all integral reduced forms of given discriminant $d>0$ may be determined and arranged into periods. The following program (See Table V, Appendix) displays all reduced forms of discriminant $d>0$, arranged by periods. It is based on Theorems 8 and 9.

Statements 6, 7: Initialize D = 1 and L1 = 1

Statements 10, 11: I keeps count of the index; K is the number of reduced forms of given discriminant D.

Statement 12: R is the greatest integer $\leq \sqrt{D}$.

Statement 13: If D is a perfect square, go to next value of D.

Statements 14, 15: If D is even, B1 = 2; if D is odd, B1 = 1.

Statements 16, 17: Any factors X of AC to be computed for this value of B1 must satisfy
$\frac{1}{2}(R - B1) + 1 \leq X \leq \frac{1}{2}(R + B1)$.

Statement 18: Run through all possible values Y for b (in [a,b,c]) from B1 to R, incrementing by 2.

Statement 19: For each value Y, compute $AC = (D - Y^2)/4$. AC is positive.

Statement 20: Check for AC = 0. (This should never happen.)

Statement 21: Run through positive divisors X of AC.

Statement 22: If X doesn't divide AC, go to next potential divisor.

Statement 23: If $X < \frac{1}{2}(R - B1) + 1$ or $X > \frac{1}{2}(R + B1)$, go to next divisor.

Statement 24: $AC = X \cdot Z$.

Statement 25: Same as statement 23 for Z.

Statements 26-34: Store pair of forms [X,Y, -Z] and [-X,Y,Z].

Statements 35, 36: Update counters.

Statements 38, 39: Compute new lower and upper bounds for factors X of AC to be used with next Y value.

Statement 40: Go back to next Y value.

Statement 41: Check to see if there were no reduced forms for a given discriminant. (This never happened.)

At this stage, all K reduced forms of discriminant D are stored as

$$[A(1),B(1),C(1)]$$

$$\vdots$$

$$[A(K),B(K),C(K)] .$$

What follows is a rearrangement of these forms into periods.

Statements 42-44: KK gives the index of the form at beginning of a period.

JJ gives the number of the period.

I counts the index of the form being examined.

Statement 45: $T(I)$ = largest integer $\leq 1/|f|$, where

f = first root of $[A(I),B(I),C(I)]$.

Statements 46-48: Compute right neighboring form:

$$[A(I),B(I),C(I)]$$

$$\Big\downarrow \begin{pmatrix} 0 & 1 \\ -1 & T(I) \end{pmatrix}$$

$$[AR,BR,CR]$$

Statements 49, 50: Test if [AR,BR,CR] equals the first form of the period; if so, go to statement 72.

Statements 51-58: Scan remaining forms for [AR,BR,CR]; when found, go to statement 62.

Statements 59-61: This code is used only if right neighboring form is not found; it was never used.

Statements 62-70: Exchange I^{th} and J^{th} forms.

Statement 71: Go compute next T(I).

Statement 72: The end of period JJ has been reached; CUT(JJ) stores the index of the last form in that period.

Statement 73: If all K forms have been classified, go to statement 78.

Statements 74-77: KK sets starting point of new period; update JJ; go to statement 45.

Statements 78-89: These concern printing out the rearranged forms, period by period.

Statements 90-93: Get next higher value of discriminant D and go to statement 8.

Sample output from the periods program is given in Table XXI, Appendix.

II. CLASSIFICATION INTO GENERA

Theorem 13. Every integral primitive form $[a,b,c]$ represents properly an integer prime to any assigned integer n. (2, p. 82).

Definition 14. Legendre - Jaccbi symbol. If p is an odd prime and m is prime to p, then $(m|p) = \pm 1$ according as m is a quadratic residue or quadratic non-residue of p. If Q is a positive odd integer and m is any integer prime to Q, then $(m|Q) = (m|p_1) \cdot \cdots \cdot (m|p_r)$ where $Q = p_1 \cdot \cdots \cdot p_r$ (p_i = prime, $i = 1, \cdots, r$).

Theorem 14. Let P and Q be positive odd integers; let m and n be integers prime to PQ. Then

(1) $(n|P)\cdot(n|Q) = (n|PQ)$.

(2) $(m|P)\cdot(n|P) = (mn|P)$.

(3) If $n \equiv m \pmod{P}$, then $(n|P) = (m|P)$.

(4) $(-1|P) = (-1)^{(P-1)/2}$; $(2|P) = (-1)^{(P^2-1)/8}$.

(5) If P and Q are relatively prime,

$$(P|Q)\cdot(Q|P) = (-1)^{\frac{1}{2}(P-1)\cdot\frac{1}{2}(Q-1)} .$$

(2, pp. 36, 37).

Definition 15. Given a form $q = [a,2b,c]$ of even discriminant d, then $D = b^2 - ac$ is the determinant of q.

Theorem 15. Given a primitive form $q = [a,b,c]$ of discriminant d, where $p_1, p_2, \cdots$ are the distinct odd prime factors of d, then $(n|p_i)$ has the same value for all integers n prime to 2d which are represented by q. If d is even, the same is true of

$$\delta = (-1)^{(n-1)/2} \quad \text{if } D \equiv 0 \text{ or } 3 \pmod 4$$

$$\varepsilon = (-1)^{(n^2-1)/8} \quad \text{if } D \equiv 0 \text{ or } 2 \pmod 8$$

$$\delta\varepsilon \quad \text{if } D \equiv 0 \text{ or } 6 \pmod 8 .$$

(2, pp. 82, 86, 87).

Definition 16. Given primitive form q = [a,b,c], the invariants $(n|p_i)$ and such of $\delta,\varepsilon,\delta\varepsilon$ as occur for given D, are the characters of q.

Definition 17. All primitive forms of given discriminant d, each of whose characters has an assigned value, form a genus.

Since equivalent forms represent the same numbers, they have the same characters. Thus, each genus is composed of one or more equivalence classes of forms. The notion of genus applies to both definite and indefinite forms.

The following genera program computes and displays all characters associated with a given primitive form q. The components of the general program are:

MAIN PROGRAM	(See Table VI, Appendix)
SUBROUTINE DPRFAC	(See Table VII, Appendix)
SUBROUTINE THM13	(See Table VIII, Appendix)
SUBROUTINE CRTHM	(See Table IX, Appendix)
SUBROUTINE LEGNDR	(See Table X, Appendix) .

The MAIN PROGRAM is based on Theorem 15:

Statements 3-6: Read special printing characters and print heading.

Statement 7: Read value D of discriminant d.

Statement 10: Store value of D as "DST".

Statement 11: If D is odd; go to statement 14.

Statements 12, 13: Z = 1 means discriminant is even; go to statement 16.

Statements 14, 15: Z = 0 means discriminant is odd; double the discriminant (value stored as D).

Statement 16: Call subroutine DPRFAC: send in value of D which is even (d or 2d as the case may be); PR(1), ... , PR(K) return as the distinct prime factors of D.

Statement 17: If discriminant d is even, $D = \frac{d}{4}$; D now stores the value of the determinant of [a,b,c].

Statement 18: If K = 1, the discriminant is a power of 2; only special characters δ, ε, or $\delta\varepsilon$ will be computed; go to statement 64 for heading.

Statements 19-25: involve printing out headings in various cases,

Statement 26: If discriminant is odd, skip special character headings to statement 29.

Statements 27, 28: Special character headings for even discriminant, (at this point, all headings for a given discriminant are printed.)

Statement 29: Read in primitive form [A,B,C].

Statement 30: A = 9999 says go to next discriminant.

Statment 31: Call THM13 subroutine; a number N, represented by [A,B,C] and prime to 2d, returns for use.

Statement 32: For case where d is a power of 2, skip to statement 37.

Statements 33-35: For each odd prime factor of d, PR(2), ... , PR(K), call subroutine LEGNDR to compute (N|PR(I)) = CHAR(I).

Statement 36: If d is odd, skip over special characters.

Statements 37, 38: Initialize all special characters to 0; any special character printed out "0" is to be ignored as a character.

Statements 39-56: Compute special characters as necessary, depending on the value of determinant D. (DEL(1) = δ, DEL(2) = ε, DEL(3) = $\delta\varepsilon$.)

Statement 57: If d is a power of 2, skip to statement 67.

Statements 58, 59: Write all characters of form (N|PR(I)).

Statement 60: If d is odd, go to statement 29.

Statements 64-66: Heading for case d is a power of 2; go to statement 27.

Statements 67-69: Write characters for d = power of 2; go to statement 29.

Subroutine DPRFAC (See Table VII, Appendix) receives an integer N and returns the number K of distinct prime factors of N, those factors PR(1), ··· , PR(K) in ascending order, and two numbers, PT(1) and PT(2), which are prime to N.

Statement 3: Store incoming value N so it will not be changed.

Statements 4-18: Determine distinct prime factors and their number.

Statements 19-29, 32, 33: Determine two numbers prime to N.

Statements 30, 31: Restore saved value of N and return.

Subroutine THM13 (See Table VIII, Appendix) implements Theorem 13. The subroutine receives a primitive form [A,B,C], the number K of distinct prime factors of N, those distinct prime factors PR(1), ··· , PR(K), and two numbers, PT(1) and PT(2), prime to N. The subroutine returns a number Q, properly represented by [A,B,C], which is prime

to N.

Statements 3-14: Run through distinct prime factors of N.

Statement 4: If PR(I) doesn't divide A, go to statement 9.

Statement 5: If PR(I) doesn't divide C, go to statement 12.

Statements 6-8: Both A and C are divisible by PR(I): take X(I) and Y(I) both prime to PR(I); go to next prime factor.

Statements 9-11: Take X(I) prime to PR(I), Y(I) divisible by PR(I); go to next prime factor.

Statements 12-14: Take X(I) divisible by PR(I), Y(I) prime to PR(I); go to next prime factor.

Statements 15, 16: Call Chinese Remainder Theorem to solve

$$XX \equiv X(1) \pmod{PR(1)} \qquad YY \equiv Y(1) \pmod{PR(1)}$$

$$\vdots \qquad \vdots \qquad \text{and} \qquad \vdots \qquad \vdots$$

$$XX \equiv X(K) \pmod{PR(K)} \qquad YY \equiv Y(K) \pmod{PR(K)}.$$

Statements 17, 18: Compute $Q = AX^2 + BXY + CY^2$ and return.

Subroutine CRTHM (See Table IX, Appendix) receives the number K of congruences to be solved, the moduli MOD(1), ··· , MOD(K) of the congruences, and the constants A(1), ··· , A(K) in the congruences. The subroutine returns solution XX to the system

$$XX \equiv A(1) \pmod{MOD(1)}$$

$$\vdots \qquad \vdots$$

$$XX \equiv A(K) \pmod{MOD(K)}$$

of linear congruences.

Statements 3-5: MM = product of the moduli.

Statements 6, 7: MB(I), I = 1, ··· , K, are products of all moduli except the I^{th}.

Statements 8-36: Produce solutions MU(I) to the congruences $MB(I) \cdot MU(I) \equiv 1 \pmod{MOD(I)}$, I = 1, ··· , K.

Statements 37-39: $XX = \sum_{I=1}^{K} MB(I) \cdot MU(I) \cdot A(I)$.

Statements 40-44: Make XX have smallest absolute value possible before returning.

Subroutine LEGNDR (See Table X, Appendix) receives any positive odd prime P and any integer N prime to P. The subroutine returns (N|P) = CHAR. It is based directly on Theorem 14.

Output from the genera program is given in Table XXII, Appendix, for definite forms, and Table XXIII, Appendix, for indefinite forms.

CHAPTER III

COMPOSITION OF BINARY QUADRATIC FORMS

I. COMPOSITION DEFINED ON EQUIVALENCE CLASSES

Theorem 16. If $b^2 \equiv d \pmod{4a}$, $\beta^2 \equiv d \pmod{4\alpha}$, and if $a, \alpha, \frac{1}{2}(b + \beta)$ have no common factor > 1, then there is a solution B of

$$B \equiv b \pmod{2a}$$

$$B \equiv \beta \pmod{2\alpha}$$

$$B^2 \equiv d \pmod{4a\alpha}$$

and B is uniquely determined modulo $2a\alpha$. (2, p. 134).

Theorem 17. If $f = ax^2 + Bxy + \alpha Cy^2$ and $\phi = \alpha\xi^2 + B\xi\eta + aC\eta^2$, then $f\phi = F = a\alpha X^2 + BXY + CY^2$ where $X = x\xi - Cy\eta$, $Y = ax\eta + \alpha y\xi + By\eta$. (2, p. 135).

Definition 19. Two forms $r = [a,b,c]$ and $\rho = [\alpha,\beta,\gamma]$ are *united* if they have the same discriminant d and if a, α, and $\frac{1}{2}(b + \beta)$ have no common factor > 1.

The following paragraph explains how composition of two united forms is performed. If r and ρ are united forms, then by Theorem 16 there is a number B such that $B^2 - d = 4a\alpha C$, $B = b + 2ak$, $B = \beta + 2\alpha\ell$ where C,k,ℓ are integers. Note that

$$\alpha C = (B^2 - d)/4a = \frac{b^2 + 4abk + 4a^2k^2 - (b^2 - 4ac)}{4a}$$

$$\alpha C = c + bk + ak^2 .$$

Similarly, $aC = \gamma + \beta\ell + \alpha\ell^2$.

We have:

$$[a,b,c] \xrightarrow{\begin{pmatrix}1 & k\\ 0 & 1\end{pmatrix}} [a,B,\alpha C] \times [\alpha,B,aC] \xleftarrow{\begin{pmatrix}1 & \ell\\ 0 & 1\end{pmatrix}} [\alpha,\beta,\gamma]$$

$$r \qquad\qquad f \qquad\qquad \phi \qquad\qquad \rho$$

$$\downarrow$$

$$F = [a\alpha,B,C] = f\phi \ .$$

The result of composing r and ρ is F.

The solution B given by Theorem 16 is unique modulo $2a\alpha$. If B is replaced by another solution $B + 2a\alpha t$, then one obtains $(B + 2a\alpha t)^2 - d = 4a\alpha C'$, $B + 2a\alpha t = b + 2ak'$, $B + 2a\alpha t = \beta + 2\alpha\ell'$ where C',k',ℓ' are integers. As before,

$$\alpha C' = c + bk' + a(k')^2$$

$$aC' = \gamma + \beta\ell' + \alpha(\ell')^2 \ .$$

We have:

$$[a,b,c] \xrightarrow{\begin{pmatrix}1 & k'\\ 0 & 1\end{pmatrix}} [a,B + 2a\alpha t,\alpha C'] \times [\alpha,B + 2a\alpha t,aC'] \xleftarrow{\begin{pmatrix}1 & \ell'\\ 0 & 1\end{pmatrix}} [\alpha,\beta,\gamma]$$

$$\downarrow$$

$$[a\alpha,B + 2a\alpha t,C'] = F'.$$

But $[a\alpha,B,C] \xrightarrow{\begin{pmatrix}1 & t\\ 0 & 1\end{pmatrix}} [a\alpha,B + 2a\alpha t,C']$ since $(B + 2a\alpha t)^2 - d = 4a\alpha C'$ and $C + Bt + a\alpha t^2 = C'$.

Hence, the equivalence class of the form resulting from composition is independent of solution B given by Theorem 16.

Theorem 18. For all choices of united forms from two equivalence classes K and L, the forms derived from them by composition belong to a unique equivalence class, denoted KL. (2, p. 136).

Thus, composition may be considered as an operation upon equivalence classes of forms.

Lemma 19. Equivalent forms have the same divisor.

Definition 19. The divisor of an equivalence class is the common divisor of its forms.

Proof of Lemma 19. Let equivalent forms $f = [a,b,c]$ and $\phi = [d,e,f]$ have divisors g and h. Since f represents a,c, and $a + b + c$, ϕ represents a,c, and $a + b + c$. Hence, $h|a$, $h|c$, $h|(a + b + c)$; thus, $h|b$ and $h \leq g$. Similarly, $g \leq h$. ##

Theorem 20. Two equivalence classes K and L admit composition if and only if they have the same discriminant and their divisors are relatively prime. Then the divisor of KL is the product of the divisors of K and L. (2, p. 139).

Thus, composition is a binary operation on the set of all primitive equivalence classes of given discriminant d.

Theorem 21. If $d<0$, the primitive equivalence classes of discriminant d form an abelian group under composition.

Proof. Composition is associative. (2, p. 140). One reduced form of discriminant d is $[1,0, -d/4]$ or $[1,1,(1 - d)/4]$, according as $d \equiv 0$ or $1 \pmod 4$.

$$[1,0,-d/4] \xrightarrow{\begin{pmatrix} 1 & b/2 \\ 0 & 1 \end{pmatrix}} [1,b,ac]$$

$$[1,1,(1-d)/4] \xrightarrow{\begin{pmatrix} 1 & (b-1)/2 \\ 0 & 1 \end{pmatrix}} [1,b,ac] .$$

In either case, if $[a,b,c]$ is any form of discriminant d, $[1,b,ac] \times [a,b,c] = [a,b,c]$. Thus, the class containing one of the above forms is the group identity element. Given primitive form $[a,b,c]$, then

$$[a,b,c] \times [c,b,a] = [ac,b,1] \xrightarrow{\begin{pmatrix} 0 & 1 \\ -1 & 1 \end{pmatrix}} [1,b,ac] .$$

Hence, the class containing $[c,b,a]$ is inverse to the class containing $[a,b,c]$. Since $[a,b,\alpha c] \times [\alpha,b,ac] = [\alpha,b,ac] \times [a,b,\alpha c] = [a\alpha,b,c]$, composition is commutative. ##

II. GROUP STRUCTURE USING PRIME POWER ORDER GENERATORS

Theorem 22. Any finite abelian group can be expressed as a direct sum of cyclic subgroups of prime power order. The number of cyclic subgroups in the direct sum and their orders are completely determined by the group. (4, p. 302).

The following program displays the group of primitive equivalence classes under composition according the Theorem 22. The group structure is unknown ahead of time. The elements of the group are arranged in no particular order except that the first element is known to be the identity; the order of the group is also known. The components of this group structure program are:

MAIN PROGRAM	(See Table XI, Appendix)
SUBROUTINE DPRFAC	(See Table XII, Appendix)
SUBROUTINE SCAN	(See Table XIII, Appendix)
SUBROUTINE SWITCH	(See Table XIV, Appendix)
SUBROUTINE CMPOSE	(See Table XV, Appendix)
SUBROUTINE GCD	(See Table XVI, Appendix)
SUBROUTINE REDUCE	(See Table II, Appendix) .

The MAIN PROGRAM is based on Theorem 23:

Statement 6: Read printing symbols KK(1) = A, KK(2) = B, ... , KK(10) = J.

Statement 10: Read discriminant D and number N of primitive forms of this discriminant. (N = order of the group.)

Statement 13: The value of D comes in positive; change to negative for calculations.

Statements 14, 15: Read and store N forms as follows:

[A(1), B(1), C(1)] = identity element.

[A(2), B(2), C(2)]

⋮

[A(I), B(I), C(I)]

⋮

[A(N), B(N), C(N)] .

Statements 16, 17: Write down identity element.

Statement 18: If order of group is 1, go to statement 103.

Statements 19, 20: Initialize TRY(I) = 0 for I = 1, ... , N. TRY(I) = 0 means that the I^{th} form has not been tried as a

possible cyclic subgroup generator.

Statement 21: Initialize BA = 1; BA gives the index of each cyclic subgroup generator.

Statement 22: Initialize I = 2; I = index of the form currently being examined.

Statement 23: TRY(I) = 1 indicates that the form currently in the I^{th} position is to be examined as a possible generator.

Statement 24: Initialize T = 1; T counts the number of elements in a cyclic subgroup, not counting the identity.

Statements 25-34: Compute and store in a temporary table the cyclic subgroup (except identity) generated by the I^{th} form:

$$\begin{matrix} [AT(1) & BT(1) & CT(1)] \\ [AT(2) & BT(2) & CT(2)] \\ \vdots & \vdots & \vdots \\ [AT(T) & BT(T) & CT(T)] \end{matrix} .$$

When this table is completed, go to

Statement 35: ORD(BA) stores the order of the BA^{th} cyclic subgroup.

Statement 36: Call subroutine DPRFAC to see if ORD(BA) is a prime power.

Statement 37: If the number K of prime factors of ORD(BA) is not 1, go to statement 93.

Statements 38, 39: The I^{th} element generates a cyclic subgroup of prime power order; LOC(BA) stores the location of the BA^{th} generator; BPR(BA) is the base prime for this subgroup. A generator table is kept for

reference.

LOC(1)	BPR(1)	ORD(1)
⋮	⋮	⋮
LOC(BA)	BPR(BA)	ORD(BA).

Statements 40-56: Scan generator table to see if the new generator should be retained or thrown out.

Statement 40: If BA = 1, there is no generator table to check; go to statement 63.

Statements 41, 42: Otherwise, scan preceding generators 1 through BA - 1.

Statement 43: If the current base prime doesn't equal that of the J^{th} generator, go to the next one.

Statement 44: At this stage, the current base prime equals the J^{th} base prime. Denote by (b) the new BA^{th} cyclic subgroup; let (a) denote the old J^{th} cyclic subgroup.

Statements 44, 45: LO and HI are the first and last elements of (a), ignoring the identity element.

Statements 46, 47: If order of (b) > order of (a), GT = 1.
If order of (b) $\leq$ order of (a), GT = 0.

Statements 48-55: Scan all forms of (a) and (b) to see if (a) $\cap$ (b) = {1}.

Statement 53: If GT = 1 and the generator of (a) $\in$ (b), then (a) $\subsetneq$ (b): go to statement 58.

Statement 54: (a) $\cap$ (b) $\neq$ {1}: go to statement 93.

Statement 57: (a) $\cap$ (b) = {1}; go to statement 63.

Statements 58-62: In this case (a) $\subsetneq$ (b); throw out (a) and replace by (b), putting (b) in as J^{th} cyclic subgroup. Set BA and I back to J; set TRY(I) = 0 from J^{th} to N^{th} forms so wiped out generators may be tried again.

Statements 63-70: Bring the new BA^{th} cyclic subgroup out of the temporary table and put it in the main group table, labeling each element.

Statement 71: I comes out of preceding section one too large: reduce it by 1.

Statement 72: If end of the group is reached, go to statement 98.

Statement 73: If the first cyclic subgroup has been computed, skip to statement 91.

Statements 74-89: Cross multiply most recently computed cyclic subgroup with preceding group elements, and assign labels.

Statement 90: If end of group is reached, go to statement 98.

Statements 91, 92: Prepare to obtain next generator: update I and BA.

Statements 93-97: Scan remaining I + 1 through N^{th} forms for one which has not been tried; when such a form is found, switch it into the I^{th} position and go to statement 23.

Statements 98-104: Print out the whole group.

Statement 105: Go read next discriminant.

The following figure schematically displays the components of this program:

GENERATOR TABLE

LOC(1) = 2 BPR(1) = 3 ORD(1) = 3

LOC(2) = 4 BPR(2) = 2 ORD(2) = 2

LOC(3) = 7 BPR(3) = 2 ORD(3) = 4

The order of the group is 24; we have $A^3 = B^2 = C^4 = 1$.

	MAIN GROUP TABLE	Label
	[A(1), B(1), C(1)]	1
LOC(1):	[A(2), B(2), C(2)]	A
	[A(3), B(3), C(3)]	AA
LOC(2):	[A(4), B(4), C(4)]	B
	[A(5), B(5), C(5)]	AB
	[A(6), B(6), C(6)]	AAB
LOC(3):	[A(7), B(7), C(7)]	C
	[A(8), B(8), C(8)]	CC
	[A(9), B(9), C(9)]	CCC
	[A(10), B(10), C(10)]	AC
	[A(11), B(11), C(11)]	AAC
	⋮	⋮
	[A(24), B(24), C(24)]	AABCCC

Subroutine DPRFAC (See Table XII, Appendix) receives a number N and returns the number K of distinct prime factors of N and those factors PR(1), ··· , PR(K).

Subroutine SCAN (See Table XIII, Appendix) receives index I and form [AC,BC,CC]. It scans forms I, ··· , N to see which one is [AC,BC,CC]; if the K^{th} form is [AC,BC,CC], switch the I^{th} and K^{th} forms, putting the form equal to [AC,BC,CC] into the I^{th} position.

Subroutine SWITCH (See Table XIV, Appendix) switches the I^{th} and K^{th} forms and their TRY(I) values.

Whereas the preceding program segments may be used (with modification) for any finite abelian group, the following three subroutines pertain to the group composition in this example.

Subroutine CMPOSE (See Table XV, Appendix) receives two forms [A1,B1,C1] and [A2,B2,C2] and their discriminant D; it returns the composition [AC,BC,CC] of these two forms.

Statements 4-37: Find a form [M,N,L] equivalent to [A2,B2,C2] such that [A1,B1,C1] and [M,N,L] are united.

Statements 4-18: Find a number M represented properly by [A2,B2,C2] such that M is prime to A1; in order to keep M small, only the following combinations of X and Y are tried:

$(X,Y) = (1, \pm 1)$ $(X,Y) = (1, \pm 2)$ $(X,Y) = (1, \pm 3)$.

$(X,Y) = (2, \pm 1)$ $(X,Y) = (3, \pm 2)$ $(X,Y) = (2, \pm 3)$.

$(X,Y) = (3, \pm 1)$

Considerable evidence indicates that these 14 possibilities give the desired M; when such an M is found, go to statement 22:

Statements 22-37: Search for roots of the congruence $n^2 \equiv d \pmod{4m}$. For each root N, compute [M,N,L] and reduce it to see if it is equivalent to [A2,B2,C2]; one such [M,N,L] will be (Theorem 5).

Statement 38: At this point, united forms [A1,B1,C1] = [a,b,c] and [M,N,L] = $[\alpha,\beta,\gamma]$ are ready to be composed. Following Theorem 16, obtain a solution to the

system

$$\alpha B \equiv \alpha b \pmod{2a\alpha}$$

$$aB \equiv a\beta \pmod{2a\alpha}$$

$$\tfrac{1}{2}(b + \beta)\, B \equiv \tfrac{1}{2}(d + b\beta) \pmod{2a\alpha}\ .$$

A solution B is given by $B = \alpha b h_1 + a\beta h_2 + \frac{1}{2}(d + b\beta)h_3$ where h_1, h_2, h_3 are such that

$$\alpha h_1 + \alpha h_2 + \frac{1}{2}(b + \beta) \equiv 1 \pmod{2a\alpha}.$$

Since α and a (M and A1 in program) are relatively prime, take $h_3 = 0$ and solve $\alpha h_1 + a h_2 = 1$.

Statement 38: Call subroutine GCD to solve M•H1 + A1•H2 = G ($\alpha h_1 + a h_2 = 1$).

Statements 39-43: Compose [a,B,αC] and [α,B,aC] to get [aα,B,C].

Statement 39: A = M•A1 (= aα).

Statement 40: $B = \alpha b h_1 + a\beta h_2$.

Statements 41, 42: Make solution B as small as possible modulo 2aα.

Statement 43: Compute C; [A,B,C] is the composition of [A1,B1,C1] and [A2,B2,C2].

Statements 44, 45: Reduce [A,B,C] to [AC,BC,CC] and return.

Subroutine GCD (See Table XVI, Appendix) receives two integers MS and PS. It solves the equation MS•C + PS•E = M where M is the greatest common divisor of MS and PS.

Sample output from this group structure program is given in Table XXIV, Appendix.

III. GROUP STRUCTURE USING MINIMAL NUMBER OF GENERATORS

For given discriminant $d<0$, the set of primitive reduced forms of discriminant d (each representing its equivalence class) forms an abelian group under composition. The following program displays the group using the minimal number of generators. The components of this program are:

MAIN PROGRAM	(See Table XVII, Appendix)
SUBROUTINE BASE	(See Table XVIII, Appendix)
SUBROUTINE SCAN	(See Table XIII, Appendix)
SUBROUTINE SWITCH	(See Table XIV, Appendix)
SUBROUTINE CMPOSE	(See Table XV, Appendix)
SUBROUTINE GCD	(See Table XVI, Appendix)
SUBROUTINE REDUCE	(See Table II, Appendix) .

The MAIN PROGRAM receives the same data as the preceding MAIN PROGRAM (Table XI, Appendix).

Statements 5-19: These statements are identical with statements 6-20 of the preceding MAIN PROGRAM (See Table XI, Appendix).

Statement 20: BA gives the printing character index; this index increases by 1 each time a new generator is chosen.

Statement 21: L gives the index of the current base element (a cyclic subgroup generator is called a base element hereafter).

Statement 22: I gives the index of the form currently being examined.

Statement 23: TRY(I) = 1 means that the form currently in the I^{th} position will be examined as a suitable base element.

Statement 24: If order of group is 2, go to statement 26.

Statement 25: Call subroutine BASE; this subroutine checks to see if the form currently in location L is suitable as a base element; if not, it puts a suitable base element in the L^{th} position.

Statements 26, 27: CO gives the number of printing characters associated with the current position's label. LAB(I,1) is the BA^{th} printing character.

Statement 28: NO(I) stores the number of printing characters for the label of the I^{th} form.

Statement 29: By composition, compute the next power of the base element; this new power is [AC,BC,CC].

Statement 30: If this power is not the identity element, go to statement 32.

Statement 31: This power is the identity; the end of a cyclic subgroup has been reached; go to statement 39.

Statement 32: Increase index by 1.

Statement 33: Increase number of printing characters by 1.

Statement 34: Call SCAN: this subroutine scans all remaining forms for [AC,BC,CC] and puts the correct one in the I^{th} position.

Statements 35, 36: Assign correct label to the I^{th} form.

Statement 37: Store the number of printing characters for the label of this form.

Statement 38: Go to statement 29.

Statement 39: A cyclic subgroup has just been completed; if the group has been fully scanned, go to statement 61.

Statement 40: If L = 2, there is no cross multiplication between different cyclic subgroups; skip to statement 57.

Statements 41-55: Cross multiply most recently computed cyclic subgroup with preceding group elements, and assign labels.

Statement 56: If entire group has been classified, go to statement 61.

Statements 57-60: Prepare to compute next cyclic subgroup: increase index; set base index up; set printing character index up; go to statement 23.

Statements 61-67: Pring out everything for the group.

Statement 68: Go to statement 9 for next group.

Subroutine BASE (See Table XVIII, Appendix) receives base index L and makes sure a suitable base element is in the L^{th} position before returning.

Statements 4-23: For a given value L, search forms L + 1 to N to see if one of them powers to give L^{th} form. At the end of statement 23, a maximal cyclic subgroup generator is in position L.

Statement 6: If TRY(I) = 1, this form has already been tried as a base element and rejected; go to next form.

Statements 7-9: Initialize [A2,B2,C2] to obtain powers of form I.

Statement 10: Compose to get next higher power of form I.

Statement 11: If this power ≠ identity, go to statement 13.

Statement 12: If this power = identity, go to $(I + 1)^{st}$ form. (Will not replace form L by form I.)

Statements 13-15: If this power of form I ≠ form L, go to statement 19.

Statements 16-18: If this power of form I = form L, switch forms I and L; set TRY(L) = 1; go to scan forms I + 1, ⋯ , N as before.

Statements 19-22: Prepare to calculate next higher power of form I; go to statement 10.

Statement 24: If L = 2, need not do second half of subroutine.

Statements 25-46: These statements check to see if the base element (form L) is suitable; if form L powers into proceeding forms 2 through L-1, a new base element is put into location L and control reverts to statement 5 of the subroutine.

Statements 25-27: Initialize [A2,B2,C2] in order to take powers of form L.

Statement 28: Compose to get next higher power of form L.

Statements 29-35: Run through forms 1 to L-1 to see if this power of form L equals one of them.

Statements 31-35: If the power is not equal to J^{th} form, go to $(J + 1)^{st}$ form.

Statement 34: If the power of form L = form J, go to statement 40.

Statements 36-39: That particular power of form L is not equal to any of the forms 1 through L-1; go to statement 28 to

compute next higher power of form L.

Statement 40: If the particular power of form L = identity (J = 1), that base element doesn't power into previous forms; then return.

Statement 41: At this stage, form L has powered into forms 2 through L-1 and must be replaced; check remaining forms L+1 through N for a replacement.

Statement 42: If TRY(I) = 1, this form has been tried as a base element before; go to next form.

Statements 43-45: Otherwise, switch forms I and L; set TRY(L) = 1 and go to statement 5.

Statement 46: Check next of remaining forms.

Statement 47-49: If no base is found, write that fact and return.

Sample output of this group structure program is given in Table XXIV, Appendix.

CHAPTER IV

OBSERVATIONS

The following observations are based on the data obtained from the programs.

(1) The continued fraction process of Theorem 12 (2, p. 113) does in fact yield the minimal positive solution (T,U) of the equation $t^2 - du^2 = 4$.

(2) Based on discriminants -3 to -500, we have: there is one genus for discriminant $d<0$ if and only if -d is prime.

(3) Based on discriminants 1 to 600, we have: there is one period of reduced forms of discriminant $d>0$ if and only if d is prime.

(4) Definition 20. A form [a,b,c] is ambiguous if a divides b.

Theorem 23. Every period which contains an ambiguous form contains exactly two ambiguous forms. (2, p. 116).

Based on discriminants 1 to 600, it appears that every period has exactly two ambiguous forms.

BIBLIOGRAPHY

BIBLIOGRAPHY

1. Davenport, Harold. The Higher Arithmetic: An Introduction to the Theory of Numbers. London : Hutchinson House, 1952.

2. Dickson, Leonard E. Introduction to the Theory of Numbers. Chicago : The University of Chicago Press, 1929.

3. Mathews, G. B. Theory of Numbers, Part I. New York : G. E. Stechert & Company, 1892.

4. Mostow, George D., Sampson, Joseph H., Meyer, Jean-Pierre. Fundamental Structures of Algebra. New York : McGraw-Hill Book Company, 1963.

APPENDIXES

TABLE I

REPRESENTATION PROGRAM FOR DEFINITE FORMS

```
 1   IMPLICIT INTEGER (A-Z)
 2   DIMENSION R(8),S(8),T(8),U(8),M1(2,2),M2(2,2),TR(2,2)
 3   DO 6 I=1,8
 4   READ (1,5) R(I),S(I),T(I),U(I)
 5   FORMAT (4I2)
 6   CONTINUE
 7   WRITE (3,8)
 8   FORMAT (1H1,///,2X,'ALL REPRESENTATIONS (X,Y) OF THE
    CNUMBER M BY THE FORM (A,B,C)',////)
 9   READ (1,10) A,B,C
10   FORMAT (3I4)
11   IF (A .EQ. 7777) CALL EXIT
12   WRITE (3,13) A,B,C
13   FORMAT (1HO,//,5X,'(A,B,C)=',2X,'(',I3,',',I3,',',I3,')',/)
14   D = B*B - 4*A*C
15   CALL REDUCE (A,B,C,AR,BR,CR,M1)
16   IF (A .EQ. C .AND. B .EQ. 0) GO TO 21
17   IF (A .EQ. B .AND. B .EQ. C) GO TO 24
18   K1 = 3
19   K2 = 4
20   GO TO 26
21   K1 = 1
22   K2 = 4
23   GO TO 26
24   K1 = 3
25   K2 = 8
26   READ (1,10) M
27   IF (M .EQ. 9999) GO TO 9
```

TABLE I (continued)

```
28   WRITE (3,29) M
29   FORMAT (1HO,9X,'M=',I5)
30   MM = 2*M-1
31   M4 = 4*M
32   N = 0
33   NN = N*N-D
34   IF (NN/M4*M4 .EQ. NN) GO TO 38
35   N = N+1
36   IF (N .GT. MM) GO TO 26
37   GO TO 33
38   L = NN/M4
39   CALL REDUCE (M,N,L,MR,NR,LR,M2)
40   IF (AR .NE. MR) GO TO 35
41   IF (BR .NE. NR) GO TO 35
42   IF (CR .NE. LR) GO TO 35
43   TR(1,1) = M1(1,1)*M2(2,2)-M1(1,2)*M2(2,1)
44   TR(2,1) = M1(2,1)*M2(2,2)-M1(2,2)*M2(2,1)
45   DO 50 I=K1,K2
46   X = R(I)*TR(1,1)+S(I)*TR(2,1)
47   Y = T(I)*TR(1,1)+U(I)*TR(2,1)
48   WRITE (3,49) X,Y
49   FORMAT (1H ,5X,'(X,Y)=',2I5)
50   CONTINUE
51   GO TO 35
52   END
```

Note: Statement 8 should occupy 72 card columns before beginning continuation card.

TABLE II

SUBROUTINE REDUCE FOR DEFINITE FORMS

```
 1   SUBROUTINE REDUCE (A,B,C,AR,BR,CR,M)
 2   INTEGER A,B,C,AR,BR,CR,S,C2,TEMP2,M(2,2),P(2,2),TEMP1
 3   AR = A
 4   BR = B
 5   CR = C
 6   M(1,1) = 1
 7   M(2,1) = 0
 8   M(1,2) = 0
 9   M(2,2) = 1
10   TEMP1 = BR
11   TEMP2 = CR
12   C2 = 2*CR
13   S = (-BR)/C2
14   BR = -BR - C2*S
15   IF (IABS(BR) .LE. CR) GO TO 22
16   IF (BR .LT. 0) GO TO 20
17   BR = BR - C2
18   S = S+1
19   GO TO 22
20   BR = BR+C2
21   S = S-1
22   CR = AR + TEMP1*S + CR*S**2
23   AR = TEMP2
24   P(1,1) = -M(1,2)
25   P(2,1) = -M(2,2)
26   P(1,2) = M(1,1) + S*M(1,2)
27   P(2,2) = M(2,1) + S*M(2,2)
28   DO 30 I = 1,2
```

TABLE II (continued)

```
29    DO 30 J=1,2
30    M(I,J) = P(I,J)
31    IF (CR .LT. AR) GO TO 10
32    IF (BR .NE. -AR) GO TO 36
33    BR = AR
34    M(1,2) = M(1,1) + M(1,2)
35    M(2,2) = M(2,1) + M(2,2)
36    IF (AR .NE. CR) GO TO 45
37    IF (BR .GE. 0) GO TO 45
38    BR = -AR
39    TEMP1 = M(1,1)
40    TEMP2 = M(2,1)
41    M(1,1) = -M(1,2)
42    M(2,1) = -M(2,2)
43    M(1,2) = TEMP1
44    M(2,2) = TEMP2
45    RETURN
46    END
```

TABLE III

REPRESENTATION PROGRAM FOR INDEFINITE FORMS

```
 1   INTEGER A,B,C,D,AR,BR,CR,M1(2,2),M2(2,2),TR(2,2),ACH(50),BCH(50)
 2   INTEGER CCH(50),T(50),X,Y,Z,W,ALP,BET,GAM,DEL,U,TT,TEMP1,TEMP2
 3   INTEGER M3(2,2),F
 4   LOGICAL TEST
 5   WRITE (3,6)
 6   FORMAT (1H1,///,15X,'PROPER REPRESENTATIONS (X,Y) OF THE
    CNUMBER M BY THE FORM (A,B,C)',///)
 7   READ (1,8) A,B,C
 8   FORMAT (3I4)
 9   IF (A .EQ. 7777) CALL EXIT
10   WRITE (3,11) A,B,C
11   FORMAT (1HO,3X,'(A,B,C)=',3I6)
12   D = B*B - 4*A*C
13   CALL REDUCE (A,B,C,ACH(1),BCH(1),CCH(1),M1,D)
14   DO 16 I = 1,2
15   DO 16 J = 1,2
16   M3(I,J) = M1(I,J)
17   K = 1
18   I = 1
19   T(I) = (2.0*FLOAT(ACH(I)))/(SQRT(FLOAT(D))-FLOAT(BCH(I)))
20   ACH(I+1) = CCH(I)
21   BCH(I+1) = -BCH(I)-2*T(I)*CCH(I)
22   CCH(I+1) = ACH(I)+BCH(I)*T(I)+CCH(I)*T(I)*T(I)
23   I = I+1
24   TEST = ACH(I).EQ.ACH(1).AND.BCH(I).EQ.BCH(1).AND.CCH(I).EQ.CCH(1)
25   IF (TEST) GO TO 28
26   K = K-1
27   GO TO 19
```

TABLE III (continued)

```
28   IF (ACH(1) .LT. 0) GO TO 32
29   J = 0
30   I = K-1
31   GO TO 34
32   J = 1
33   I = K
34   IF (K .EQ. 2) GO TO 45
35   Y = 1
36   X = IABS(T(I-1))
37   Z = IABS(T(I))
38   W = Z
39   Z = Z*X+Y
40   I = I-1
41   IF (I .EQ. J+1) GO TO 47
42   X = IABS(T(I-1))
43   Y = W
44   GO TO 38
45   Z = IABS(T(I))
46   W = 1
47   U = Z/ACH(J+1)
48   TT = 2*W+BCH(J+1)*U
49   READ (1,8) M
50   IF (M .EQ. 9999) GO TO 7
51   WRITE (3,52) M
52   FORMAT (1H0,5X,'M=',I6)
53   MM = 2*IABS(M)-1
54   M4 = 4*M
55   N = 0
56   NN = N*N-D
```

TABLE III (continued)

```
57   IF (NN/M4*M4.EQ.NN) GO TO 61
58   N = N+1
59   IF (N .GT. MM) GO TO 49
60   GO TO 56
61   L = NN/M4
62   CALL REDUCE (M,N,L,MR,NR,LR,M2,D)
63   DO 68 I=1,K
64   IF (MR .NE. ACH(I)) GO TO 68
65   IF (NR .NE. BCH(I)) GO TO 68
66   IF (LR .NE. CCH(I)) GO TO 68
67   GO TO 70
68   CONTINUE
69   GO TO 58
70   IF (I .EQ. 1) GO TO 80
71   I = I-1
72   DO 79 J=1,I
73   TEMP1 = -M1(1,2)
74   TEMP2 = -M1(2,2)
75   M1(1,2) = M1(1,1)+T(J)*M1(1,2)
76   M1(2,2) = M1(2,1)+T(J)*M1(2,2)
77   M1(1,1) = TEMP1
78   M1(2,1) = TEMP2
79   CONTINUE
80   TR(1,1) = M1(1,1)*M2(2,2)-M1(1,2)*M2(2,1)
81   TR(2,1) = M1(2,1)*M2(2,2)-M1(2,2)*M2(2,2)
82   DO 84 I=1,2
83   DO 84 J=1,2
84   M1(I,J) = M3(I,J)
85   F = -1
```

TABLE III (continued)

```
 86   DO 90 I=1,2
 87   F = - F
 88   X = F*TR(1,1)
 89   Y = F*TR(2,1)
 90   WRITE (3,102) X,Y
 91   DO 103 I=1,2
 92   U = -U
 93   DO 103 J=1,2
 94   TT = -TT
 95   ALP = (TT-B*U)/2
 96   BET = -C*U
 97   GAM = A*U
 98   DEL = (TT+B*U)/2
 99   X = ALP*TR(1,1)+BET*TR(2,1)
100   Y = GAM*TR(1,1)+DEL*TR(2,1)
101   WRITE (3,102) X,Y
102   FORMAT (1H ,5X,'(X,Y)=',2I12)
103   CONTINUE
104   GO TO 58
105   END
```

Note: Statement 6 should occupy 72 card columns before beginning continuation card.

TABLE IV

SUBROUTINE REDUCE FOR INDEFINITE FORMS

```
 1   SUBROUTINE REDUCE (AR,BR,CR,A,B,C,M,D)
 2   INTEGER A,B,C,D,AR,BR,CR,DD,R,H,M(2,2),CHECK,BP,TEMP1,TEMP2,CA
 3   A = AR
 4   B = BR
 5   C = CR
 6   M(1,1) = 1
 7   M(2,1) = 0
 8   M(1,2) = 0
 9   M(2,2) = 1
10   CHECK = 0
11   DD = SQRT(FLOAT(D))
12   IF (IABS(A) .LE. DD) GO TO 39
13   H = (-B)/(2*A)
14   R = -B-2*A*H
15   IF (IABS(R) .LE. IABS(A)) GO TO 26
16   IF (R .GT. 0) GO TO 24
17   IF (A .GT. 0) GO TO 21
18   R = R-2*A
19   H = H+1
20   GO TO 26
21   R = R+2*A
22   H = H-1
23   GO TO 26
24   IF (A .GT. 0) GO TO 18
25   GO TO 21
26   IF (CHECK .EQ. 1) GO TO 44
27   H = -H
28   TEMP1 = A
```

TABLE IV (continued)

```
29    A = A*H*H-B*H+C
30    B = R
31    C = TEMP1
32    TEMP1 = M(1,1)
33    TEMP2 = M(2,1)
34    M(1,1) = M(1,1)*H-M(1,2)
35    M(2,1) = M(2,1)*H-M(2,2)
36    M(1,2) = TEMP1
37    M(2,2) = TEMP2
38    GO TO 12
39    IF (IABS(B) .LE. IABS(A)) GO TO 49
40    CHECK = 1
41    H = B/(2*A)
42    R = B-2*A*H
43    GO TO 15
44    H = -H
45    C = A*H*H+B*H+C
46    B = R
47    M(1,2) = M(1,1)*H+M(1,2)
48    M(2,2) = M(2,1)*H+M(2,2)
49    IF (IABS(2*C) .LE. DD) GO TO 60
50    TEMP1 = A
51    A = C
52    B = -B
53    C = TEMP1
54    TEMP1 = M(1,1)
55    TEMP2 = M(2,1)
56    M(1,1) = -M(1,2)
57    M(2,1) = -M(2,2)
```

TABLE IV (continued)

```
58    M(1,2) = TEMP1
59    M(2,2) = TEMP2
60    CA = 2*IABS(C)
61    K = B/CA
62    BP = B-CA*K
63    IF (BP .GE. DD-CA+1) GO TO 67
64    BP = BP+CA
65    K = K-1
66    GO TO 63
67    IF (C .LT. 0) K = -K
68    A = A-K*B+C*K*K
69    B = BP
70    M(1,1) = M(1,1)-K*M(1,2)
71    M(2,1) = M(2,1)-K*M(2,2)
72    RETURN
73    END
```

TABLE V

PROGRAM FOR LISTING REDUCED INDEFINITE FORMS BY PERIODS

```
 1   INTEGER X,Y,Z,D,AC,R,UB,B1,AR,BR,CR,TEMP1,TEMP2,TEMP3
 2   INTEGER A(100),B(100),C(100),T(100),CUT(20)
 3   LOGICAL TEST
 4   WRITE (3,5)
 5   FORMAT (1H1,///,30X,'REDUCED INTEGRAL FORMS OF
    CDISCRIMINANT D>0 (D NOT A SQUARE)'/50X,'ARRANGED IN PERIODS'/)
 6   D = 5
 7   L1 = 1
 8   WRITE (3,9) D
 9   FORMAT (1HO,2X,'D=',I4)
10   I = 1
11   K = 0
12   R = SQRT(FLOAT(D))
13   IF (R*R .EQ. D) GO TO 90
14   B1 = 2
15   IF (D/2*2 .NE. D) B1 = 1
16   LB = (R-B1)/2+1
17   UB = (R+B1)/2
18   DO 40 Y=B1,R,2
19   AC = (D-Y*Y)/4
20   IF (AC .EQ. 0) GO TO 40
21   DO 37 X=1,AC
22   IF (AC/X*X .NE. AC) GO TO 37
23   IF (X.LT.LB .OR. X.GT.UB) GO TO 37
24   Z = AC/X
25   IF (Z.GT.UB .OR. Z.LT.LB) GO TO 37
26   MA = -X
27   MC = -Z
```

TABLE V (continued)

```
28    A(I) = X
29    B(I) = Y
30    C(I) = MC
31    I = I+1
32    A(I) = MA
33    B(I) = Y
34    C(I) = Z
35    I = I+1
36    K = K+2
37    CONTINUE
38    LB = LB-1
39    UB = UB+1
40    CONTINUE
41    IF (K .EQ. 0) GO TO 90
42    KK = 1
43    JJ = 1
44    I = 1
45    T(I) = (2.0*FLOAT(A(I)))/(SQRT(FLOAT(D))-FLOAT(B(I)))
46    AR = C(I)
47    BR = -B(I)-2*T(I)*C(I)
48    CR = A(I)+B(I)*T(I)+C(I)*T(I)*T(I)
49    TEST = AR.EQ.A(KK) .AND. BR.EQ.B(KK) .AND. CR.EQ.C(KK)
50    IF (TEST) GO TO 72
51    I = I+1
52    DO 58 J=I,K
53    IF (AR .EQ. A(J)) GO TO 55
54    GO TO 58
55    IF (BR .EQ. B(J)) GO TO 57
56    GO TO 58
```

TABLE V (continued)

```
57    IF (CR .EQ. C(J)) GO TO 62
58    CONTINUE
59    WRITE (3,60)
60    FORMAT (1HO,'RIGHT NEIGHBORING FORM NOT FOUND IN LIST')
61    GO TO 90
62    TEMP1 = A(I)
63    TEMP2 = B(I)
64    TEMP3 = C(I)
65    A(I) = A(J)
66    B(I) = B(J)
67    C(I) = C(J)
68    A(J) = TEMP1
69    B(J) = TEMP2
70    C(J) = TEMP3
71    GO TO 45
72    CUT(JJ) = I
73    IF (I .EQ. K) GO TO 78
74    I = I+1
75    KK = I
76    JJ = JJ+1
77    GO TO 45
78    K1 = 1
79    DO 89 L=1,JJ
80    WRITE (3,81) L
81    FORMAT (1HO,2X,'BEGINNING OF PERIOD',I3)
82    K2 = CUT(L)
      DO 86 I=K1,K2
84    WRITE (3,85)
85    FORMAT (1H ,2X,'(A,B,C)=',3I5,5X,'DELTA=',I3)
```

TABLE V (continued)

```
86    CONTINUE
87    WRITE (3,88) L
88    FORMAT (1H ,8X,'END OF PERIOD', I3)
89    K1 = K2+1
90    L1 = 4-L1
91    D = D+L1
92    IF (D .GT. 500) CALL EXIT
93    GO TO 8
94    END
```

Note: Statement 5 should occupy 72 card columns before beginning continuation card.

TABLE VI

MAIN PROGRAM FOR GENERA PROGRAM

```
 1   INTEGER A,B,C,D,E,DEL(3),Z,CHAR(10),PR(10),DST,PT(2),F(8)
 2   LOGICAL TEST
 3   READ (1,4) F
 4   FORMAT (8A4)
 5   WRITE (3,6)
 6   FORMAT (1H1,//,37X,'CHARACTERS OF PRIMITIVE FORMS FOR
    CGIVEN DISCRIMINANT',/)
 7   READ (1,8) D
 8   FORMAT (3I4)
 9   IF (D .LE. 0) CALL EXIT
10   DST = D
11   IF (D/2*2 .NE. D) GO TO 14
12   Z = 1
13   GO TO 16
14   D = 2*D
15   Z = 0
16   CALL DPRFAC (D,PR,K,PT)
17   IF (Z .EQ. 1) D=D/4
18   IF (K .EQ. 1) GO TO 64
19   WRITE (3,20) DST
20   FORMAT (1HO,9X,'D=',I4,23X,'(N,')
21   IF (K .LE. 2) GO TO 24
22   WRITE (3,23) (F(I),I=3,K)
23   FORMAT (1H+,41X,8(5X,A4))
24   WRITE (3,25) (PR(I),I=2,K)
25   FORMAT (1H+,41X,8(I3,')',5X))
26   IF (Z .EQ. 0) GO TO 29
27   WRITE (3,28)
```

TABLE VI (continued)

```
28   FORMAT (1H+,110X,'DEL  EPS  (DEL)(EPS)')
29   READ (1,8) A,B,C
30   IF (A .EQ. 9999) GO TO 7
31   CALL THM13 (A,B,C,PR,K,PT,N)
32   IF (K .EQ. 1) GO TO 37
33   DO 35 I=2,K
34   CALL LEGNDR (N,PR(I),CHAR(I))
35   CONTINUE
36   IF (Z .EQ. 0) GO TO 57
37   DO 38 I=1,3
38   DEL(I) = 0
39   TEST = (D/4*4 .EQ. D) .OR. ((D-3)/4*4 .EQ. (D-3))
40   IF (TEST) GO TO 46
41   TEST = (D/8*8 .EQ. D) .OR. ((D-2)/8*8 .EQ. (D-2))
42   IF (TEST) GO TO 50
43   TEST = (D/8*8 .EQ. D) .OR. ((D-6/8*8 .EQ. (D-6))
44   IF (TEST) GO TO 54
45   GO TO 57
46   E = (N-1)/2
47   DEL(1) = -1
48   IF (E/2*2 .EQ. E) DEL(1) = 1
49   GO TO 41
50   E = (N*N-1)/8
51   DEL(2) = -1
52   IF (E/2*2 .EQ. E) DEL(2)=1
53   GO TO 43
54   E = (N-1)/2+(N*N-1)/8
55   DEL(3) = -1
56   IF (E/2*2 .EQ. E) DEL(3)=1
57   IF (K .EQ. 1) GO TO 67
```

TABLE VI (continued)

```
58   WRITE (3,59) A,B,C,N,(CHAR(I),I=2,K)
59   FORMAT (1H ,2X,'(A,B,C)=',3I4,3X,'N=',I9,8(4X,I2,3X))
60   IF (Z .EQ. 0) GO TO 29
61   WRITE (3,62) DEL
62   FORMAT (1H+,111X,I2,3X,I2,6X,I2)
63   GO TO 29
64   WRITE (3,65) DST
65   FORMAT (1HO,9X,'D=',I4)
66   GO TO 27
67   WRITE (3,68) A,B,C,N,DEL
68   FORMAT (1H ,2X,'(A,B,C)=',3I4,3X,'N=',I9,75X,I2,3X,I2,6X,I2)
69   GO TO 29
70   END
```

Notes: Statement 6 should occupy 72 card columns before beginning continuation card. The value of discriminant D should be punched positive in a data card, regardless of the algebraic sign of D. The first data card in this program is punched as follows:

```
123456789
 (N, (N, (N, (N, (N, (N, (N, (N,
```

Data for the genera program comes from a program for listing reduced definite forms (See Table XXV, Appendix) and from the periods program (See Table V, Appendix).

TABLE VII

SUBROUTINE DPRFAC FOR GENERA PROGRAM

```
 1   SUBROUTINE DPRFAC (N,PR,K,PT)
 2   INTEGER PT(2),PR(10),P
 3   NSTORE = N
 4   K = 0
 5   IF (N/2*2 .NE. N) GO TO 10
 6   K = K+1
 7   PR(K) = 2
 8   N = N/2
 9   IF (N/2*2 .EQ. N) GO TO 8
10   P = 3
11   IF (P .GT. N) GO TO 19
12   IF (N/P*P .NE. N) GO TO 17
13   K = K+1
14   PR(K) = P
15   N = N/P
16   IF (N/P*P .EQ. N) GO TO 15
17   P = P+2
18   GO TO 11
19   L = 1
20   IF (P .GE. 5) GO TO 23
21   P = P+2
22   GO TO 20
23   I1 = SQRT(FLOAT(P))+1.0
24   DO 26 I2=3,I1,2
25   IF (P/I2*I2 .EQ. P) GO TO 32
26   CONTINUE
27   PT(L) = P
28   L = L+1
```

TABLE VII (continued)

```
29    IF (L .LT. 3) GO TO 32
30    N = NSTORE
31    RETURN
32    P = P+2
33    GO TO 23
34    END
```

TABLE VIII

SUBROUTINE THM13

```
 1   SUBROUTINE THM13 (A,B,C,PR,K,PT,Q)
 2   INTEGER X(10),Y(10),PR(10),A,B,C,XX,YY,Q,PT(2)
 3   DO 14 I=1,K
 4   IF (A/PR(I)*PR(I) .NE. A) GO TO 9
 5   IF (C/PR(I)*PR(I) .NE. C) GO TO 12
 6   X(I) = PT(1)
 7   Y(I) = PT(2)
 8   GO TO 14
 9   X(I) = PT(1)
10   Y(I) = PR(I)
11   GO TO 14
12   X(I) = PR(I)
13   Y(I) = PT(1)
14   CONTINUE
15   CALL CRTHM (K,X,PR,XX)
16   CALL CRTHM (K,Y,PR,YY)
17   Q = A*XX**2+B*XX*YY+C*YY**2
18   RETURN
19   END
```

TABLE IX

SUBROUTINE CRTHM

```
 1   SUBROUTINE CRTHM (K,A,MOD,XX)
 2   INTEGER A(10),MOD(10),MU(10),Q(25),MB(10),B,C,D,P,XX
 3   MM = 1
 4   DO 5 I=1,K
 5   MM = MM*MOD(I)
 6   DO 7 I=1,K
 7   MB(I) = MM/MOD(I)
 8   DO 36 J=1,K
 9   M = MB(J)
10   P = MOD(J)
11   M1 = M/P
12   M = M-M1*P
13   IF (M .LT. 0) M=M+P
14   I = 1
15   Q(I) = P/M
16   R = P-Q(I)*M
17   IF (R .EQ. 0) GO TO 22
18   I = I+1
19   P = M
20   M = R
21   GO TO 15
22   IF (I .EQ. 1) GO TO 33
23   B = -Q(1)
24   IF (I .EQ. 2) GO TO 35
25   C = 1+Q(1)*Q(2)
26   IF (I .EQ. 3) GO TO 36
27   I1 = I-1
28   DO 31 KK=3,I1
```

TABLE IX (continued)

```
29    D = -Q(KK)*C+B
30    B = C
31    C = D
32    GO TO 36
33    C = 1
34    GO TO 36
35    C = B
36    MU(J) = C
37    XX = 0
38    DO 39 J=1,K
39    XX = XX+MB(J)*MU(J)*A(J)
40    D = XX/MM
41    XX = XX-D*MM
42    IF (IABS(XX-MM).LT.IABS(XX)) XX=XX-MM
43    IF (IABS(XX+MM).LT.IABS(XX)) XX=XX+MM
44    RETURN
45    END
```

TABLE X

SUBROUTINE LEGNDR

```
 1   SUBROUTINE LEGNDR (N,P,CHAR)
 2   INTEGER N,P,CHAR,E,TEMP
 3   NST1 = N
 4   NST2 = P
 5   CHAR = 1
 6   IF (N .GT. 0) GO TO 10
 7   E = (P-1)/2
 8   IF ((E-1)/2*2 .EQ. (E-1)) CHAR=(-1)*CHAR
 9   N = -N
10   IF (N .LT. P) GO TO 13
11   M = N/P
12   N = N-M*P
13   IF (N/2*2 .EQ. N) GO TO 21
14   IF (N .EQ. 1) GO TO 25
15   E = (P-1)*(N-1)/4
16   IF ((E-1)/2*2 .EQ. (E-1)) CHAR=(-1)*CHAR
17   TEMP = N
18   N = P
19   P = TEMP
20   GO TO 11
21   N = N/2
22   E = (P*P-1)/8
23   IF ((E-1)/2*2 .EQ. (E-1)) CHAR=(-1)*CHAR
24   GO TO 13
25   N = NST1
26   P = NST2
27   RETURN
28   END
```

TABLE XI

GROUP STRUCTURE PROGRAM USING PRIME POWER

ORDER GENERATORS, MAIN PROGRAM

```
 1    IMPLICIT INTEGER (A-Z)
 2    COMMON A(40),B(40)C(40),TRY(40),D,N
 3    DIMENSION NO(40),AT(40),BT(40),CT(40),LOC(10),BPR(10),ORD(10)
 4    DIMENSION PR(6)
 5    REAL KK(10),LAB(40,40)
 6    READ (1,7) KK
 7    FORMAT (10A1)
 8    WRITE (3,9)
 9    FORMAT (1H1,//,25X,'GROUP UNDER COMPOSITION OF PRIMITIVE
     CFORMS OF DISCRIMINANT D',/)
10    READ (1,11) D,N
11    FORMAT (3I4)
12    IF (D .EQ. 9999) CALL EXIT
13    D = -D
14    DO 15 I=1,N
15    READ (1,11) A(I),B(I),C(I)
16    WRITE (3,17) D,A(1),B(1),C(1)
17    FORMAT (1HO,8X,'D= ',I5,/3X,'(A,B,C)=',3I4,6X,'1',2X)
18    IF (N .EQ. 1) GO TO 103
19    DO 20 I=1,N
20    TRY(I) = 0
21    BA = 1
22    I = 2
23    TRY(I) = 1
24    T = 1
25    AT(T) = A(I)
26    BT(T) = B(I)
```

TABLE XI (continued)

```
27   CT(T) = C(I)
28   CALL CMPOSE (A(I),B(I),C(I),AT(T),BT(T),CT(T),AC,BC,CC,D)
29   IF (CC .EQ. C(1)) GO TO 35
30   T = T+1
31   AT(T) = AC
32   BT(T) = BC
33   CT(T) = CC
34   GO TO 28
35   ORD(BA) = T+1
36   CALL DPRFAC (ORD(BA),PR,K)
37   IF (K .NE. 1) GO TO 93
38   LOC(BA) = I
39   BPR(BA) = PR(1)
40   IF (BA .EQ. 1) GO TO 63
41   BA1 = BA-1
42   DO 56 J=1,BA1
43   IF (PR(1) .NE. BPR(J)) GO TO 56
44   LO = LOC(J)
45   HI = LOC(J)+ORD(J)-2
46   GT = 0
47   IF (ORD(BA) .GT. ORD(J)) GT = 1
48   DO 55 K2=LO,HI
49   DO 55 K1=1,T
50   IF (AT(K1) .NE. A(K2)) GO TO 55
51   IF (BT(K1).NE.B(K2)) GO TO 55
52   IF (CT(K1).NE.C(K2)) GO TO 55
53   IF (GT.EQ.1 .AND. K2.EQ.LO) GO TO 58
54   GO TO 93
55   CONTINUE
56   CONTINUE
```

TABLE XI (continued)

```
57    GO TO 63
58    ORD(J) = ORD(BA)
59    BA = J
60    I = J
61    DO 62 J=I,N
62    TRT(J) = 0
63    CO = 1
64    DO 70 K=1,T
65    CALL SCAN (I,AT(K),BT(K),CT(K))
66    DO 67 J=1,CO
67    LAB(I,J) = KK(BA)
68    NO(I) = CO
69    I = I+1
70    CO = CO+1
71    I = I-1
72    IF (I .EQ. N) GO TO 98
73    IF (BA .EQ. 1) GO TO 91
74    L1 = LOC(BA)-1
75    I1 = I
76    L = LOC(BA)
77    DO 89 K=L,I1
78    DO 89 M=2,L1
79    CALL CMPOSE (A(K),B(K),C(K),A(M),B(M),C(M),AC,BC,CC,D)
80    I = I+1
81    CALL SCAN (I,AC,BC,CC)
82    NOM = NO(M)
83    DO 84 J=1,NOM
84    LAB(I,J) = LAB(M,J)
85    JL = NO(M)+1
86    JU = NO(M)+NC(K)
```

TABLE XI (continued)

```
 87   DO 88 J=JL,JU
 88   LAB(I,J) = LAB(K,J-NO(M))
 89   NO(I) = JU
 90   IF (I .EQ. N) GO TO 98
 91   I = I+1
 92   BA = BA+1
 93   DO 97 J=I,N
 94   IF (TRY(J).EQ.1) GO TO 97
 95   CALL SWITCH (I,J)
 96   GO TO 23
 97   CONTINUE
 98   DO 102 I=2,N
 99   NOI = NO(I)
100   WRITE (3,101) A(I),B(I),C(I),(LAB(I,J),J=1,NOI)
101   FORMAT (1H ,2X,'(A,B,C)=',3I4,6X,40A1)
102   CONTINUE
103   WRITE (3,104) N
104   FORMAT (1H ,8X,'ORDER OF GROUP IS',I4)
105   GO TO 10
106   END
```

Notes: Statement 9 should occupy 72 card columns before beginning continuation card. As written, the program handles groups of order ≤ 40. The first data card in this program is punched as follows:

```
1 2 3 4 5 6 7 8 9 10
A B C D E F G H I J
```

TABLE XII

SUBROUTINE DPRFAC FOR GROUP STRUCTURE PROGRAM

```
 1   SUBROUTINE DPRFAC (N,PR,K)
 2   INTEGER PR(6),P
 3   NSTORE = N
 4   K = 0
 5   IF (N/2*2 .NE. N) GO TO 10
 6   K = K+1
 7   PR(K) = 2
 8   N = N/2
 9   IF (N/2*2 .EQ. N) GO TO 8
10   P = 3
11   IF (P .GT. N) GO TO 19
12   IF (N/P*P .NE. N) GO TO 17
13   K = K+1
14   PR(K) = P
15   N = N/P
16   IF (N/P*P .EQ. N) GO TO 15
17   P = P+2
18   GO TO 11
19   N = NSTORE
20   RETURN
21   END
```

TABLE XIII

SUBROUTINE SCAN

```
 1   SUBROUTINE SCAN (I,AC,BC,CC)
 2   IMPLICIT INTEGER (A-Z)
 3   COMMON A(40),B(40),C(40),TRY(40),D,N
 4   DO 9 K=I,N
 5   IF (AC .NE. A(K)) GO TO 9
 6   IF (BC .NE. B(K)) GO TO 9
 7   IF (CC .NE. C(K)) GO TO 9
 8   GO TO 12
 9   CONTINUE
10   WRITE (3,11) I
11   FORMAT (1HO,'SCAN FAILS,I=',I4)
12   CALL SWITCH (I,K)
13   RETURN
14   END
```

TABLE XIV

SUBROUTINE SWITCH

```
 1   SUBROUTINE SWITCH (I,K)
 2   IMPLICIT INTEGER (A-Z)
 3   COMMON A(40),B(40),C(40),TRY(40),D,N
 4   T = A(I)
 5   A(I) = A(K)
 6   A(K) = T
 7   T = B(I)
 8   B(I) = B(K)
 9   B(K) = T
10   T = C(I)
11   C(I) = C(K)
12   C(K) = T
13   T = TRY(I)
14   TRY(I) = TRY(K)
15   TRY(K) = T
16   RETURN
17   END
```

TABLE XV

SUBROUTINE CMPOSE

```
 1   SUBROUTINE CMPOSE (A1,B1,C1,A2,B2,C2,AC,BC,CC,D)
 2   IMPLICIT INTEGER (A-Z)
 3   DIMENSION M2(2,2)
 4   Y = 1
 5   X = 1
 6   CALL GCD (X,D3,Y,D4,G)
 7   IF (G. NE. 1) GO TO 14
 8   DO 13 J=1,2
 9   Y = -Y
10   M = A2*X*X+B2*X*Y+C2*Y*Y
11   CALL GCD (A1,D3,M,D4,G)
12   IF (G .EQ. 1) GO TO 22
13   CONTINUE
14   X = X+1
15   IF (X .GE. 4) GO TO 17
16   GO TO 6
17   Y = Y+1
18   IF (Y .LE. 3) GO TO 5
19   WRITE (3,20)
20   FORMAT (1H ,'COMPOSE FAILS',2X)
21   CALL EXIT
22   MM = 2*M-1
23   M4 = 4*M
24   N = 0
25   NN = N*N-D
26   IF (NN/MA*M4 .EQ. NN) GO TO 33
27   N = N+1
28   IF (N .GT. 1000) GO TO 30
```

TABLE XV (continued)

```
29    GO TO 25
30    WRITE (3,31)
31    FORMAT (1H ,'N GT 1000',2X)
32    CALL EXIT
33    L = NN/M4
34    CALL REDUCE (M,N,L,MR,NR,LR,M2)
35    IF (A2 .NE. MR) GO TO 27
36    IF (B2 .NE. NR) GO TO 27
37    IF (C2 .NE. LR) GO TO 27
38    CALL GCD (M,H1,A1,H2,G)
39    A = M*A1
40    B = M*B1*H1+A1*N*H2
41    Q0 = B/(2*A)
42    B = B-Q0*(2*A)
43    C = (B*B-D)/(4*A)
44    CALL REDUCE (A,B,C,AC,BC,CC,M2)
45    RETURN
46    END
```

TABLE XVI

SUBROUTINE GCD

```
 1   SUBROUTINE GCD (MS,C,PS,E,M)
 2   IMPLICIT INTEGER (A-Z)
 3   DIMENSION Q(25)
 4   M = MS
 5   P = PS
 6   I = 1
 7   Q(I) = P/M
 8   R = P-Q(I)*M
 9   IF (R .EQ. 0) GO TO 14
10   I = I+1
11   P = M
12   M = R
13   GO TO 7
14   IF (I .EQ. 1) GO TO 25
15   B = -Q(1)
16   IF (I .EQ. 2) GO TO 27
17   C = 1+Q(1)*Q(2)
18   IF (I .EQ. 3) GO TO 28
19   I1 = I-1
20   DO 23 KK=3,I1
21   D = -Q(KK)*C+B
22   B = C
23   C = D
24   GO TO 28
25   C = 1
26   GO TO 28
27   C = B
28   E = (M-MS*C)/PS
29   RETURN
30   END
```

TABLE XVII

GROUP STRUCTURE PROGRAM USING MINIMAL NUMBER

OF GENERATORS, MAIN PROGRAM

```
 1   IMPLICIT INTEGER (A-Z)
 2   COMMON A(40),B(40),C(40),TRY(40),D,N
 3   DIMENSION NO(40)
 4   REAL KK(10),LAB(40,40)
 5   READ (1,6) KK
 6   FORMAT (10A1)
 7   WRITE (3,8)
 8   FORMAT (1H1,//,25X,'GROUP UNDER COMPOSITION OF PRIMITIVE
    CFORMS OF DISCRIMINANT D',/)
 9   READ (1,10) D,N
10   FORMAT (3I4)
11   IF (D .EQ. 9999) CALL EXIT
12   D = -D
13   DO 14 I=1,N
14   READ (1,10) A(I),B(I),C(I)
15   WRITE (3,16) D,A(1),B(1),C(1)
16   FORMAT (1HO,8X,'D=',I5,/3X,'(A,B,C)=',3I4,6X,'1',2X)
17   IF (N .EQ. 1) GO TO 66
18   DO 19 I=1,N
19   TRY(I) = 0
20   BA = 1
21   L = 2
22   I = 2
23   TRY(I) = 1
24   IF (N .EQ. 2) GO TO 26
25   CALL BASE(L)
26   CO = 1
```

TABLE XVII (continued)

```
27   LAB(I,1) = KK(BA)
28   NO(I) = CO
29   CALL CMPOSE (A(I),B(I),C(I),A(L),B(L),C(L),AC,BC,CC,D)
30   IF (CC .NE. C(1)) GO TO 32
31   GO TO 39
32   I = I+1
33   CO = CO+1
34   CALL SCAN (I,AC,BC,CC)
35   DO 36 J=1,CO
36   LAB(I,J) = KK(BA)
37   NO(I) = CO
38   GO TO 29
39   IF (I .EQ. N) GO TO 61
40   IF (L .EQ. 2) GO TO 57
41   L1 = L-1
42   I1 = I
43   DO 55 K=L,I1
44   DO 55 M=2,L1
45   CALL CMPOSE (A(K),B(K),C(K),A(M),B(M),C(M),AC,BC,CC,D)
46   I = I+1
47   CALL SCAN (I,AC,BC,CC)
48   NOM = NO(M)
49   DO 50 J=1,NOM
50   LAB(I,J) = LAB(M,J)
51   JL = NO(M)+1
52   JU = NO(M)+NO(K)
53   DO 54 J=JL,JU
54   LAB(I,J) = LAB(K,J-NO(M))
55   NO(I) = JU
56   IF (I .EQ. N) GO TO 61
```

TABLE XVII (continued)

```
57   I = I+1
58   L = I
59   BA = BA+1
60   GO TO 23
61   DO 65 I=2,N
62   NOI = NO(I)
63   WRITE (3,64) A(I),B(I),C(I),(LAB(I,J),J=1,NOI)
64   FORMAT (1H ,2X,'(A,B,C)=',3I4,6X,40A1)
65   CONTINUE
66   WRITE (3,67) N
67   FORMAT (1H ,8X,'ORDER OF GROUP IS',I4)
68   GO TO 9
69   END
```

Notes: Statement 8 should occupy 72 card columns before beginning continuation card. As written, the program handles groups of order $\leq$ 40. The first data card in this program is punched as follows:

```
1 2 3 4 5 6 7 8 9 10
A B C D E F G H I J
```

TABLE XVIII

SUBROUTINE BASE

```
 1   SUBROUTINE BASE(L)
 2   IMPLICIT INTEGER (A-Z)
 3   COMMON A(40),B(40),C(40),TRY(40),D,N
 4   L1 = L+1
 5   DO 23 I=L1,N
 6   IF (TRY(I) .EQ. 1) GO TO 23
 7   A2 = A(I)
 8   B2 = B(I)
 9   C2 = C(I)
10   CALL CMPOSE (A(I),B(I),C(I),A2,B2,C2,AC,BC,CC,D)
11   IF (CC .NE. C(1)) GO TO 13
12   GO TO 23
13   IF (AC .NE. A(L)) GO TO 19
14   IF (BC .NE. B(L)) GO TO 19
15   IF (CC .NE. C(L)) GO TO 19
16   CALL SWITCH (I,L)
17   TRY(L) = 1
18   GO TO 23
19   A2 = AC
20   B2 = BC
21   C2 = CC
22   GO TO 10
23   CONTINUE
24   IF (L .EQ. 2) RETURN
25   A2 = A(L)
26   B2 = B(L)
27   C2 = C(L)
28   CALL CMPOSE (A(L),B(L),C(L),A2,B2,C2,AC,BC,CC,D)
```

TABLE XVIII (continued)

```
29    L2 = L-1
30    DO 35 J=1,L2
31    IF (AC .NE. A(J)) GO TO 35
32    IF (BC .NE. B(J)) GO TO 35
33    IF (CC .NE. C(J)) GO TO 35
34    GO TO 40
35    CONTINUE
36    A2 = AC
37    B2 = BC
38    C2 = CC
39    GO TO 28
40    IF (J .EQ. 1) RETURN
41    DO 46 I=L1,N
42    IF (TRY(I) .EQ. 1) GO TO 46
43    CALL SWITCH (I,L)
44    TRY(L) = 1
45    GO TO 5
46    CONTINUE
47    WRITE (3,48)
48    FORMAT (1H,2X,'NO BASE FOUND',2X)
49    RETURN
50    END
```

TABLE XIX

SAMPLE RESULTS FROM REPRESENTATION PROGRAM FOR DEFINITE FORMS

All proper Representations (X,Y) of the number M by the form

[A,B,C] = [3,4,5]

M = 23

(X,Y) = (-3,2)
(X,Y) = (3,-2)

M = 60

(X,Y) = (5,-1)
(X,Y) = (-5,1)
(X,Y) = (1,3)

(X,Y) = (-5,3)
(X,Y) = (5,-3)

[A,B,C] = [2,-2,1]

M = 17

(X,Y) = (1,5)
(X,Y) = (-1,-5)
(X,Y) = (-1,3)
(X,Y) = (1,-3)

M = 97

(X,Y) = (-4,5)
(X,Y) = (4,-5)
(X,Y) = (-4,-13)
(X,Y) = (4,13)

[A,B,C] = [1,1,1]

M = 19

(X,Y) = (-2,5)
(X,Y) = (2,-5)
(X,Y) = (-5,3)
(X,Y) = (5,-3)
(X,Y) = (3,2)
(X,Y) = (-3,-2)
(X,Y) = (-2,-3)
(X,Y) = (2,3)
(X,Y) = (3,-5)
(X,Y) = (-3,5)
(X,Y) = (-5,2)
(X,Y) = (5,-2)

[A,B,C] = [1,0,1]

M = 74

(X,Y) = (-7,5)
(X,Y) = (7,-5)
(X,Y) = (-5,-7)
(X,Y) = (5,7)
(X,Y) = (-7,-5)
(X,Y) = (7,5)
(X,Y) = (5,-7)
(X,Y) = (-5,7)

TABLE XX

SAMPLE RESULTS FROM REPRESENTATION PROGRAM FOR INDEFINITE FORMS

Proper Representation (X,Y) of the number M by the form [A,B,C]:

[A,B,C] = [4,-1,-2]

M = 67

(X,Y) = (-15,-17)
(X,Y) = (15,17)
(X,Y) = (677,803)
(X,Y) = (-13,21)
(X,Y) = (13,-21)
(X,Y) = (-677,-803)
(X,Y) = (-5,-3)
(X,Y) = (5,3)
(X,Y) = (183,217)
(X,Y) = (-47,79)
(X,Y) = (47,-79)
(X,Y) = (-183,-217)

[A,B,C] = [1,4,-1]

M = -2

(X,Y) = (3,-1)
(X,Y) = (-3,1)
(X,Y) = (-1,1)
(X,Y) = (11,-3)
(X,Y) = (-11,3)
(X,Y) = (1,-1)

[A,B,C] = [1,0,-13]

M = -12

(X,Y) = (-1,1)
(X,Y) = (1,-1)
(X,Y) = (-1691,-469)
(X,Y) = (-2989,829)
(X,Y) = (2989,-829)
(X,Y) = (1691,469)
(X,Y) = (-25,-7)
(X,Y) = (25,7)
(X,Y) = (32605,9043)
(X,Y) = (155,-43)
(X,Y) = (-155,43)
(X,Y) = (-32605,-9043)
(X,Y) = (155,43)
(X,Y) = (-155,-43)
(X,Y) = (-201215,-55807)
(X,Y) = (-25,7)
(X,Y) = (25,-7)
(X,Y) = (201215,55807)
(X,Y) = (1,1)
(X,Y) = (-1,-1)
(X,Y) = (-2989,-829)
(X,Y) = (-1691,469)
(X,Y) = (1691,-469)
(X,Y) = (2989,829)

TABLE XXI

SAMPLE OUTPUT FROM PERIODS PROGRAM

REDUCED INTEGRAL FORMS OF DISCRIMINANT D>0,
ARRANGED BY PERIODS

d = 41

PERIOD 1

[2,3,-4]	δ = 1
[-4,5,1]	δ = - 5
[1,5,-4]	δ = 1
[-4,3,2]	δ = - 2
[2,5,-2]	δ = 2
[-2,3,4]	δ = - 1
[4,5,-1]	δ = 5
[-1,5,4]	δ = - 1
[4,3,-2]	δ = 2
[-2,5,2]	δ = - 2

d = 128

PERIOD 1

[4,4,-7]	δ = 1
[-7,10,1]	δ = -10
[1,10,-7]	δ = 1
[-7,4,4]	δ = - 1

PERIOD 2

[2,8,-8]	δ = 1
[-8,8,2]	δ = - 4

PERIOD 3

[4,8,-4]	δ = 2
[-4,8,4]	δ = - 2

PERIOD 4

[8,8,-2]	δ = 4
[-2,8,8]	δ = - 1

PERIOD 5

[7,4,-4]	δ = 1
[-4,4,7]	δ = - 1
[7,10,-1]	δ = 10
[-1,10,7]	δ = - 1

d = 212

PERIOD 1

[7,4,-7]	δ = 1
[-7,10,4]	δ = - 3
[4,14,-1]	δ = 14
[-1,14,4]	δ = - 3
[4,10,-7]	δ = 1
[-7,4,7]	δ = - 1
[7,10,-4]	δ = 3
[-4,14,1]	δ = -14
[1,14,-4]	δ = 3
[-4,10,7]	δ = - 1

PERIOD 2

[2,14,-2]	δ = 7
[-2,14,2]	δ = - 7

d = 397

PERIOD 1

[3,17,-9]	δ = 2
[-9,19,1]	δ = -19

TABLE XXI (continued)

REDUCED INTEGRAL FORMS OF DISCRIMINANT D>0,
ARRANGED BY PERIODS

d = 397 (continued)

PERIOD 1

[1,19,-9]	δ = 2
[-9,17,3]	δ = - 6
[3,19,-3]	δ = 6
[-3,17,9]	δ = - 2
[9,19,-1]	δ = 19
[-1,19,9]	δ = - 2
[9,17,-3]	δ = 6
[-3,19,3]	δ = - 6

d = 480

PERIOD 1

[8,8,-13]	δ = 1
[-13,18,3]	δ = - 6
[3,18,-13]	δ = 1
[-13,8,8]	δ = - 1

PERIOD 2

[6,12,-14]	δ = 1
[-14,16,4]	δ = - 4
[4,16,-14]	δ = 1
[-14,12,6]	δ = - 2

PERIOD 3

[12,12,-7]	δ = 2
[-7,16,8]	δ = - 2
[8,16,-7]	δ = 2
[-7,12,12]	δ = - 1

PERIOD 4

[7,12,-12]	δ = 1
[-12,12,7]	δ = - 2
[7,16,-8]	δ = 2
[-8,16,7]	δ = - 2

PERIOD 5

[14,12,-6]	δ = 2
[-6,12,14]	δ = - 1
[14,16,-4]	δ = 4
[-4,16,14]	δ = - 1

PERIOD 6

[13,8,-8]	δ = 1
[-8,8,13]	δ = - 1
[13,18,-3]	δ = 6
[-3,18,13]	δ = - 1

PERIOD 7

[1,2-,-20]	δ = 1
[-20,20,1]	δ = -20

PERIOD 8

[2,20,-10]	δ = 2
[-10,20,2]	δ = -10

TABLE XXI (continued)

REDUCED INTEGRAL FORMS OF DISCRIMINANT D>0,
ARRANGED BY PERIODS

d = 480 (continued)

PERIOD 9

[4,20,-5]	$\delta = 4$
[-5,20,4]	$\delta = -5$

PERIOD 10

[5,20,-4]	$\delta = 5$
[-4,20,5]	$\delta = -4$

PERIOD 11

[10,20,-2]	$\delta = 10$
[-2,20,10]	$\delta = -2$

PERIOD 12

[20,20,-1]	$\delta = 20$
[-1,20,20]	$\delta = -1$

d = 500

PERIOD 1

[11,4,-11]	$\delta = 1$
[-11,18,4]	$\delta = -5$
[4,22,-1]	$\delta = 22$
[-1,22,4]	$\delta = -5$

d = 500 (continued)

PERIOD 1

[4,18,-11]	$\delta = 1$
[-11,4,11]	$\delta = -1$
[11,18,-4]	$\delta = 5$
[-4,22,1]	$\delta = -22$
[1,22,-4]	$\delta = 5$
[-4,18,11]	$\delta = -1$

PERIOD 2

[5,20,-5]	$\delta = 4$
[-5,20,5]	$\delta = -4$

PERIOD 3

[2,22,-2]	$\delta = 11$
[-2,22,2]	$\delta = -11$

PERIOD 4

[10,10,-10]	$\delta = 1$
[-10,10,10]	$\delta = -1$

Note: δ is the number such that $\begin{pmatrix} 0 & 1 \\ -1 & \delta \end{pmatrix}$ transforms the given form to the next form, its right neighboring reduced form.

TABLE XXII

SAMPLE OUTPUT FROM GENERA PROGRAM FOR DEFINITE FORMS

CHARACTERS OF PRIMITIVE FORMS FOR GIVEN DISCRIMINANT D<0

d = - 39		(N|3)	(N|13)		
[1,1,110]	N = 289	1	1		
[2,1,5]	N = 7715	-1	-1		
[2,-1,5]	N = 9431	-1	-1		
[3,3,4]	N = 2245	1	1		
d = - 56		(N|7)	δε		
[1,0,14]	N = 9	1	1		
[2,0,7]	N = 375	1	-1		
[3,2,5]	N = 27	-1	1		
[3,-2,5]	N = 27	-1	1		
d = - 128		δ	ε	δε	
[1,0,32]	N = 1	1	1	1	
[3,2,11]	N = 3	-1	-1	1	
[3,-2,11]	N = 3	-1	-1	1	
[4,4,9]	N = 9	1	1	1	
d = - 195		(N|3)	(N|5)	(N|13)	
[1,1,49]	N = 289	1	1	1	
[7,1,7]	N = 2023	1	-1	-1	
[3,3,17]	N = 294797	-1	-1	1	
[5,5,11]	N = 74999	-1	1	-1	
d = - 352		(N|11)	δ	ε	δε
[1,0,88]	N = 81	1	1	1	1
[8,0,11]	N = 1363	-1	-1	-1	1

TABLE XXII (continued)

CHARACTERS OF PRIMITIVE FORMS FOR GIVEN DISCRIMINANT D<0

d = -352 (continued)		(N|11)	δ	ε	δε
[4,4,23]	N = 2887	1	-1	1	-1
[8,8,13]	N = 1781	-1	1	-1	-1

d = -364		(N|7)	(N|13)	δ
[1,0,91]	N = 289	1	1	1
[7,0,13]	N = 43727	-1	-1	-1
[4,2,23]	N = 198899	1	1	-1
[4,-2,23]	N = 225835	1	1	-1
[5,4,19]	N = 1445	-1	-1	1
[5,-4,19]	N = 1445	-1	-1	1

d = -420		(N|3)	(N|5)	(N|7)
[1,0,105]	N = 121	1	1	1
[3,0,35]	N = 191183	-1	-1	-1
[5,0,21]	N = 193301	-1	1	-1
[7,0,15]	N = 70807	1	-1	1
[2,2,23]	N = 582257	-1	-1	1
[6,6,19]	N = 21691	1	1	-1
[11,8,11]	N = 1331	-1	1	1
[10,10,13]	N = 4633	1	-1	-1

d = -443		(N|443)
[1,1,111]	N = 190969	1
[3,1,37]	N = 572907	1
[3,-1,37]	N = 572907	1
[9,5,13]	N = 1718721	1
[9,-5,13]	N = 1718721	1

TABLE XXII (continued)

CHARACTERS OF PRIMITIVE FORMS FOR GIVEN DISCRIMINANT D<0

d = -480		(N|3)	(N|5)	δ	ε	δε
[1,0,120]	N = 49	1	1	1	1	1
[3,0,40]	N = 4027	1	-1	-1	-1	1
[5,0,24]	N = 3581	-1	1	1	-1	-1
[8,0,15]	N = 3887	-1	-1	-1	1	-1
[11,2,11]	N = 539	-1	1	-1	-1	1
[4,4,31]	N = 6751	1	1	-1	1	-1
[8,8,17]	N = 3377	-1	-1	1	1	1
[12,12,13]	N = 1333	1	-1	1	-1	-1

TABLE XXIII

SAMPLE OUTPUT FROM GENERA PROGRAM FOR INDEFINITE FORMS

CHARACTERS OF PRIMITIVE FORMS FOR GIVEN DISCRIMINANT D>0

d = 60		(N|3)	(N|5)	δ
[1,6,-6]	N = 49	1	1	1
[2,6,-3]	N = -1267	-1	-1	1
[3,6,-2]	N = -353	1	-1	-1
[6,6,-1]	N = 479	-1	1	-1

d = 105		(N|3)	(N|5)	(N|7)
[4,3,-6]	N = -62201	1	1	1
[5,5,-4]	N = -22999	-1	1	-1
[7,7,-2]	N = -10973	1	-1	-1
[2,9,-3]	N = -104233	-1	-1	1

d = 128		δ	ε	δε
[4,4,-7]	N = -7	1	1	1
[7,4,-4]	N = 7	-1	1	-1

d = 173		(N|173)
[1,13,-1]	N = 27889	1

d = 181		(N|181)
[5,9,-5]	N = 146205	1

d = 192		(N|3)	δ	ε	δε
[1,12,-12]	N = 1	1	1	1	1
[3,12,-4]	N = 83	-1	-1	-1	1
[4,12,-3]	N = 61	1	1	-1	-1
[12,12,-1]	N = -1	-1	-1	1	-1

TABLE XXIII (continued)

CHARACTERS OF PRIMITIVE FORMS FOR GIVEN DISCRIMINANT D>0

d = 321		(N|3)	(N|5)
[5,9,-12]	N = 59405	-1	-1
[12,9,-5]	N = 294619	1	1
[2,15,-12]	N = -687815	-1	-1
[4,15,-6]	N = -45887	1	1
[1,17,-8]	N = 11881	1	1
[8,15,-3]	N = -970351	-1	-1

d = 396		(N|3)	(N|11)	δ
[9,6,-10]	N = -5299	-1	1	1
[7,12,-9]	N = 1183	1	-1	-1
[5,14,-10]	N = 845	-1	1	1
[10,6,-9]	N = -9761	1	-1	-1
[1,18,-18]	N = 169	1	1	1
[2,18,-9]	N = -20881	-1	-1	-1
[9,18,-2]	N = -3803	1	1	1
[18,18,-1]	N = 5435	-1	-1	-1

TABLE XXIV

SAMPLE OUTPUT FROM GROUP STRUCTURE PROGRAMS

PRIME POWER ORDER GENERATORS		MINIMAL NUMBER OF GENERATORS	
d = -440		d = -440	
[1,0,110]	1	[1,0,110]	1
[2,0,55]	A	[7,-6,17]	A
[5,0,22]	B	[9,8,14]	AA
[10,0,11]	AB	[2,0,55]	AAA
[9,8,14]	C	[9,-8,14]	AAAA
[9,-8,14]	CC	[7,6,17]	AAAAA
[7,6,17]	AC	[5,0,22]	B
[3,-2,37]	BC	[6,-4,19]	AB
[6,4,19]	ABC	[3,-2,37]	AAB
[7,-6,17]	ACC	[10,0,11]	AAAB
[3,2,37]	BCC	[3,2,37]	AAAAB
[6,-4,19]	ABCC	[6,4,19]	AAAAAB
$A^2 = B^2 = C^3 = 1$ ORDER OF GROUP IS 12		$A^6 = B^2 = 1$ ORDER OF GROUP IS 12	
d = -471		d = -471	
[1,1,118]	1	[1,1,118]	1
[2,1,59]	A	[8,-3,15]	A
[4,-3,30]	AA	[10,-3,12]	AA
[8,-3,15]	AAA	[6,-3,20]	AAA
[10,7,13]	AAAA	[10,-7,13]	AAAA
[5,-3,24]	AAAAA	[2,-1,59]	AAAAA
[10,-3,12]	AAAAAA	[4,-3,30]	AAAAAA
[6,3,20]	AAAAAAA	[5,-3,24]	AAAAAAA
[3,3,40]	AAAAAAAA	[3,3,40]	AAAAAAAA

TABLE XXIV (continued)

PRIME POWER ORDER GENERATORS		MINIMAL NUMBER OF GENERATORS	
d = -471 (continued)		d = -471 (continued)	
[6,-3,20]	AAAAAAAAA	[5,3,24]	AAAAAAAAA
[10,3,12]	AAAAAAAAAA	[4,3,30]	AAAAAAAAAA
[5,3,24]	AAAAAAAAAAA	[2,1,59]	AAAAAAAAAAA
[10,-7,13]	AAAAAAAAAAAA	[10,7,13]	AAAAAAAAAAAA
[8,3,15]	AAAAAAAAAAAAA	[6,3,20]	AAAAAAAAAAAAA
[4,3,30]	AAAAAAAAAAAAAA	[10,3,12]	AAAAAAAAAAAAAA
[2,-1,59]	AAAAAAAAAAAAAAA	[8,3,15]	AAAAAAAAAAAAAAA
$A^{16} = 1$ ORDER OF GROUP IS 16		$A^{15} = 1$ ORDER OF GROUP IS 16	
d = -615		d = -615	
[1,1,154]	1	[1,1,154]	1
[3,3,52]	A	[8,-5,20]	A
[4,3,39]	B	[10,-5,16]	AA
[10,5,16]	BB	[2,1,77]	AAA
[10,-5,16]	BBB	[4,3,39]	AAAA
[4,-3,39]	BBBB	[5,5,32]	AAAAA
[12,3,13]	AB	[4,-3,39]	AAAAAA
[7,1,22]	ABB	[2,-1,77]	AAAAAAA
[7,-1,22]	ABBB	[10,5,16]	AAAAAAAA
[12,-3,13]	ABBBB	[8,5,20]	AAAAAAAAA
[5,5,32]	C	[3,3,52]	B
[14,13,14]	AC	[11,-1,14]	AB
[8,5,20]	BC	[7,-1,22]	AAB
[2,1,77]	BBC	[6,-3,26]	AAAB
[2,-1,77]	BBBC	[12,3,13]	AAAAB
[8,-5,20]	BBBBC	[14,13,14]	AAAAAB

TABLE XXIV (continued)

PRIME POWER ORDER GENERATORS		MINIMAL NUMBER OF GENERATORS	
d = -615 (continued)		d = -615 (continued)	
[11,1,14]	ABC	[12,-3,13]	AAAAAAB
[6,-3,26]	ABBC	[6,3,26]	AAAAAAAB
[6,3,26]	ABBBC	[7,1,22]	AAAAAAAAB
[11,-1,14]	ABBBBC	[11,1,14]	AAAAAAAAAB
$A^2 = B^5 = C^2 = 1$ ORDER OF GROUP IS 20		$A^{10} = B^2 = 1$ ORDER OF GROUP IS 20	
d = -759		d = -759	
[1,1,190]	1	[1,1,190]	1
[10,1,19]	A	[6,-3,32]	A
[3,3,64]	AA	[4,-3,38]	AA
[10,-1,19]	AAA	[8,3,24]	AAA
[12,3,16]	B	[12,3,16]	AAAA
[12,-3,16]	BB	[2,-1,95]	AAAAA
[7,-5,28]	AB	[3,3,64]	AAAAAA
[4,3,48]	AAB	[2,1,95]	AAAAAAA
[10,-9,21]	AAAB	[12,-3,16]	AAAAAAAA
[10,9,21]	ABB	[8,-3,24]	AAAAAAAAA
[4,-3,48]	AABB	[4,3,48]	AAAAAAAAAA
[7,5,28]	AAABB	[6,3,32]	AAAAAAAAAAA
[14,5,14]	C	[14,5,14]	B
[8,3,24]	AC	[10,-9,21]	AB
[11,11,20]	AAC	[14,9,15]	AAB
[8,-3,24]	AAAC	[10,1,19]	AAAB
[5,-1,38]	BC	[5,-1,38]	AAAAB
[5,1,38]	BBC	[7,5,28]	AAAAAB

TABLE XXIV (continued)

PRIME POWER ORDER GENERATORS		MINIMAL NUMBER OF GENERATORS	
d = -759 (continued)		d = -759 (continued)	
[2,1,95]	ABC	[11,11,20]	AAAAAAB
[14,-9,15]	AABC	[7,-5,28]	AAAAAAAB
[6,-3,32]	AAABC	[5,1,38]	AAAAAAAAB
[6,3,32]	ABBC	[10,-1,19]	AAAAAAAAAB
[14,9,15]	AABBC	[14,-9,15]	AAAAAAAAAAB
[2,-1,95]	AAABBC	[10,9,21]	AAAAAAAAAAAB
$A^4 = B^3 = C^2 = 1$ ORDER OF GROUP IS 24		$A^{12} = B^2 = 1$ ORDER OF GROUP IS 24	

TABLE XXV

PROGRAM FOR LISTING REDUCED DEFINITE FORMS

```
 1   INTEGER D,CC,AC,A,C,BB,B(100)
 2   D = 0
 3   CC = 3
 4   D = CC+D
 5   IF (D .GT. 500) CALL EXIT
 6   WRITE (3,7) D
 7   FORMAT (1H ,5X,'D=',I5)
 8   CC = 4-CC
 9   L = SQRT(FLOAT(D/3))
10   IF (D/4*4 .EQ. D) GO TO 17
11   J = 1
12   K = (L+1)/2
13   DO 15 I=1,K
14   B(I) = J
15   J = J+2
16   GO TO 22
17   J = 0
18   IF (L/2*2 .EQ. L) GO TO 20
19   GO TO 12
20   K = (L+3)/2
21   GO TO 13
22   DO 38 I=1,K
23   AC = (B(I)*B(I)+D)/4
24   M = SQRT(FLOAT(AC))
25   DO 38 A=1,M
26   C = AC/A
27   IF (C*A .EQ. AC) GO TO 29
28   GO TO 38
```

TABLE XXV (continued)

```
29    IF (B(I) .LE. A) GO TO 31
30    GO TO 38
31    WRITE (3,32) A,B(I),C
32    FORMAT (1H ,5X,3I6)
33    IF (B(I) .EQ. 0) GO TO 38
34    IF (-A .LT. -B(I) .AND. A .NE. C) GO TO 36
35    GO TO 38
36    BB = -B(I)
37    WRITE (3,32) A,BB,C
38    CONTINUE
39    GO TO 4
40    END
```

VITA

David Hume was born in Knoxville, Tennessee, on January 18, 1945. He attended public schools in that city and graduated from West High School in 1962. He entered the University of Tennessee the following September, receiving a Bachelor of Science in Mathematics in June, 1966. He attended the University of California at Berkeley from 1966 to 1968. Since September, 1968, he has been employed as Instructor of Mathematics at Western Carolina University. He is a member of Phi Kappa Phi and Phi Beta Kappa.