Nate Randall

ネイト・ランダル

* =

*P

*W (number)

*C omposer

*S ubtitle

*Q

*L

*E

*

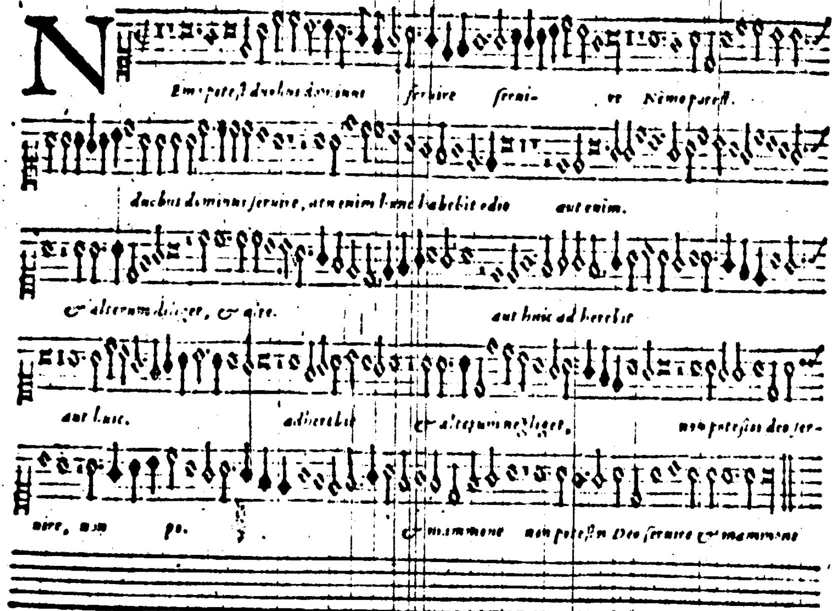
FASTCODE: A Code for Polyphonic Music

[Designed for White Mensural Notation]

Users' Manual

Thomas Hall
January 1975
Revised, October 1976
Revised, June 1978
Edition for Susato/Lassus Project
Revised October 1981

Ioannes Louuys   Matthei, Sexto


N
Emo potest duobus dominis   seruire   serui-   re   Nemo potest.
duobus dominis seruire, aut enim vnum habebit odio   aut enim.
& alterum diliget, & aite.   aut huic adherebit
aut huic.   adherebit   & alterum negliget.   non potestis deo ser-
uire, non   po.   & mammone   non potestis Deo seruire & mammone

Chapter 1

FASTCODE: SOME SAMPLE CODING

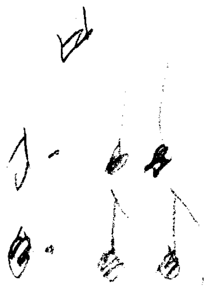
FASTCODE has few syntactical requirements and these may be learned simply by inspecting some coded music. The present chapter will therefore discuss briefly the coding of a short section of music--the opening of the Superior voice of the motet Nemo potest by Johannes Louys as found in the ninth book of Latin motets published by Susato in 1554 [RISM 1554/9]. A photocopy of the appropriate folio is given as an insert. This chapter is intended as a general introduction only--it does not discuss all of the features of the language; a complete, systematic explanation constitutes Chapter 2, which the reader must study after he has followed the coding in Chapter 1.

The first step in coding a composition is to enter, before the coding of any actual notes, the general bibliographical and musical information that applies to the compositon, or sub-section (1) or sub-section of the composition, as a whole. The information may be entered in any order.

For instance, to enter the title (Code-Character = '*T'), the name of the composer (CC = '*C'), and the title of the sub-section (CC = '*S') of this composition, the following might be punched(2):

3

```
*T MATTHEI, SEXTO
*C IOANNES LOJUYS
*S NEMO POTEST
```



CU39+*=

C139+*=

D1+*1

----

(1) The word 'sub-section' is here used, rather than 'section', to avoid confusion between, for example, the Kyrie I (sub-section) and the complete Kyrie (section) of a mass. In other genres, 'sub-section' is used for the parts of a motet, the stanzas of a hymn, the verses of a Magnificat, etc.

(2) FASTCODE is specifically designed for input on the keypunch, which makes it difficult to enter lower-case letters. A solution to this problem is given in Chapter Two.

(Each of these lines represents, of course, a separate punch-card.) Additional bibliographical information may be entered similarly under several other code characters. The siglum of a source is, by convention, stored under the code character '*P' (for publisher)[3] e.g.[4],

```
*P 1554/9 SUSATO BK. 9
```

Any of the above information could be omitted in the coding; however, certain general musical information concerning the number of 'voices' or 'parts' of music, and information concerning the clefs, 'key-signatures', and 'time-signatures' used for each voice <u>must</u> precede the coding of the actual notes. This information also may be entered in any order and may even precede or be intermingled with the above bibliographical information.

The plotting program calculates how many 'parts' or 'voices' of coded music will be processed from a card (CC ='*L') indicating the number of voices; in this case, since the motet has five voices, the following *L-card could be submitted:

```
*L5
```

Information about the clefs, 'key-signatures', and 'time-signatures' must precede the coding of the actual notes. The clefs (CC = either '*<' or '*>') for this section could be entered as follows:

```
*< C1 C3 C3 C4 F
```

There are no spacing requirements on a *<-card; thus, the following erratic spacing would also be acceptable:

```
*<C   1
```

[3] This convention, like some others, is explainable on historical rather than logical grounds. When the system was devised, coding from manuscript sources was not contemplated.

[4] For the purposes of the Susato/Lassus Project at Princeton, the *P-card should contain for a printed source the appropriate RISM number, or for a manuscript the proper siglum from the lists prepared by Professor Charles Hamm for the Illinois Archives.

The 'key-signature' could be entered as:

```
*K C
```

(the computer will assume that the same 'key'- or mensuration designation refers to all voices) or by:

```
*K C C C C C
```

There are no spacing requirements on a *K-card.

The mensuration-signature could be entered by:

```
*M C/
```

or by:

```
*MC/ C/ C/ C/ C/
```

Each successive designation of mensuration or mensuration-signature must be separated from the preceding by a space. One codes 'C/' to represent ₵ and '0/' to represent Φ. Representations of more complex mensuration-signatures are given in Chapter Two and in Appendix A.

On both the *M(ensuration)-card and the *K(ey)-card, if only one unit of information is entered, this is taken to apply to all succeeding voices of coded music in the section; if more than one, each unit is applied in order to each successive part of coded music.

The information about the 'key', 'meter' or mensuration, and clefs may be entered either all at once for all parts or voices, before the coding of any notes, or singly, before the coding of each individual voice. This latter method is useful when coding from partbooks(5).

Because a large number of the motets in the Susato prints are written without signature accidentals and in ₵, provision has been made for the computer to supply such information by 'DEFAULT', if it is found lacking in the coding. Thus it is not necessary to provide a *K-card for compositions written without

(5) For instance, the *< card *<C1 would be used for our example.

signature-accidentals, or an *M-card for compositions writtan in ¢. Other DEFAULT values are noted below and in Chapter Two.

In summary, the following coding would be appropriate up to this point:

coding	comments
*TMATTHEI, SEXTO	not necessary, except for tha first section of a work.
*CIOANNES LOUUYS	ditto
*P1554/9 SUSATO BK. 9	ditto
*MC/	not necessary--provided by DEFAULT.
*KC	not necessary--provided by DEFAULT.
*<C1	
*L5	

(The above cards could be submitted in any order.)

* * * * *

The coding of tha actual notes may now begin. The information concerning a single note may be divided into: a) information about the pitch of the note, b) information about the duration, and c) additional information; every coded note need not include all three types of information, but the information must be entered in this order. More than one note may be coded on a single note-card by seprating each note from the previous by a space. The first seven notes and rests of the example may be coded as follows:

```
R11 R1 C11. B1 C11 G2 C
```

Pitch is indicated by the diatonic letter names ('ABCDEFG'); rests, by the letter 'R'. Durations are indicated by the numbers shown in the following table:

1	Semibreve
2	Minim
4	Semiminim
8	Fusa
16	Semifusa
11	Breve (imperfect = 2 semibreves) RW
111	Longa (imperfect = 4 semibreves) — R ✓
1111	Maxima (imperfect = 8 semibreves)

Other note-values can be coded by combining any of the above values with ties (see Chapter Two) or by dots. Augmentation dots--that is, dots that indicate that the duration of a note is to be extended by half the value indicated by the preceding symbol--are represented by periods; more than one dot for a note would be represented by the appropriate number of periods.

If the duration of a note is the same as the preceding note's duration, it may be omitted in the coding; thus, only the first of the two consecutive minims in the example has the duration indication '2'. This feature speeds the coding considerably.

Breve and longa rests may be coded as RW and RL, respectively; they will be interpreted as triple or duple according to the mensuration-signature. The savings in key-strokes may be considerable. See Chapter Two for a complete discussion of this.

To avoid the necessity of pressing the 'numeric' key on the keypunch each time a number is to be punched, the translating program has been designed to accept, in place of the number indicating the duration, the letter or character found on the same key as the number. Thus, the first ~~six~~ seven notes of the coding shown above could also be coded:

```
     RUU  RU  CUU. BU CUU GI C
```

This device <u>greatly</u> shortens the coding-time and should be used by every coder. At first, the mental conversions may be difficult to make if coding from a terminal. The difficulties in correcting the code are eliminated by a syntax-checking program, in which the conversion to durations in numerical values is made.

If entering data on a terminal with no word-wrap capability, the prompt ("beeper") should be audible to avoid entering lines longer than 80 characters(6).

If entering data on a kaypunch, the following drum card should be used to avoid running out of card for a note:
1AAAA1AAAAA1AAAAA1 ...etc. to col. 70: 1AAAAAAA---
The 'skip'-key is hit between notes or rests--this will allow a new card to eject and register after the note or rest entered in col. 70.

[6] At present, no public terminal in the Princeton facility has word-wrap capability.

If accidentals occur within the line of music, one of the following characters is placed <u>before</u> the pitch-letter:

N for natural
S for sharp
T for flat

To include a flat, natural, or sharp as an editorial suggestion, place it in parentheses.

In order to differentiate any note from its octave-equivalent, coding is done in 'registers.' The normal register for the treble clef extends from the first line of the staff (first line being the bottom line) to the fourth line--that is, from E up to D, inclusive. If the music goes above the fourth line, a 'U' (for 'upper') must be placed on the first note in the new register. As long as the music remains in that new register, there is no need for any further indication; in other words, this new register becomes the normal register. If the music comes back down out of this register into the next lower register, 'L' must be indicated on the first note within this lower register. Thus, in the first line, beginning with the A minim over the word 'duobus', could be coded:

```
C2 UE E LD2. D4 C2. B8 A B2
```

(Note the omission of the duration for a note or rest of the same duration as the immediately preceding note or rest.)

The 'U' or 'L' must come before the pitch-letter (but may precede or follow any indication of an accidental). The computer assumes that each voice will begin in the normal register; if it does not, the proper register must be indicated on the first note.

There must not be any spaces between the coding for accidentals, register, pitch, or duration.

Failure to indicate a change in register will cause all the music in that voice to be misplaced by an octave for the remainder of the section. This is not a particularly grievous error--any such octave-transposition will be quickly detected when the music is printed out for proofreading and is easily corrected. The error is, however, persistent; the coding of the register-changes is perhaps the most bothersome aspect of the language.

To alleviate this problem, the program has been designed to generate automatically many of these register-changes. To do this, the program assumes that normally there will be no leaps of a seventh in the music being coded. If the music <u>moves by step</u> across the register- boundary--that is, from fourth-line D up to E or from first-line E down to D (in the treble clef)--the program <u>automatically</u> provides the 'U' or 'L', without which the coded music would contain a leap of a seventh(7). Thus, the same passage could be coded more economically:

```
C2 UE E D2. D4 C2. B8 A B2
```

Notice that the 'U' was still necessary when the music <u>leaped</u> across a register boundary--here, from C to UF. Rests are neutral in register and are ignored by the translating program when generating register changes; thus, melodic movement from D to E will still cause a register change to be generated even when a rest or rests intervene.

This automatic generation of register changes greatly facilitates the coding--the reader is advised to make full use of this convenience and to supply a 'U' or 'L' only where a register-change occurs by skip.

Because the register-changes occur in different places on the staff in different clefs, and for several other reasons, experience has shown that the coder should make use of another convenience provided by the language: coding can be done as if all parts are written in the treble clef. The use of this expedient increases the speed of the coding, even by those thoroughly familiar with the various C and F clefs and significantly reduces the frequency of errors(8).

----

(7) If a seventh between E and D actually occurs in the music, the automatic generation of registers must be turned off by submitting an *X-card (see Chapter Two). The program assumes, by DEFAULT, that the user wishes the automatic generation of registers to be functioning.

(8) The not infrequent occurrence of clefs such as C2, C5, F3, and F5 in mensural sources increases the usefulness of this expedient, as does the fact that "altus" parts are often written in clefs other than C3 and "tenor" parts often in clefs other than C4. Changes of clef are also not infrequent. It has proved easy to forget what clef prevails at a given moment--above all when one has been interrupted in the course of coding. For these reasons, the programming necessary for the automatic generation of registers in clefs other than the treble clef has not been added to the translating program. Coders who wish to code the original clefs must

Therefore all coding and proof-correcting should be done as if the G clef on the second line prevailed. The computer will make the necessary transposition to the clef designated on the *<-card for the voice involved. A coder therefore does not need to know how to read any clef other than the treble-clef. Even the most experienced clef-reader however, will find that coding only in the treble-clef is by far the easiest, speediest, and most accurate method.

A number of successive notes of the same pitch and duration, or rests of the same duration, may be indicated by placing an integer before the note-information to indicate the number of repetitions; thus, if a piece contains a passage of 13 breve rests, only '13RUU' need be entered. Any addtional information indicated (see below on tags) will be taken to apply only to the last note of the series.

Changes in mensuration, 'key' or clef occurring within a voice are coded by simply interrupting the stream of notes at the appropriate point and supplying an *M-, *K-, or *<-card. Nota Bene: after any change of clef, key, or mensuration that occurs within a voice, the normal mid-staff register is assumed, regardless of the register of the preceding note. Thus a note above or below that register must be coded with a U or an L.

Once a section of music has been printed out for proofreading, the coder will occasionally discover that two different mensurations occur simultaneously in two different voices and that the measure-length for both voices is meant to remain the same. A 'reduce'-card (CC = '*-') should then be introduced at the mensuration change in one of the voices to indicate the 'true' value of the duration of the notes in one mensuration in relation to those in the other mensuration. Thus, a card reading:

*-3/2

should be introduced just after the first *M-card to indicate that in this voice the notes are to be interpreted as two thirds of their written value. (The fraction indicates that three notes are to have the same total duration that two notes had previously.)

A card (in this case '*-2/3' that nullifies the effect of the reduce card is produced automatically by the program when either a new *M-card or the end of the voice is encountered.

therefore indicate all register changes (see Chapter Two under 'Register').

The end of a sub-section of music is indicated by three slashes; for instance, the final three notes in example could be entered:

```
C1. C2 C111///
```

These final three slashes are extremely important--if omitted, the computer will not be able to distinguish this voice from the following. After the three slashes, each voice <u>must</u> begin on a new card.

How to encode such 'additional information' as coloration, triplets, fermatas, etc., is explained in the following chapter.

In the initial coding of the music, text is omitted. It can be supplied later according to a procedure to be described below.

Finally, perhaps the most useful feature of the coding language: the character '#' punched on a note card anywhere within the information-field about a note (in other words, anywhere in the space between two notes) will cause all of that information to be ignored. Thus, if an error is punched and detected immediately, it is not necessary to duplicate the correct information about other notes already punched on the card; one may merely punch this erase character (be sure not to punch <u>over</u> any other information), skip to the next note position, and re-code the correct note. On a terminal, of course, one may back-space over the error to correct it.

IMPORTANT HINTS FOR THE CODER

Coding is not even a reasonable academic exercise; it is an unpleasant task which the researcher will want to finish as quickly and painlessly as possible. A scholar's normal concerns for orderliness and mental 'cleanness' may in coding prove a hindrance. Time spent in making the code musical or excessively orderly is wasted--the computer, the only entity which will normally read the code--does not know the difference. Speed and accuracy are the essential qualities of good coding. Therefore:

* Always code as if all music were written in the treble-clef. Even the most experienced clef-reader will make mistakes, particularly in indicating register changes, when constantly switching from clef to clef. The coding should become, as much as possible, a mechanical process; reading the music in the wrong clef will be aggravating and troublesome for a short while only. A musical reading of the source may, and should, be done when proofreading from the computer-generated transcription.

* code the 'U' and 'L' indicating register changes only when the music enters a new register by skip.

* always use a drum-card on the keypunch machine when coding note cards. In this way, the skip-key can be used to jump to the next position after each note is coded.

* if coding from a keypunch, look ahead for any clef-, 'key'- or mensuration-changes and prepare appropriate cards before beginning the coding of the notes for a section of music. These may then be manually slipped into the coding stream without interrupting the automatic duplicating feature of the drum card.

* use the character '#' without hesitation.

Chapter 2

FASTCODE AND THE SYNTAX-CHECKING PROGRAM

After the music is coded, it is run through a syntax checking program. This program greatly simplifies the coding process in the following ways:

a) An elementary diagnostic examination of the coded music is performed. The SYNTAX program will not produce FASTCODE that cannot be read by the plotting program; if difficulties are encountered, ERROR-messages or NOTEs will be produced.

b) A map of the measure-numbers of the voices, systems, and sections of the composition is printed; the register of the final note of each voice is also given.

c) Non-numeric duration values are translated into their numeric equivalents for ease of correcting.

d) Any omission of certain other kinds of information, such as the title or composer of the composition, will be NOTEd by the program.

e) Information too long will be automatically truncated to the maximum length allowed and an ERROR message produced.

f) Strange registral leaps in the coded music will be NOTEd.

2.1 GENERAL CONSIDERATIONS

In the description of the code below, 'CC' refers to the code-character that must be entered before any other information on a card; 'DATA' indicates the nature and form of the information to be entered on the card; 'TRANS' refers to possible actions taken--and the ERROR messages and NOTEs produced by the SYNTAX program; and 'DEFAULT' refers to the information that will be supplied automatically if omitted in the coding.

The information to be coded may be divided into three categories: a) preliminary bibliographical information, b) general musical information, and c) note-specific information. The preliminary and general information may be entered in any order, but must precede the note-specific information.

2.2 PRELIMINARY BIBLIOGRAPHICAL INFORMATION

Preliminary bibliographical information may be submitted in any order or omitted. The INPUT information may contain internal spaces. There are limitations on the length of the information that may be stored under each CC the maximum number of characters permissible is indicated below in parentheses. "By convention" indicates practices adopted at Princeton for the Susato/Lassus project. The SYNTAX program will automatically convert to lower case all <u>but the first letter</u> of literal keypunched DATA entered as preliminary bibliographical information unless such action is overridden by any of the special literal codes described in paragraph 2.5 below. No more than one card for any of the CCs may be submitted for the same section of music.

CC		comments
*T	TITLE (44)	1. DATA: by convention, the title of the composition being coded. 2. TRANS: if INPUT too long, truncate to 44 characters and produce an ERROR message; if more than one *T-card submitted for the same section of music, ignore second, ERROR message.
*C	COMPOSER (32)	1. DATA: by convention, the composer of the music being coded. 2. TRANS: truncate excess over 32 characters; if more than one per section of music, ignore second, ERROR.
*S	SUBTITLE (24)	1. DATA: By convention, the text-incipit of the composition. 2. TRANS: truncate excess characters, ERROR; if more than one per section of music, second ignored, ERROR.
*P	PUBLISHER (24)	1. DATA: by convention, the siglum of the manuscript, or, for a printed edition, the RISM indication. 2. TRANS: truncate excess characters, ERROR.

* W

*Q KEYPUNCHER (28) 1. DATA: By convention, the name of the encoder.
2. TRANS: truncate excess characters, ERROR.

2.3 GENERAL MUSICAL INFORMATION

Cards containing general musical information may be submitted in any order and may precede or may be intermingled with the general bibliographical information. DATA information that refers to the various parts or voices of music must be in the same order that the various parts or voices will be coded. Generally, the coding is done in order of the 'highest' voice to the 'lowest'-- the term 'highest' and 'lowest' referring only to the desired position in the graphic representation of the score, not necessarily to relative pitch.

*L [LYNE]

a) DATA: a single integer indicating the number of the 'parts' or 'voices.' Uniform voice names, according to the titlepage of the print, will be supplied later.

b) DEFAULT: '(1) (2) (3) (4) (5)' [Parentheses are used by convention, here and elsewhere, to indicate information that does not appear in the source being coded.]

*< or *> [clef]

a) DATA: a list of the clefs of the various parts, in the order in which the parts are to be coded, or a single clef. There are no spacing requirements--that is, spaces may be introduced or omitted anywhere. The clefs are entered in the following code: the letter 'G' (G-clefs), 'C' (C-clefs), or 'F' (F-clefs), followed by a number that indicates the line of the staff (counting from the lowest up). For instance, 'C1' indicates a soprano clef. 'U' (upper) or 'L' (lower) before a clef-name indicates an octave transposition. 'G' or 'F', without any number, may be used to indicate the most common positions (G2 and F4) of those clefs. Clef information may be entered either in a list of all the clefs, before any note information, or singly before the coding of each individual part of music. To indicate a change of clef, the stream of notes is interrupted at the appropriate point and a *< card is introduced.

b) TRANS: if more than one clef is specified and the number of clefs specified differs from the number of voices, an ERROR message will be produced.

*K [KEY]

a) DATA: the 'key-signature', or signature-accidentals, for one or more parts of music, represented in either of two ways:

1) by a letter or letters signifying the tonic(s) of the major key(s) which in tonal music would be indicated by the 'key-signature'. 'T' and 'S' are used to signify 'flat' or 'sharp', respectively. Thus, the signature of two flats would be coded as 'BT'. There are no spacing requirements.

2) by a list(s) of the actual sharps and/or flats involved; e.g., '(AS,BT,GS)', where the (hypothetical) signature consists of sharps on A and C and a flat on B. Each list for each signature must be enclosed in parentheses and the accidentals within the list must be separated by commas. There are no spacing requirements.

For either method of representing the signature-accidentals, if only one signature is given on a card, that signature will be applied to all voice lines of music in the section, unless another *K-card is encountered just before the coding for a new voice. Thus, if all voices of a section have the same signature, one need enter only one signature on the *K-card. If the 'key-signatures' differ for the various voices, the signatures can be entered either as a list (no spacing requirements) on a single *K-card before the coding of any notes, or singly on individual *K-cards before each voice.

To indicate a change of 'key', the stream of notes is interrupted at the appropriate point and a *K card with only one component is introduced; any *K-card introduced during the coding of the actual notes for a single voice will not affect the opening 'key signatures' of any other voices.

b) DEFAULT: no signature-accidentals ('key' of C).

*M [MENSURATION]

a) DATA: one or more mensuration-signatures or mensuration-signatures. A signature must have no internal spaces. If several signatures are specified, they must be separated from each other by at least one space. Time-signatures are entered in fractional form; e.g., '4/4'. The numerator divided by the denominator must equal the number of whole notes desired in each measure. If it does not, or if a mensuration-signature is given, the correct fractional interpretation must be given on a *R(etime)-card, unless such an *R-card will be supplied by DEFAULT (see below). For instance, if one codes '0'[9] for tempus perfectum, the computer will automatically supply an *R-card equivalent to ['*R 0 3/1'] which will cause the program to interpret the '0' as 3/1 and provide barlines accordingly. If the coder wishes the '0' to be interpreted as 3/2, he must supply an appropriate *R-card to override the DEFAULT. Conventional codes for the various mensuration-signatures, and the fractional Retime values supplied by DEFAULT, are given in Appendix A. The code for a mensuration signature may not contain parentheses.

If only one signature is given on a card, it is taken to apply to all succeeding voices until another *M-card is encountered when the coding of a new voice begins. To indicate a change of mensuration in a voice, the stream of notes is interrupted at the appropriate point and an *M-card with only one component is introduced; any *M-card introduced during the coding of the actual notes will not effect the opening signature of any other voice.

b) DEFAULT: 'C/' (=2/1) for all voices.

*R [RETIME]

a) DATA: two components, separated from each other by at least one blank. The first component is a mensuration-signature that corresponds to a signature given on an *M-card (see above); the second is a mathematical fraction that indicates how the first component is to be interpreted; this fraction must be equal to the number of whole-notes desired in each measure. For example, entering: '*R C/ 1/1' will cause C/ to be interpreted as 1/1, one whole note per measure. An *R-card may be introduced at any point in the coding and remains in effect only until another *M-card is encountered, at which point the interpretation of all mensurations

[9] On both the *M and *R cards, the SYNTAX program will automatically convert any zeros to O's--thus, either is permissible.

reverts to the DEFAULT values for those mensurations. DEFAULT *R-cards will be supplied automatically for a number of common mensuration-signatures. These default values are given in Appendix A.

b) TRANS: if more or less than two components are specified, ERROR.

*- [REDUCE]

a) DATA: a single fraction in the form 'numerator/denominator' (e.g. 3/2), without internal spaces. In some compositions, one will run across cases of two different mensurations occurring simultaneously in different voices where the measures in both cases are meant to be the same length. Here, a REDUCE card must be used to indicate the 'true' value of the duration of the notes in one meter in relation to those in other meters. Thus,

*-3/2

would indicate that from this point the durational values of all notes in this voice are to be interpreted as two-thirds of their values in the other voices. This reduced value will stay in effect for that voice only until the program encounters: a) a new reduce card, either countermanding the previous or setting new values, b) the end of the section for that voice, or c) an *M-card. A reduce card may be introduced at any relevant point in the coding. Note that if a section of sesquialtera ('*-3/2') is introduced in one or more voices _against_ prevailing tempus imperfectum ('*MC/') in one or more other voices, an *R-card '*R3 6/2' (the actual mensuration-signature indicating the sesquialtera may vary, but is usually a '3' or '3|2') must be introduced with the new *M-card to insure that the length of the measures is the same for both the sesquialtera voices and the voices that remain in the original mensuration. Likewise, if *-3/2 is introduced in a prevailing tempus perfectum ('O'), the *R-card '*R3 9/2' is necessary to make the length of the reduced measures equal the length of the measures of tempus perfectum.

b) TRANS: if more than one component found, ERROR.

*X [TUCK and UNTUCK]

a) DATA: 'TUCK' specified anywhere on the card indicates to the translating program that from this point on the encoding of the staff-positons of the notes will be done using the diatonic letter-names indicated by the actual clef in the music; 'TUCK' also turns off the automatic generation of registers (see below under 'register'). If 'UNTUCK' or nothing is specified, the program assumes that the encoding of the staff-positions of the notes will be done as if they were in the treble-clef; 'UNTUCK' turns on the automatic register-generation. 'NOREG' or 'REG' specified in the card field will cause the automatic register generation to be turned off or on, respectively, but will leave the TUCK or UNTUCK mode unchanged.

b) DEFAULT: UNTUCK mode (all coding to be done as if in treble-clef) and the automatic generation of registers turned on. The encoder is advised not to use *X-cards at any time, except to countermand the automatic generation of registers ('*X NOREG') when a leap of a seventh between the first and fourth staff lines is found.

** [non-analyzable comment]

a) DATA: a comment of any length.

2.4 NOTE-SPECIFIC INFORMATION

The coding concerning each note may be divided into four components: 1) the pitch-group, 2) the duration, and 3) additional information. Although the information about a single note need not include all three components, the components must be entered in the order given above--that is, first the pitch-group, then the duration, etc. The information included on a note-card may be summarized as follows:

1. Pitch-group

a) Staff-position.

Staff-positions are indicated by the letter names of the notes:

A B C D E F G

A rest is indicated by the letter 'R'. Generally, to facilitate coding, all staves are coded as if they were written in the treble clef (cf. the *X-card above and Chapter One.) The letter Z may be used to indicate a dummy note of specified duration. A Z plus a duration code will insert blank horizontal spacing in a voice and may be used to insure the proper vertical alignment of the voices in a score by 'padding' any voice that has less music than the others. This feature is particularly useful for indicating where a source mistakenly omits one or more notes, or contains a note of shorter value than the one intended, (see below under 4.g, Corrections or Errors in the Source), or for aligning first and second endings of a section of music.

A '#' punched anywhere in the blank space between notes will cause all the information in that note-field to be ignored.

b) Accidentals

To indicate an accidental, one of the following characters is used:

N for natural
S for sharp
T for flat

Any indication of an accidental must precede the information of the staff-position of the note. Editorially suggested accidentals may be indicated by placing the appropriate character in parentheses.

c) Register

The indication of the register to which a note belongs is governed by the following set of conventions:

1. The normal register for the treble-clef, or any clef whose notes are being interpreted as if they were in treble-clef (UNTUCK mode), is defined as extending from the first line (the bottom line) through the fourth line of the staff.

2. Place a U before the pitch-name of a note in the register one octave above the normal register, or preceding note. UU is coded for two octaves above the previous register; UUU, for three octaves, etc. L is placed similarly if the register of the current note is one octave lower than the previous register; LL if two octaves lower, etc.

3. The register of each note is governed by that of the preceding note, unless there are indications (L's or U's) to the contrary. Once the indication U appears before a note, that register now governs all successive

notes in the same voice until an end of section is indicated, or until a change of clef, 'key', or mensuration is encountered (see iv below), or until it is cancelled by an L, as it must be when a change to the next lower register occurs. That is, except at the beginning of a section, where the normal register is assumed, the indications U and L are relative to the register of what has immediately preceded.

4. At the beginning of any section of music, or after any change of clef, 'key', or mensuration in the coded part, the normal register is assumed, irrespective of the register of the immediately preceding note.

5. The program automatically generates the register-change whenever the music moves across a register-boundary <u>by step</u>; there is therefore no need to specify a U or L in this case. The program does this by assuming that no leaps of a seventh will occur between the two notes that define the register-boundaries; if such a leap does occur, the automatic generation of register-changes must be turned off by an *X-card. Register-changes must be provided by the coder whenever the music <u>leaps</u> across a register-boundary.

   Normal registers for clefs which are read according to the TUCK convention--that is, according to their actual pitches--are as follows:

   i) G-clefs: first line to fourth line.

   ii) C-clefs: whatever line represents middle-C to the line representing the first B above. (This has the effect of putting most of the notes in the tenor clef below the normal register.)

   iii) F-clefs: from the space representing the C below middle-C to the space representing the B just below middle-C.

<u>The automatic generation of register changes will presently function for the treble-clef only</u>. The SYNTAX program also assumes that no music will go more than one ledger line above the staff or one ledger line below. if it does, ERROR messages are produced and the automatic generation of register changes becomes unreliable.

The character indicating a register-change must precede the indication of staff-position but may precede or follow any character indicating an accidental.

d) <u>Repeated notes</u>

A number of successive notes of the same pitch and duration, or rests of the same duration, may be indicated by placing a numeral before the note-information to indicate the number of repetitions; thus, if a piece contains a passage of 13 breve rests, only 13R11, or 13RW (see below) need be entered. Any 'additional information' (see under 4 below) will be taken to apply only to the <u>last</u> note of the series. The numeral must precede any information about pitch, register, or accidentals. Any indication of register change will be applied to the first note only; thus '5UF2' is equivalent to 'UF2 4F2'.

2. <u>Duration</u>

Durations are indicated by the <u>numbers</u> shown in the following table of equivalents:

<u>White Mensural Notation</u>	<u>Modern value</u>	<u>Coded as:</u>
Semibreve	whole	1
Minim	half	2
Semiminim	quarter	4
Fusa	eighth	8
Semifusa	sixteenth	16
Breve (imperfect = 2 semibreves)		11 or W
Breve (perfect = 3 semibreves)		11. or W.
Longa (imperfect = 4 semibreves)		111 or L
Longa (perfect = 6 semibreves)		111. or L.
Maxima (imperfect = 8 semibreves)		1111 or M
Maxima (perfect = 12 semibreves)		111. or M.
Rest one space high, perf. or imperf.		W
Rest two spaces high, perf. or imperf.		L
Rest three spaces high		P or M

A dot signifying that the duration of a note is to be extended by half the value indicated by the preceding symbol is represented by a period.

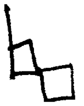
Because constantly pressing the 'numeric' key on the keypunch in order to code durations is bothersome and time-consuming, the translating program has been designed to accept durations expressed by the <u>letters</u> or characters found on the same keys as the numbers; thus, a breve may be coded 'UU'; a minim, 'I', etc.

When coding from mensural notation, the coder must indicate the actual durations of the notes--that is, he or she must interpret notes in ligature or coloration and must perform any necessary alteration or imperfection, with the exception of breve, longa, and three-space *rests* as noted in the following paragraph.

The letters W ("whole"), L, and M may be substituted respectively for breves, longas and maximas, as shown in the above table. *For rests only*, the dots indicating perfection may be omitted when coding in certain mensuration-signatures. For instance, in a piece in *tempus perfectum*, 'RW' is equivalent to 'R 11.'.

3. Additional information

Units of additional information may be entered in any order. They may not be separated from the duration or pitch group by a space. However, if there is more than one unit of additional information, *each must be separated from the preceding by a* '+' (*no spaces*). No unit of additonal information may contain internal spaces.



= B12

a) LIGATURE: a 'B' (Bracket) followed by a numeral indicating the number of notes in a ligature. A 'B' only, without a number, may be used to indicate a two-note ligature. (Also see below under TAG, no. 7).

b) TRIPLETS: a '3' followed by a number indicating the number of notes in the triplet group. The '3' may be separated from the number by a '-' if desired; thus 3-6 or 36 indicates a triplet group of six notes.

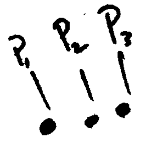
Pitch Dur B

No in bracket if not 2

c) FERMATA: an 'F' only. This may also be used to indicate coronas.

d) TAG: The following table shows the FASTCODE code-characters which will produce a special code. These code-characters are added to the code for a particular note in the same manner as any other unit of additional information; if there is more than one tag, the second one must be preceded by a plus-sign ('+'). An explanation of the coding of each of these features follows the table.

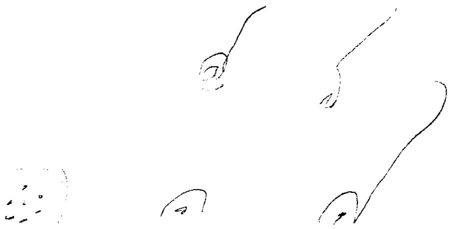
triplets



■ = D2.*

= P43-3 P_2 P_3

FASTCODE Code	Used to indicate
|	line-ending in source
*	colored note
*=	begin colored passage
*|	end colored passage
C	correction in source
X	error in source
&	early placement of accidental
(.)	punctus
@	sign of congruence
(pitch = Z)	[a 'Z' or dummy note]
F	fermata or corona



1. Line-endings.

The note at the end of the line is tagged.

2. Coloration.

The coder must make the necessary calculations of duration; that is, he or she must translate into modern note-values. Thus, a colored semibreve would be coded as 2.*. For long passages in coloration, the first note may be tagged with a '*=' and the last with '*|'.

3. Corrections or Errors in the source.

The following procedure has been adopted:

a) Corrections: TAG the note with a 'C'. at a certain point. One may also provide a non-analyzable comment on a **-card to describe the correction; preface such a comment with 'COR:'. For instance:

```
** COR: G SEMIBREVE CROSSED OUT.
```

b) Errors: after a section of music has been printed out for proofreading, one may discover that the source contains an error in the duration of a note(s), or has omitted a note(s), causing the music to be misaligned. It is best at this point to correct the note(s) to the proper duration and do one or a combination of the following:

1. if a note or notes are omitted in the source, provide a dummy note(s)--that is, a note with 'staff position Z'--of a duration equivalent to the omitted note(s). In the proofreading program, these dummy notes will insert blank spacing and insure the proper alignment of the score.

2. if a note(s) in the source is of a wrong duration, code the note with the correct duration followed by the errant duration in parentheses. Thus, if a dotted minim appears in a source where the music demands a semibreve, the duration of the note should be coded '1(2.)' or 'U(2.)'.

One or a combination of the above expedients will solve most problems of rhythmic errors in a source. A less desirable but sometimes necessary solution is to merely tag any errant note with an 'X'; this will indicate that some kind of error exists in the source at that point. A non-analyzable comment card, prefaced with 'ERR:', may also be provided. However, the research must correct the note values to insure the proper vertical alignment of the music.

4. Early placement of an accidental.

If an accidental is placed several notes before the note to which it applies, tag the note before which the accidental is placed with an '&' and indicate the accidental in the normal manner on the note to which the accidental applies. Both must be indicated.

5. Punctus.

The numbers and dots indicating the duration of a note must represent the proper musical value, not just the physical appearance of the notes. Thus, when coding from sources in white mensural notation, it is often necessary to code a dot for a perfected note even though that dot does not appear in the source. To make it possible to recover the original notation, the following intellectually awkward but time-saving rule has been adopted for any mensuration that demands the perfecting on one or more metric levels:

> On any metric level, notes to which the rules of perfection and imperfection apply, but which are notated with an explicit dot in the source, must be tagged with a '(.)'. Thus, a breve in tempus perfectum is coded as '11.(.)' if the dot is explicitly notated in the source, and as '11.' if the breve is perfect but notated without a dot. A dotted semibreve with an explicit dot in tempus perfectum, minor prolation would not be so tagged because the rules of perfection and imperfection do

not apply at that metric level. Any _punctus divisionis_ is also represented by '(.)'.

6. _Sign of congruence_.

The note over which the sign occurs is tagged.

7. _Ligatures_.

The coder _must_ interpret the proper duration of the notes in the ligature.

8. _Dummy note_.

See above under 2. _Pitch-group_.

9. _Fermata_ (_Corona_).

The note over or under which the symbol occurs is tagged.

2.5 SPECIAL CODES TO INDICATE UPPER- AND LOWER-CASE LETTERING IN LITERALS

NB: This section applies only to code entered on a keypunch. The normal actions taken by the translating program for each type of literal have been noted in the preceding description of the language. To override these actions or to indicate upper- and lower-case lettering elsewhere, the following codes may be used (the code letters themselves will disappear when the conversion is made):

'$' will cause the next letter to be capitalized.

'¢' will cause the next letter to be in lower case.

'%' will cause all succeeding letters to be capitalized until the end of the lettering, a space, or a '¢' is encountered.

'_' indicates that all succeeding text is to be italicized up to an '=', or a space, or the end of the lettering.

'=' will indicate the end of an italicized passage.

'¬' will insert a space in any literal but will not be taken as syntactically significant.

Chapter 3

THE PROOFREADING PROGRAM (WRTFCD)

3.1 GENERAL DESCRIPTION

The proofreading program reads music coded in FASTCODE and converts the code back into musical notation by means of a standard 'X-Y' plotting device. The musical characters plotted by the program are those of white mensural notation. However, because the principal purpose of the program is to produce copy for proofreading, speed and economy have been considered more important than aesthetic appearance or literal adherence to all details--most of which however are explicitly indicated. Other plotting programs may be used if a result more pleasing to the eye and to musicianly taste is desired, although all of the plotting programs use Renaissance note-shpaes.

3.2 HOW TO RUN THE PROGRAM.

Because the program is lengthy, it is best to have a copy of it installed on an easily accessible memory-device, preferably a disk. In order to invoke the program, the user need only supply a few job-control cards with the FASTCODE deck.

The programs that control most plotting-devices require that space for a plot be allocated before plotting is actually begun; therefore, it is normally necessary to include with any FASTCODE deck that is to be plotted a card (Code-Character: '*=') giving an estimate of the length of the plotted music. The figure given on the card must be <u>at least</u> as long as the actual plot. At Princeton, the length is stated in hundreths of inches; the following formula has a built-in safety margin of three inches:

((no. of semibreves per mea.) x (no. of mea.) x 40) + 300

Thus, a piece in ₵ to be transcribed in 60 measures, two semibreves per measure, would require the following card:

*=5100

There are no spacing requirements on the *=-card. It must be placed in the FASTCODE deck anywhere before the first card containing note information. If no *=-card is submitted, a DEFAULT length of one-hundred inches (i.e., '*=10000') will be supplied. The program will also assume that a section of music--any number of sections may be submitted in the same job--will have the same length as the preceding section unless a new *=-card is introduced.

If a new section of music is the same length or shorter than the preceding and if there is room vertically on the plotting paper, the program will plot the new section below the previous section. By carefully grouping together sections of similar length, the user may considerably conserve paper and cut costs.

The support-programs for most plotting-devices will normally terminate a plot if the plot overruns the estimated length, even though the programs that produce the instructions for the plotter may have processed all of the data. In order to avoid a situation in which perhaps the first line of music overruns the length estimate and the costly computing for the remainder of the music is lost, the proofreading program will not advance beyond the estimated length but will merely (and messily) overprint notes until it reaches the end of any line of music that exceeds the length estimate. This allows the music for subsequent lines to be plotted and a preliminary proofreading made before re-submitting the deck with a longer *=-card.

Further care has been taken to insure that the proofreading program will not terminate abnormally because of improper coding. It is wise, however, before reading-in a FASTCODE deck to double-check the L-card and to be sure each voice terminates with '///'. If irregularities in the FASTCODE are encountered by the program, error messages similar to those produced by the translating program are produced in the printed output that normally accompanies the plotted output.

3.3 HOW TO READ THE PLOTTED OUTPUT.

With the exception of a few symbols for certain bibliographical and musical features (see below), the plotted music is self-explanatory to anyone who reads musical notation. The lines of coded music are combined into a score for each section by plotting the music proportionally--that is, the horizontal spacing between notes is directly proportional to their durations; if a note is omitted or coded with an incorrect duration, the music will not be properly aligned. To aid in the correction of durational errors, numbers showing the total number of semibreves in each voice in each section are printed over the final note of each line.

Preliminary bibliographical information is printed above each section of music. Because the plotter will not produce lower-case letters, all lower-case letters (including those in text-syllables and comments) are converted to upper-case; special literal codes (see Chapter 2, Section 5) are printed as coded.

Triplets: a '/3' is printed below the staff the beginning of any triplet passage, and a '3/' at the end.

Tags and special signs.

1. Line-endings.

   A small vertical line through the top line of the staff indicates a line-ending.

2. Coloration.

   The first note of a passage of colored notes is marked with an open-bracket above the staff; the last with a close-bracket. Single colored notes are marked by 'COL.' written above the staff.

3. Corrections in the source.

   'COR.' is written above any note tagged as having been corrected.

   Errors in the source.

   a) If coded with staff-position 'Z', a blank space proportional to the duration desired will be be left and a downward arrow will be plotted above the staff at the point of the omission in the source.

   b) If the erroneous note value was coded in parentheses, the erroneous note-shape will be printed but the horizontal spacing for the note will be according to the correct duration.

   c) Any note tagged with an 'X' will have "ERR." written over it.

4. Early placement of an accidental.

   The accidental will be positioned as it is positioned in the source.

5. Punctus.

   Any coded in parentheses indicated by '(.)' above the staff.

6. Sign of congruence.

   Sign of congruence appears above the note coded.

7. Ligatures.

A horizontal line below the staff connects notes in ligature.

8. Dummy note.

A downward arrow marks the location.

3.4 A BRIEF TECHNICAL DESCRIPTION

The program normally requires about 300K 8-bit bytes of core (16-bit integers are sufficient) to operate and is best used in a batch environment. The program consists of three modules:

1. MAIN.

Written in SNOBCL but best run under the more efficient SPITBOL compiler, this part of the program parses the FASTCODE, calculates the positioning of the different musical symbols, and passes these calculations to SUBROUTINE IN. Normally, it should not be necessary to make modifications in MAIN when adapting the program for use at other computer installations.

2. SUBROUTINE ISTR.

Written in IBM Assembler (but easily translated to other assemblers), this very short routine calculates the number of characters in a literal string. ISTR is called exclusively in statement numbered 3 of SUBROUTINE IN.

3. SUBROUTINE IN.

Written in FORTRAN, this lengthy but structurally simple subroutine controls the movements of the plotter. The subroutine is subdivided into thirty-two independent sections, each of which performs a specific function, such as drawing a particular musical symbol, or repositioning the pen.

Subroutine IN calls four other subroutines (NGRAPH, PLOT, SYMBOL, and ENDPLOT) that are part of the support-programs for the plotter at Princeton. If the proofreading-program is to be used elsewhere, the calls to these subroutines will probably have to be modified.

This should be a simple operation: either intermediate subroutines (named NGRAPH, PLOT, SYMBOL, and ENDPLOT) can be written to issue proper calls to similar subroutines available at a specific computer-installation, or the calls within subroutine IN itself can be modified. Because there are many calls within subroutine IN to these four subroutines, the latter option is advised only if the modifications can be made automatically, by either a small program or an editing system.

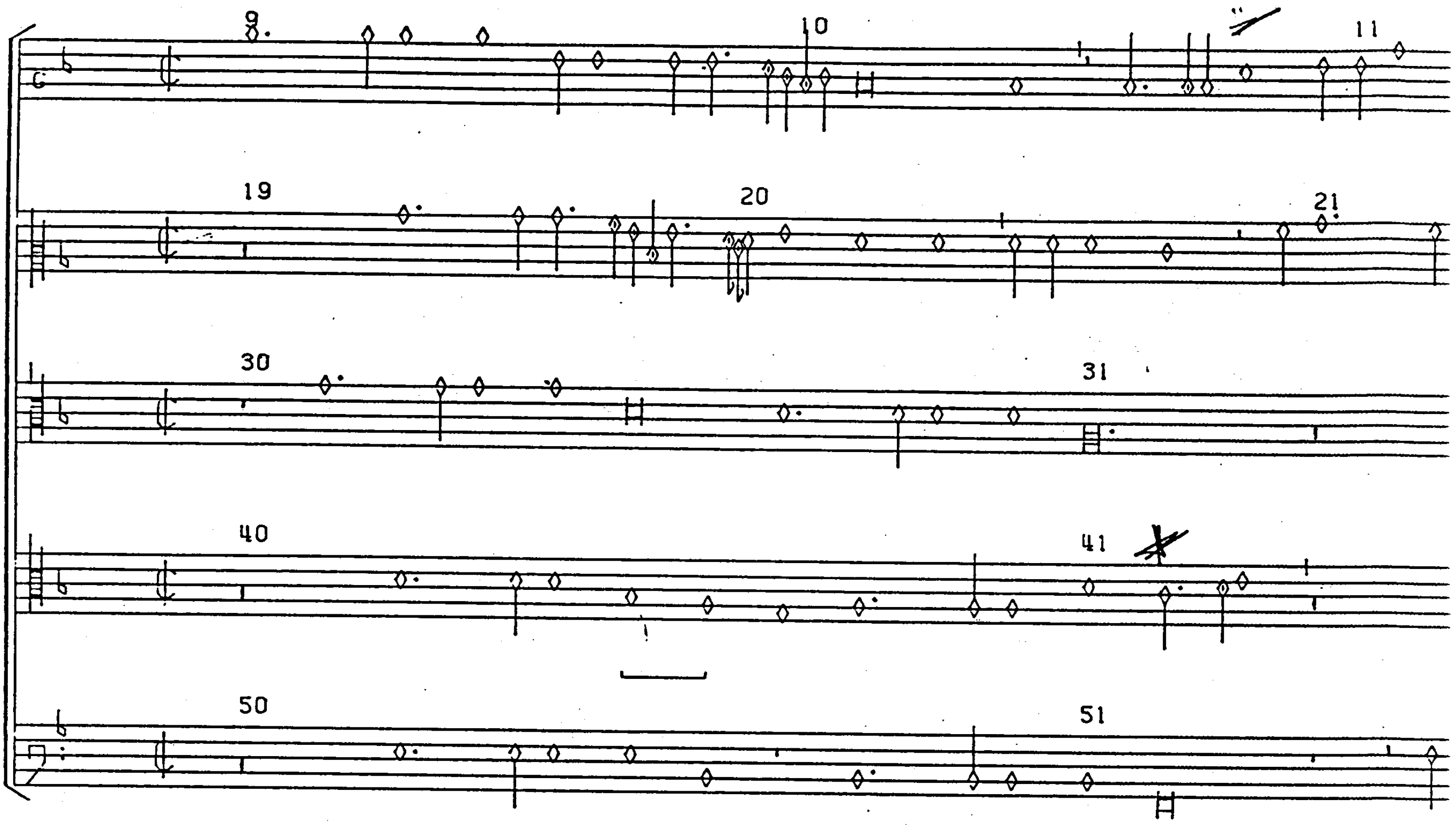
Appendix A

THE CODING OF SOME MENSURATION AND PROPORTION SIGNS

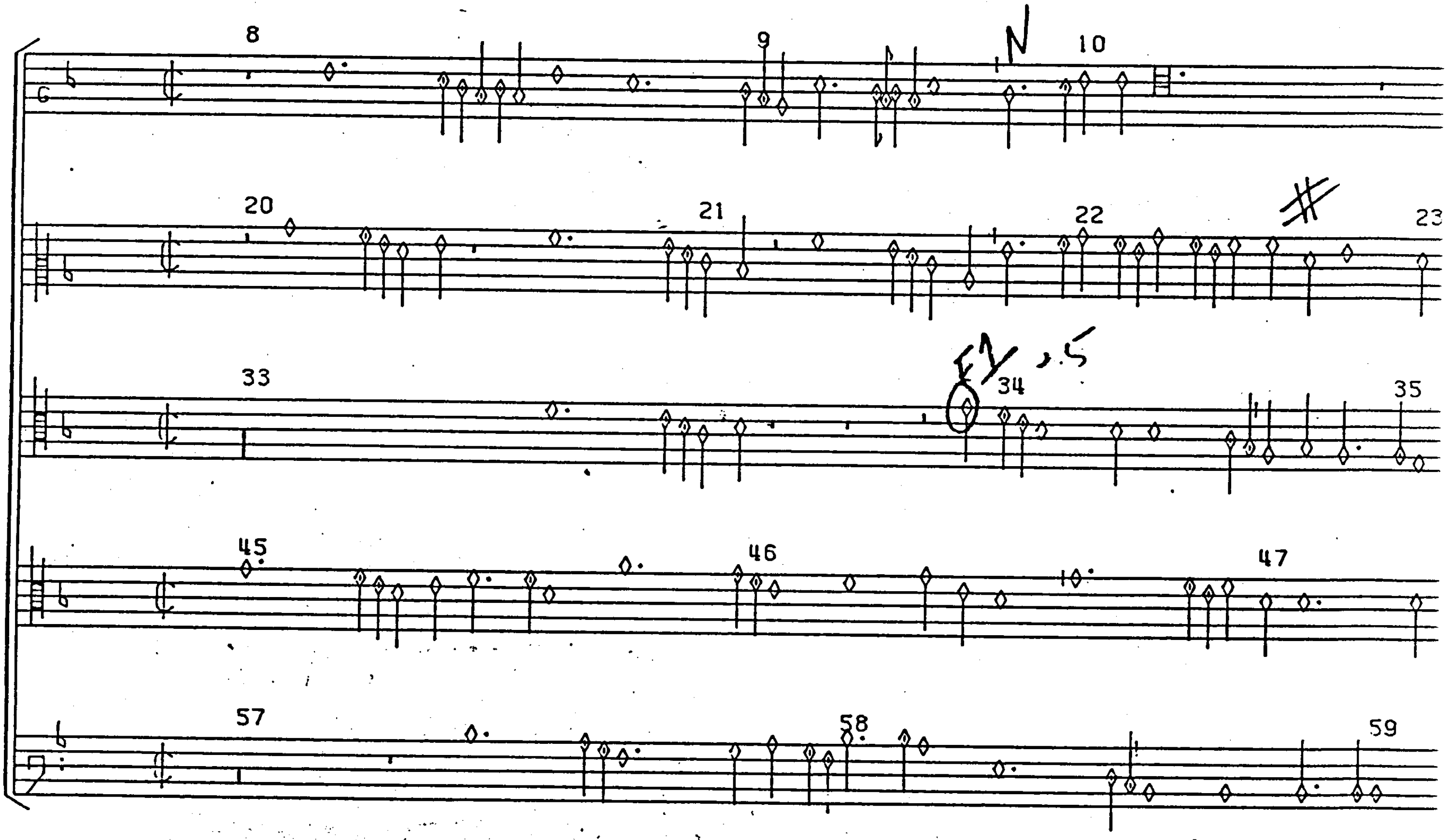
Sign	Conventionally coded as	DEFAULT Retime value
0	0	3/1
Ø	0/	3/1
0 3	0 |3	3/1
3	3	3/1
3 2	3 |2	3/1
₵3	C/3	3/1
02	02	6/1
C	C	2/1
₵	C/	2/1
Ċ	C.	6/2
C2	C2	2/1
⊙	0.	none
)	>	none
)̇	.>	none
₵̇	C/.	none

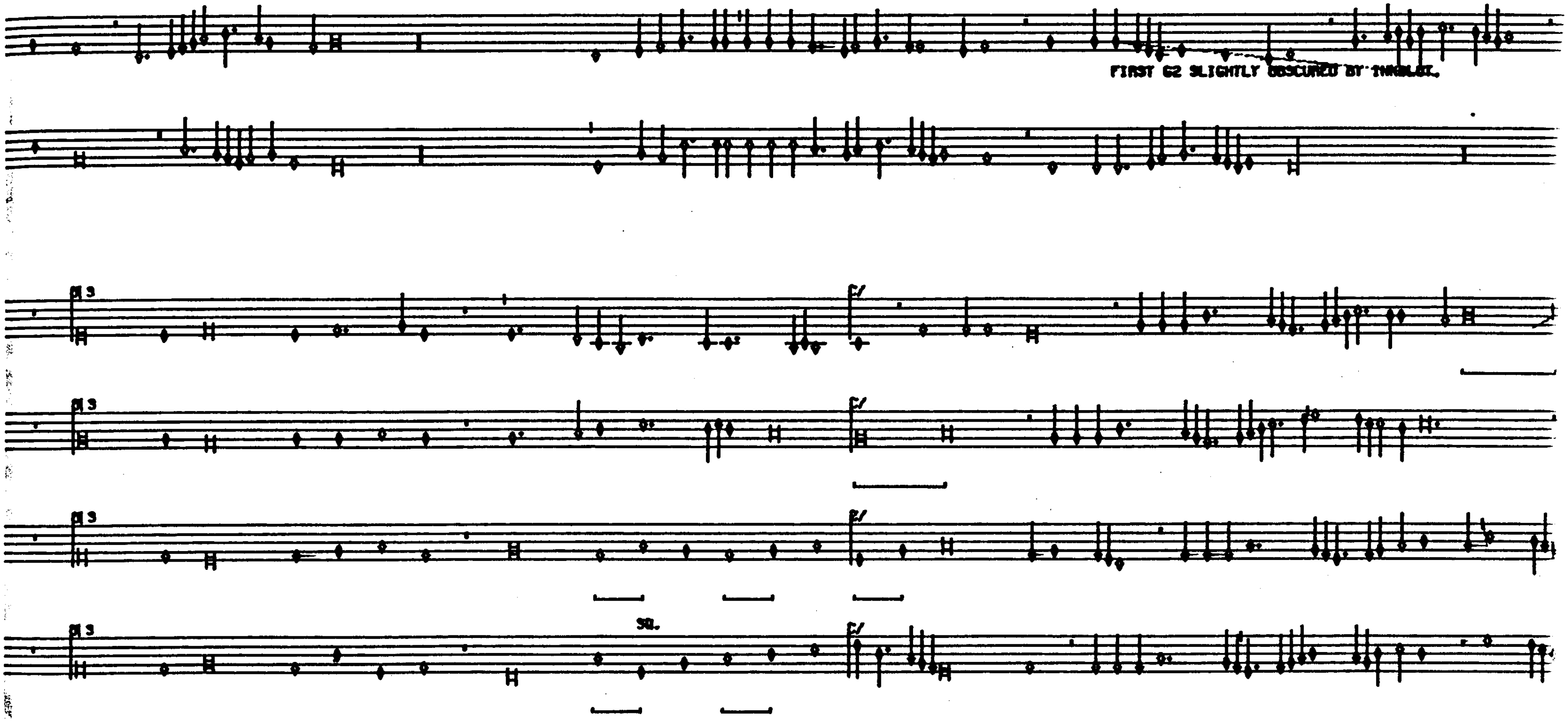
Note: for the sign), a close parenthesis, while more graphically representative, is unacceptable.

PØPULE MEUS 1582D 1582D 10M1 JENSEN

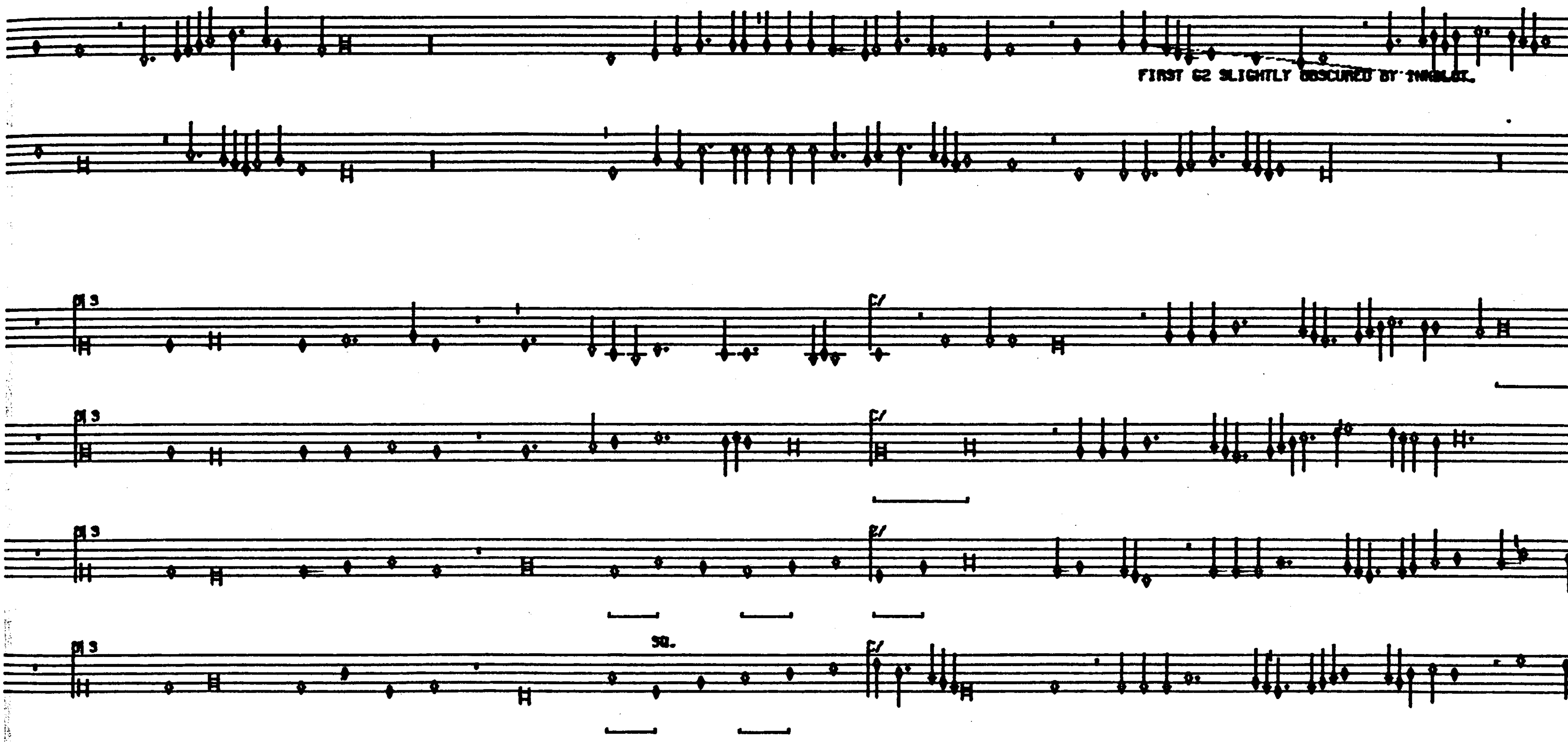


SECUNDA PARS NUMQUID REDDITUR 1582D 1582D 10M2



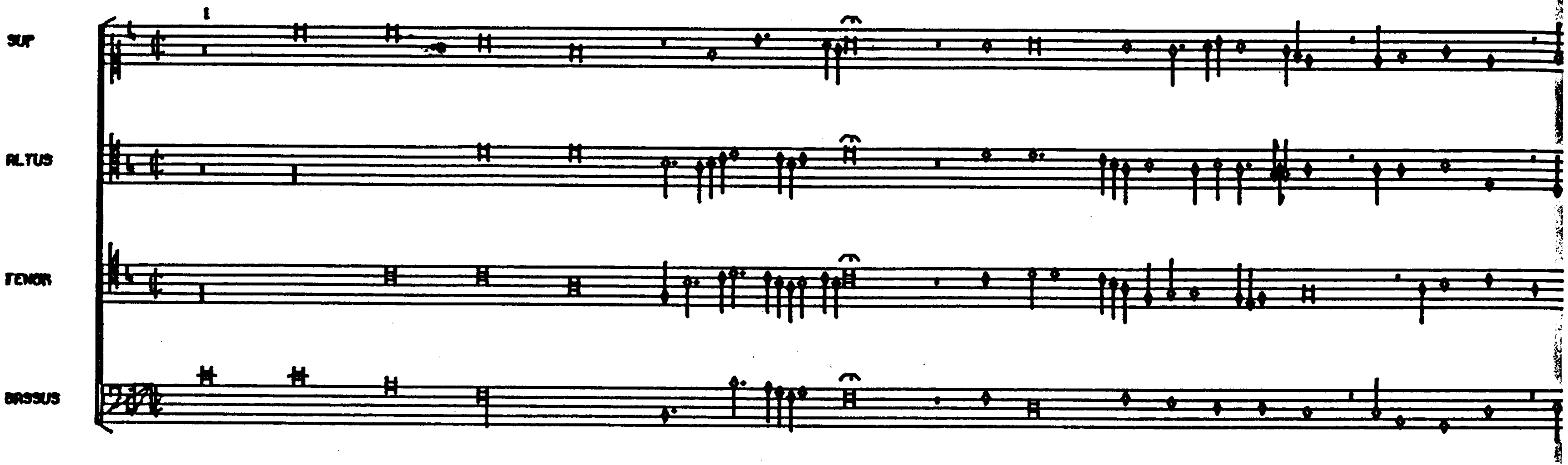


FIRST G2 SLIGHTLY OBSCURED BY INKBLOT.

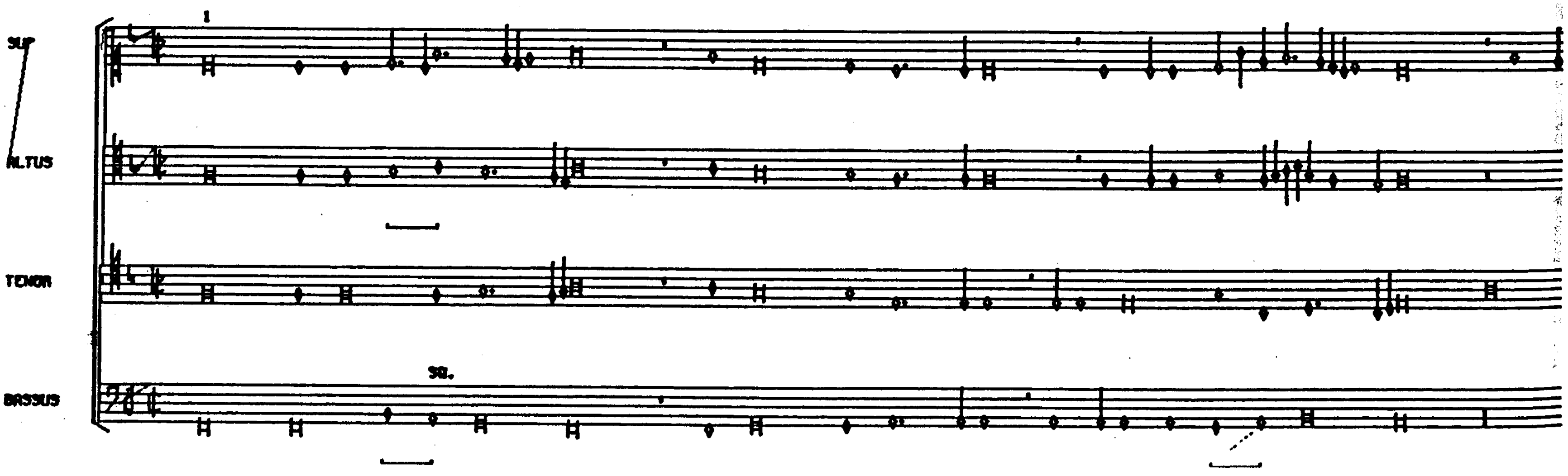


FIRST G2 SLIGHTLY OBSCURED BY INKBLOT.

(JOSQUIN) (PLANXIT AUTEM DAVID) (RRIMA RARS) 1504/1 RARTBOOKS RASMUSSEN

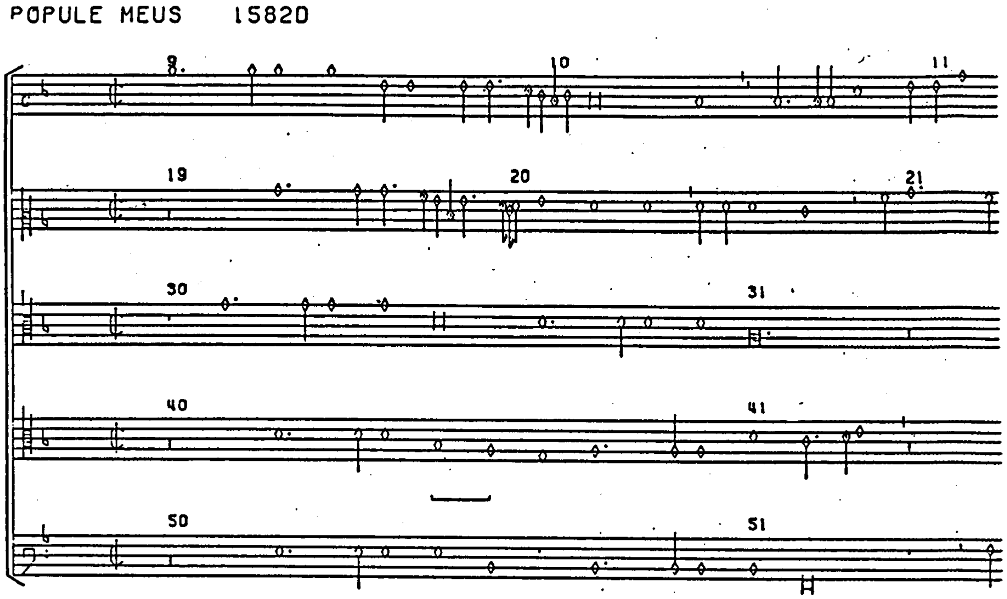


(SECUNDA RARS) 1504/1

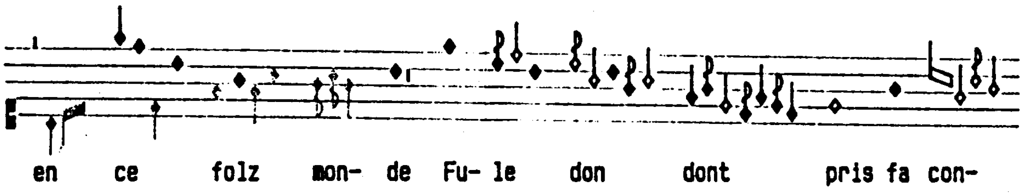


Systems for Printing Mensural Notation

Many systems for meeting the special printing needs of early repertories have been developed. In addition to *SCORE* and *Finale*, which both provide capabilities in this area, the following examples come from systems are among those displayed or discussed in previous years.



White mensural notation in score produced by Tom Hall's FASTCODE, c.1981.



Red [here grey] and black mensural notation produced by SCRIBE c.1988.