DEPARTMENT OF THE INTERIOR

U.S. GEOLOGICAL SURVEY

Source codes of CLAYFORM and associated file of data constants,
a FORTRAN 77 computer program for calculating structural
formulae of clay minerals and other silicates

by

Marc W. Bodine, Jr.¹

Open-File Report 87-0004-A

DISCLAIMER

This program was written in FORTRAN 77 for the PRIME computer. Although program tests have been made, no guarantee (expressed or implied) is made by the author regarding program correctness, accuracy, or proper execution on all computer systems.

1987

¹U.S. Geological Survey
MS 939, Box 25046, Denver Federal Center
Denver, CO 80225

* Any use of trade names in this report is for descriptive purposes only and does not imply endorsement by the U.S. Geological Survey. This report is preliminary and has not been reviewed for conformity with U.S. Geological Survey editorial standards.

TABLE OF CONTENTS

Abstract.....	1
Introduction.....	1
CLAYFORM source code.....	2
Program CLAYFORM.....	2
Documentation.....	2
Specification statements.....	3
Input identification and review.....	6
Entering mineral analysis data.....	10
Selecting structural formula specifications.....	14
Structural formula calculation.....	26
Formatting results of calculation.....	32
Printing out sample analysis and structural formula.....	39
Program continuation or termination.....	40
Format statements.....	41
Subroutine YESNO.....	47
Subroutine INSERT.....	47
Subroutine LISTCLAYS.....	48
Subroutines ARRAY and DGELG.....	50
FORMDAT source code, file of data constants for CLAYFORM.....	52
References.....	53

SOURCE CODES OF CLAYFORM AND ASSOCIATED FILE OF DATA CONSTANTS,
A FORTRAN 77 COMPUTER PROGRAM FOR CALCULATING STRUCTURAL
FORMULAE OF CLAY MINERALS AND OTHER SILICATES

by

Marc W. Bodine, Jr.

U.S. Geological Survey

MS 939, Box 25046, Denver Federal Center

Denver, Colorado 80225

ABSTRACT

The computer program CLAYFORM with its associated file of data constants calculates the structural formula of clay minerals and other silicate minerals from their chemical analyses. This report contains the source code for the program and a listing of the associated data file, FORMDAT.

INTRODUCTION

The FORTRAN 77 computer program CLAYFORM and its associated data file of constants, FORMDAT, were written to carry out rapid calculations of structural formulae of clay minerals and other silicate minerals from their conventional (weight percentages of constituent oxides) chemical analyses. Program rationale, use, and user options have been given in Bodine (1987) and only the program source code and a listing of FORMDAT is presented here. ARRAY and DGELG, two short subroutines in the listing, are published IBM codes (IBM Application Program, 1970) for solving linear simultaneous equations.

The program is operating on the U.S. Geological Survey's PRIME system at the Denver Federal Center and the ASCII source code is available on an IBM PC diskette (Open-File Report 87-0004-B) from Open-File Services, U.S. Geological Survey, Mail Stop 517, Box 25046, Denver, CO 80225.

SOURCE CODES

Program CLAYFORM

Documentation

```

c ***** 0001
c                                     CLAYFORM 0002
c ***** 0003
c                                     May 1986 0004
c 0005
c CLAYFORM, an interactive FORTRAN 77 program, calculates and displays 0006
c the structural formula of a clay or other silicate mineral from its 0007
c conventional (weight percentages of constituent oxides) chemical 0008
c analysis. The program also creates and reads data files of silicate 0009
c mineral analyses. The structural formula may be expressed as (1) a 0010
c general silicate mineral formula with cation sites undesignated, (2) 0011
c a general silicate mineral formula with cation sites designated but 0012
c unlabeled, or (3) a clay-mineral formula with cation sites desig- 0013
c nated and labeled (tetrahedral, octahedral, interlayer) and charge 0014
c distribution among the layers calculated. The user enters the type 0015
c of structural formula and, if a clay-mineral type is specified, de- 0016
c fault specifications are assigned that can be modified by the user. 0017
c If a general silicate mineral formula is specified the user enters 0018
c the specifications of the structural formula. These include the 0019
c parameters of the ideal stoichiometry that are to define the struc- 0020
c tural formula and a variety of properties (cation site assignments, 0021
c components to be deleted from the structural formula, redox state of 0022
c any iron in the analysis, character of the analyzed water as OH, H2O, 0023
c H3O+, or omitted from the structural formula). The default specifi- 0024
c cations for the ten structural types of clay minerals can be similar- 0025
c ly modified. 0026
c 0027
c References: 0028
c 0029
c Bodine, M. W., Jr., 1987, CLAYFORM, a FORTRAN 77 computer program 0030
c apportioning the constituents in the chemical analysis of a 0031
c clay or other silicate mineral into a structural formula: 0032
c Computers & Geosciences, v. 13, p. 77-88. 0033
c 0034
c A discussion of the rationale and use of CLAYFORM. 0035
c 0036
c Bodine, M. W., Jr., 1987, Source codes of CLAYFORM and associated 0037
c file of data constants, a FORTRAN 77 program for calculating 0038
c structural formulae of clay minerals and other silicates: 0039
c U.S. Geological Survey Open-File Report 87-0007, 53 p. 0040
c 0041
c A listing (87-0007-A) and IBM-PC diskette (87-0007-B) of 0042
c CLAYFORM and FORMDAT (data constants) ASCII source codes. 0043
c 0044
c ***** 0044

```

Specification statements

character*1 ly,ln,la,lr,id,ll,al	0046
character*2 lid,lib,li	0047
character*4 none,lc,jc,a4,a4b	0048
character*5 lox,kfile,a5,a5b	0049
character*8 lcap,lfile,inpt,lform,a8,llst,a8b	0050
character*16 lcat,lbl,lbig,lll	0051
character*20 lraa,lras	0052
character*22 ltaa,ltas	0053
character*25 lt7	0054
character*27 lcz	0055
character*28 ha,habl,hff	0056
character*29 lt4,lcaa,lcas	0057
character*30 lt8	0058
character*31 lqa,lqb	0059
character*32 ltl1	0060
character*33 lsit	0061
character*34 ltl0,lwat,lctn	0062
character*35 hfa,hfp,lt2	0063
character*36 lt6	0064
character*37 lt5	0065
character*38 ltoc	0066
character*39 lneg	0067
character*40 lbt,ltl	0068
character*41 llt	0069
character*47 lt3	0070
character*49 lt9	0071
character*51 hf,hfbl	0072
character*75 ltit,175	0073
character*79 179,a79,a79b	0074
character*80 ldes,180	0075
logical*4 mol	0076
double precision adp,xdp	0077
dimension adp(3),xdp(9)	0078
dimension wm(50),kc(50),nc(50),kn(50),nn(50),io(50),ky(50),ny(50),	0079
a ndto(10),nd(10),nh(10),pw(50),xm(50),pm(50),xc(50),c(50),xa(50),	0080
a jtet(10),cz(3),nt(7),ntt(10),no(20),xnn(50),xnc(50),ndel(50),	0081
a mt(20),mo(20),mi(20),js(10),ne(6),qe(6),stot(5),a(50),nz(50),	0082
a ktt(7),na(10),ta(3),tx(3,3),ty(9),loo(50),iww(3),xmm(50),	0083
a nw(10),ntto(10),	0084
s lr(20),lox(50),lc(50),jc(50),lbt(10),lcap(4),llt(10),lcat(5),	0085
s lcz(3),li(50),ha(99),hf(99),lcaa(5),lwat(3),a8(8),	0086
s id(5),al(8),a4(30),lraa(2),ltaa(4),llst(3,30),a79(20),a5(30)	0087
data lbt/'1:1 layer clay (kaolinite).	0088
-'2:1 layer clay (mica).	0089
-'2:1 + 60H layer clay (chlorite).	0090
-'Mica-kaolinite 1:1 mixed-layer.	0091
-'Mica-mica mixed-layer.	0092
-'Chlorite-kaolinite 1:1 mixed-layer.	0093
-'Chlorite-mica 1:1 mixed-layer.	0094
-'2:1 ribbon clay (palygorskite).	0095
-'2:1 ribbon clay (sepiolite-8 oct.).	0096

- '2:1 ribbon clay (sepiolite-9 oct.).	0097
data lcap/'weight %', mole % ',-----', Total	0098
data lraa/' with changes.	0099
data ltaa/'Tetr. filling order: ', To octahedral sites: ',	0100
- 'To interlayer sites: ', To big cation sites: '/	0101
data llt/' 1:1 layer clay (kaolinite type)	0102
- ' 2:1 layer clay (mica type)	0103
- ' 2:1 + 60H layer clay (chlorite type)	0104
- ' Mixed-layer clay (1:1 mica-kaolinite)	0105
- ' Mixed-layer clay (mica-mica)	0106
- 'Mixed-layer clay (1:1 chlorite-kaolinite)	0107
- ' Mixed-layer clay (1:1 chlorite-mica)	0108
- ' 2:1 double chain (palygorskite)	0109
- '2:1 triple chain (sepiolite) 8 oct. sites'	0110
- '2:1 triple chain (sepiolite) 9 oct. sites'/	0111
data lwat/'as OH anion in structural formula.'	0112
- 'as H2O molecular water.	0113
- 'omitted from structural formula. '/	0114
data lcz/'Tetrahedral layer charge = '	0115
- 'Octahedral layer charge = '	0116
- 'Interlayer cation charge = '/	0117
data ltl/'Character of water in structural formula'/	0118
data lt2/'Iron redox in structural formula: '/	0119
data lt3/'Component(s) deleted from structural formula: '/	0120
data lt4/'Type of structural formula: '/	0121
data lt5/'Oxygen equivalents per formula unit: '/	0122
data lt6/'Tetrahedral sites per formula unit: '/	0123
data lt7/'Cation site assignments: '/	0124
data lt8/'Anion sites per formula unit: '/	0125
data lt9/'Tetrahedral-octahedral cations per formula unit: '/	0126
data lt10/'(OH,F,Cl) sites per formula unit: '/	0127
data lt11/'Total cations per formula unit: '/	0128
data lcaa/'Standard clay-mineral default',	0129
- 'Chlorite clay-mineral default',	0130
- '"Ribbon" clay-mineral default',	0131
- 'Reassign selected cations	0132
- 'As previously user-modified	0133
data hff/'Set to Fe2+/(Fe2+ + Fe3+) = '/	0134
data hfa/'FeO and Fe2O3 values in analysis. '/	0135
data a8b/'	0136
data a5b/'	0137
data a4b/'	0138
data none/'None'/	0139
data lb1/'	0140
data lib/'	0141
data nd/7,11,14,18,22,21,25,21,32,33/	0142
data na/9,12,18,21,24,27,30,26,39,39/	0143
data jtet/2,4,4,6,8,6,8,8,12,12/	0144
data nt/1,3,0,0,0,0,0/	0145
data ntt/1,22,30,3,16,4,2,0,0,0/	0146
data nw/4,2,8,6,4,12,10,2,4,6/	0147
data ndto/4,6,8,10,12,12,14,13,20,21/	0148
data ntto/5,7,10,12,14,15,17,13,20,21/	0149

```

data ly/'y'/                                0150
data ln/'n'/                                0151
data lbig/'Large cations  '/                0152
data lqa/'Negative charge to be balanced?'/ 0153
data lqb/'Type of clay structural formula'/ 0154
data lneg/'Oxygen equivalents per formula unit = '/ 0155
data lsit/'Total anions per formula unit = '/ 0156
data ltoc/'Tet + oct cations per formula unit = '/ 0157
data lctn/'Total cations per formula unit = '/ 0158
data kfile/'CLAY.'/                          0159
data id/ '&', '$', '=', '*', ' ' /          0160
data inpt/' UNFILED'/                        0161
data io0/1,2,14,30,3,15,16,4,5,6,17,        0162
-18,19,20,21,22,7,8,23,24,25,              0163
-9,10,26,27,28,29,11,12,13,20*0/           0164
c                                             0165
c*****                                     0166
c               READING IN DATA FILE "FORMDAT" 0167
c*****                                     0168
c                                             0169
write(hab1,9232)                             0170
write(hfb1,9233)                             0171
nwat=1                                       0172
xwat=1.0                                    0173
la=ln                                       0174
jntt=7                                     0175
nq = 0                                     0176
nl=0                                       0177
kal=0                                      0178
lr(8)=ln                                   0179
nox = 0                                    0180
ntoc=0                                    0181
itoc=0                                    0182
xtoc=0.0                                  0183
nofile=0                                  0184
nono=2                                    0185
nbeg=1                                    0186
notet=0                                   0187
nfic=0                                    0188
nfus=0                                    0189
open(l1,status='old',file='formdat')        0190
c                                             0191
Molecular components                        0191
do 12 i=1,50                               0192
if(nq.ne.0) go to 10                        0193
read(l1,881)lox(i),jc(i),wm(i),kc(i),kn(i),ky(i) 0194
if(wm(i).eq.0.0) go to 10                  0195
nox=nox+1                                  0196
go to 11                                    0197
10 nq=1                                     0198
lox(i)=1b1                                 0199
jc(i)=1b1                                  0200
wm(i)=0.0                                  0201
kc(i)=0                                    0202

```

kn(i)=0	0203
ky(i)=0	0204
11 continue	0205
lc(i)=jc(i)	0206
nc(i)=kc(i)	0207
nn(i)=kn(i)	0208
ny(i)=ky(i)	0209
io(i)=ioo(i)	0210
xnc(i)=float(nc(i))	0211
xnn(i)=float(nn(i))	0212
12 continue	0213
nos=nox	0214
nax=nox	0215
lr(8)=ln	0216
close(11)	0217
write(180,891)	0218
read(180,892) (lcat(i),i=1,5)	0219
hdel=0	0220
nl=100	0221
xiron=2.0	0222
nll=0	0223
ndbl=1	0224
do 80 i=1,50	0225
if(i.lt.11) nh(i)=nd(i)	0226
if(i.lt.21) lr(i)=ln	0227
if(i.lt.4) iww(i)=0	0228
ndel(i)=0	0229
nz(i)=0	0230
80 continue	0231
kloi=0	0232
iwat=0	0233
nform=0	0234
kwat=0	0235
kkwat=0	0236
ncats=0	0237
nzz=0	0238
itet=0	0239
go to 110	0240
c	0241

Input identification and review

c*****	0242
c ANALYSIS FILE IDENTIFICATION, OPENING, AND REVIEWING	0243
c*****	0244
c	0245
100 continue	0246
do 108 i=1,10	0247
ii=i	0248
103 continue	0249
if(ntt(i).eq.0) go to 105	0250
if(ntt(i).le.nax) go to 104	0251
ii=ii+1	0252

if(ii.gt.10) ntt(i)=0	0253
if(ntt(i).eq.0) go to 105	0254
ntt(i)=ntt(ii)	0255
go to 103	0256
104 jntt=i+1	0257
105 ii=i	0258
if(i.gt.7) go to 108	0259
106 continue	0260
if(nt(i).eq.0) go to 108	0261
if(nt(i).le.nax) go to 108	0262
ii=ii+1	0263
if(ii.gt.7) nt(i)=0	0264
if(nt(i).eq.0) go to 108	0265
nt(i)=nt(ii)	0266
go to 106	0267
108 continue	0268
110 continue	0269
nono=2	0270
do 112 i=11,20	0271
112 lr(i)=ln	0272
nal=0	0273
114 write(1,851)	0274
call yesno(lr(12),nerr)	0275
if(nerr.ne.0) go to 114	0276
if(lr(12).eq.ly) go to 140	0277
115 write(1,852)	0278
call yesno(lr(11),nerr)	0279
if(nerr.ne.0) go to 115	0280
if(lr(11).eq.ln) go to 117	0281
c	0282
c Identifying a new data file to be created	0283
c -----	0284
c	0285
116 write(1,8531)	0286
call yesno(lr(14),nerr)	0287
if(nerr.ne.0) go to 116	0288
do 125 i=1,999	0289
write(lfile,854) i	0290
inquire(file=lfile,exist=mol)	0291
if(mol) go to 125	0292
open(12,status='new',file=lfile,err=125)	0293
write(1,857) lfile	0294
117 write(1,855)	0295
read(1,999,err=117) ltit	0296
write(1,999) ltit	0297
write(1,994)	0298
call yesno(la,nerr)	0299
if(nerr.ne.0) go to 117	0300
if(la.eq.ln) go to 117	0301
if(lr(11).eq.ln) go to 200	0302
write(12,999) ltit	0303
open(14,file='CLAY.ID',status='old',err=200)	0304
read(14,998,end=123) l80	0305

do 121 j=1,999	0306
read(14,853,end=123) jj,175	0307
if(jj.eq.i) go to 122	0308
121 continue	0309
122 continue	0310
backspace(14)	0311
123 continue	0312
write(14,853) i,1tit	0313
close(14)	0314
go to 200	0315
125 continue	0316
c	0317
c Naming and opening an existing data file	0318
c -----	0319
c	0320
140 continue	0321
write(1,858)	0322
read(1,9921,err=140) nq	0323
if(nq.ne.0) go to 141	0324
call listclays	0325
go to 140	0326
141 write(1file,854) nq	0327
if(nox.eq.nax) go to 145	0328
n=nax+1	0329
do 142 i=1,50	0330
io(i)=ioo(i)	0331
if(i.lt.n) go to 142	0332
xnn(i)=0.0	0333
xnc(i)=0.0	0334
lox(i)=1b1	0335
jc(i)=1b1	0336
kc(i)=0	0337
kn(i)=0	0338
ky(i)=0	0339
wm(i)=0.0	0340
142 continue	0341
nox=nax	0342
145 continue	0343
open(13,file=1file,status='old',err=146)	0344
go to 150	0345
146 write(1,859) 1file	0346
call yesno(1a,nerr)	0347
if(nerr.ne.0) go to 146	0348
close(13)	0349
if(1a.eq.1y) go to 140	0350
go to 799	0351
150 write(1,861)	0352
call yesno(1a,nerr)	0353
if(nerr.ne.0) go to 150	0354
nal = 0	0355
if(1a.eq.1y) write(1,869) 1file	0356
read(13,850,err=146,end=146) 11,179	0357
c	0358

c	Scanning existing data file	0359
c	-----	0360
c		0361
	do 163 i=1,1000	0362
	read(13,850,err=762,end=164) 11,179	0363
	if(11.eq.id(1).or.11.eq.id(3)) go to 163	0364
	if(11.eq.id(2).and.1r(11).eq.1y) go to 163	0365
	write(180,850) 11,179	0366
	if(11.ne.id(2)) go to 162	0367
c	Reading file for non-bank components	0368
	nox=nox+1	0369
	n=nox	0370
	read(180,873,err=762)11,1ox(n),jc(n),wm(n),kc(n),kn(n),	0371
	-ky(n)	0372
	io(n)=n	0373
	lc(n)=jc(n)	0374
	nc(n)=kc(n)	0375
	mn(n)=kn(n)	0376
	ny(n)=ky(n)	0377
	xnc(n)=float(nc(n))	0378
	xnn(n)=float(mn(n))	0379
	call insert(ny,nc,mn,io,n)	0380
	if(ny(n).ne.0) go to 161	0381
	ny(n)=1	0382
	ntt(jntt)=n	0383
	jntt=jntt+1	0384
161	continue	0385
	go to 163	0386
c	Listing samples	0387
162	nal=nal+1	0388
	if(1a.eq.1y) write(1,863) nal,180	0389
163	continue	0390
164	continue	0391
	if(1a.eq.1y) write(1,997)	0392
	rewind(13)	0393
	nal=0	0394
c	Designating specific samples in file for calculation	0395
165	write(1,864)	0396
	call yesno(1r(13),nerr)	0397
	if(nerr.ne.0) go to 165	0398
	if(1r(13).eq.1n) go to 180	0399
166	write(1,865)	0400
	read(1,866,err=166) (js(i),i=1,10)	0401
180	continue	0402
c		0403
c	Reading a single analysis from existing data file	0404
c	-----	0405
c		0406
	do 181 i=1,50	0407
181	pw(i)=0.0	0408
	if(nal.eq.0) read(13,999)1tit	0409
	read(13,850,end=762) 11,179	0410
	if(11.eq.id(2)) go to 762	0411

if(11.eq.id(1).or.11.eq.id(3)) go to 761	0412
write(180,850) 11,179	0413
ldes=180	0414
nal=nal+1	0415
do 184 i=1,20	0416
read(13,850,end=761) 11,179	0417
write(180,850) 11,179	0418
if(11.ne.id(1)) go to 183	0419
read(180,872,err=761) 11,(ne(k),qe(k),k=1,6)	0420
do 182 k=1,6	0421
if(ne(k).gt.0) pw(ne(k))=qe(k)	0422
182 continue	0423
go to 184	0424
183 if(11.ne.id(3)) go to 761	0425
read(180,874) 11,1r(7)	0426
if(1r(7).ne.ly.and.1r(7).ne.ln) go to 761	0427
go to 185	0428
184 continue	0429
185 continue	0430
if(1r(13).eq.ln) go to 200	0431
do 186 i=1,10	0432
if(nal.eq.js(i)) go to 200	0433
if(js(i).eq.0) go to 180	0434
186 continue	0435
go to 180	0436
200 continue	0437
c	0438

Entering mineral analysis data

c*****	0439
c ENTERING MINERAL ANALYSIS DATA	0440
c*****	0441
c	0442
1r(15)=ln	0443
if(nono.eq.0) go to 210	0444
if(1r(14).eq.ly) go to 210	0445
205 write(1,9871)	0446
call yesno(1r(15),nerr)	0447
if(nerr.ne.0) go to 205	0448
210 continue	0449
c	0450
	Entering major oxides
if(1r(12).eq.ly) go to 280	0451
write(1,901)	0452
do 212 i=1,50	0453
pw(i)=0.0	0454
if(i.gt.13) go to 212	0455
211 write(1,902) lox(i)	0456
read(1,903,err=211) pw(i)	0457
212 continue	0458
c	0459
	Entering minor oxides
213 write(1,904)	0460
call yesno(1a,nerr)	0461

if(nerr.ne.0) go to 213	0462
if(la.eq.ln) go to 231	0463
write(1,905)	0464
nl=18	0465
do 215 n3=1,10	0466
if(nl.ge.nox) go to 216	0467
215 nl=nl+5	0468
216 nl=14	0469
do 217 i=1,n3	0470
n2=nl+(4*n3)	0471
write(1,906) (k,lox(k),k=nl,n2,n3)	0472
217 nl=nl+1	0473
218 write(1,907)	0474
read(1,908,err=218) n,q	0475
if(n.eq.0) go to 220	0476
pw(n)=q	0477
go to 218	0478
220 continue	0479
c	Entering unlisted components 0480
221 write(1,909)	0481
call yesno(la,nerr)	0482
if(nerr.ne.0) go to 221	0483
if(la.eq.ln) go to 231	0484
222 write(1,911)	0485
n=nox+1	0486
read(1,912,err=122) lox(n)	0487
223 write(1,913)	0488
read(1,914,err=223) jc(n)	0489
lc(n)=jc(n)	0490
224 write(1,915)	0491
read(1,903,err=224) wm(n)	0492
if(wm(n).eq.0.0) go to 231	0493
225 write(1,916)	0494
read(1,991,err=225) kc(n)	0495
nc(n)=kc(n)	0496
226 write(1,917)	0497
read(1,991,err=226) kn(n)	0498
nn(n)=kn(n)	0499
227 write(1,918)	0500
read(1,991,err=227) ky(n)	0501
ny(n)=ky(n)	0502
nq=ky(n)+1	0503
write(1,926) lox(n),jc(n),wm(n),kc(n),kn(n),lcat(nq)	0504
228 write(1,994)	0505
call yesno(la,nerr)	0506
if(nerr.ne.0) go to 228	0507
if(la.eq.ln) go to 222	0508
nox=n	0509
if(ny(n).ne.0) go to 229	0510
ny(n)=1	0511
ntt(jntt)=n	0512
jntt=jntt+1	0513
229 continue	0514

	io(n)=nox	0515
	xnc(n)=float(nc(n))	0516
	xnn(n)=float(nn(n))	0517
	call insert(ny,nc,nn,io,n)	0518
	write(1,901)	0519
230	write(1,902) lox(nox)	0520
	read(1,903,err=230) pw(nox)	0521
	go to 221	0522
231	continue	0523
c	Status of iron analytical data	0524
	lr(7)=ln	0525
	n=0	0526
	if(pw(4).gt.0.0) n=n+1	0527
	if(pw(5).gt.0.0) n=n+1	0528
	if(n.ne.1) go to 233	0529
232	write(1,919)	0530
	call yesno(lr(7),nerr)	0531
	if(nerr.ne.0) go to 232	0532
233	continue	0533
c	Entering sample description	0534
234	write(1,920)	0535
	read(1,998,err=234) ldes	0536
235	continue	0537
	nofile=0	0538
c		0539
c	Confirming analysis entry	0540
c	-----	0541
c		0542
238	write(1,921)	0543
	do 240 i=1,nox	0544
	if(pw(i).eq.0.0) go to 240	0545
	write(1,922) i,lox(i),pw(i)	0546
240	continue	0547
241	write(1,923)	0548
	call yesno(la,nerr)	0549
	if(nerr.ne.0) go to 241	0550
	if(la.eq.ly) go to 250	0551
242	write(1,924)	0552
243	write(1,907)	0553
	read(1,908,err=243) i,q	0554
	if(i.eq.99) go to 200	0555
	if(i.eq.0) go to 250	0556
	if(i.le.nox) go to 249	0557
	nl=5	0558
	do 245 n3=1,10	0559
	if(nl.ge.nox) go to 246	0560
245	nl=nl+5	0561
246	nl=1	0562
	do 247 i=1,n3	0563
	n2=nl+(4*n3)	0564
	write(1,906) (k,lox(k),k=nl,n2,n3)	0565
247	nl=nl+1	0566
	go to 243	0567

249	pw(i)=q	0568
	go to 242	0569
250	continue	0570
	if(la.eq.ln) go to 238	0571
255	write(1,925) ldes	0572
	call yesno(la,nerr)	0573
	if(nerr.ne.0) go to 255	0574
	if(la.eq.ly) go to 260	0575
256	write(1,920)	0576
	read(1,998) ldes	0577
260	continue	0578
c		0579
c	Writing clay analysis into new data file	0580
c	-----	0581
c		0582
	if(lr(12).eq.ly) go to 280	0583
	if(nofile.eq.1) go to 280	0584
	nal=nal+1	0585
	if(lr(11).eq.ln) go to 280	0586
	write(12,998)ldes	0587
	nq=0	0588
	do 265 i=1,nox	0589
	if(pw(i).eq.0.0.and.i.lt.nox) go to 265	0590
	if(pw(i).eq.0.0) go to 262	0591
	nq=nq+1	0592
	ne(nq)=i	0593
	qe(nq)=pw(i)	0594
	if(nq.eq.6) go to 264	0595
	if(i.lt.nox) go to 265	0596
262	n=nq+1	0597
	if(n.gt.6) go to 264	0598
	do 263 k=n,6	0599
	ne(k)=0	0600
263	qe(k)=0.0	0601
264	write(12,872) id(1),(ne(k),qe(k),k=1,6)	0602
	nq=0	0603
265	continue	0604
	write(12,874) id(3),lr(7)	0605
	if(lr(14).eq.ly) go to 755	0606
	go to 280	0607
c		0608
c	Writing non-bank components into new data file	0609
c	-----	0610
c		0611
270	continue	0612
	if(nax.eq.nox) go to 275	0613
	nnqq=nax+1	0614
	do 271 i=nnqq,nox	0615
271	write(12,873,err=761,end=275) id(2),lox(i),jc(i),wm(i),kc(i),kn(i)	0616
	-,ky(i)	0617
275	lr(11)=ln	0618
	close(12)	0619
	go to 780	0620

280	continue	0621
c		0622
c	Status of LOI (loss-on-ignition)	0623
c	-----	0624
c		0625
	if(pw(13).eq.0.or.lr(15).eq.1n) go to 282	0626
	write(1,875)	0627
	read(1,991,err=280) kloir	0628
	if(kloir.lt.0) go to 280	0629
282	continue	0630
	if(lr(15).eq.ly) go to 400	0631
	qiron=pw(4)+pw(5)	0632
	nfic=0	0633
	nfus=0	0634
	if(qiron.eq.0.0) go to 284	0635
	if(pw(4).gt.0.0) nfic=1	0636
	if(pw(5).gt.0.0) nfus=1	0637
	if(xiron.gt.1.1) go to 284	0638
	nfic=1	0639
	nfus=1	0640
	if(xiron.eq.0.0) nfus=0	0641
	if(xiron.eq.1.0) nfic=0	0642
284	continue	0643
c		0644

Selecting structural formula specification

c	*****	0645
c	SELECTING STRUCTURAL FORMULA SPECIFICATIONS	0646
c	*****	0647
c		0648
c		0649
c	Summary listing of current specifications	0650
c	-----	0651
c		0652
	if(nono.eq.0) go to 400	0653
	if(lr(12).eq.ly) write(1,998) ldes	0654
	if(nl.ne.100) go to 305	0655
	nnoh=0	0656
	q=pw(11)+pw(12)	0657
	if(kloir.ne.0) q=q+pw(13)	0658
	if(q.gt.0.0) nnoh=1	0659
	nform=0	0660
	go to 340	0661
305	continue	0662
	nbeg=2	0663
	write(1,8001)	0664
c	Listing of water characteristics	0665
	nnoh=0	0666
	qwat=pw(11)+pw(12)+pw(13)	0667
	if(qwat.eq.0.0) go to 3053	0668
	write(1,8002) ltl	0669
	if(pw(13).eq.0.0) go to 3051	0670

	if(klio.eq.0) write(1,876)	0671
	if(klio.ne.0) write(1,877)	0672
	if(klio.eq.0.and.pw(13).ne.0.0) iww(3)=3	0673
3051	continue	0674
	do 3052 i=1,3	0675
	if(i.eq.3.and.kloi.eq.0) go to 3052	0676
	k=i+10	0677
	if(pw(k).eq.0.0) go to 3052	0678
	if(i.eq.1.and.iww(i).eq.0) iww(i)=1	0679
	if(i.gt.1.and.iww(i).eq.0) iww(i)=3	0680
	write(1,8003) lox(k),lwat(iww(i))	0681
	if(iwat.gt.0.and.iww(i).eq.1) nnoh=1	0682
3052	continue	0683
	if(nnoh.eq.0) go to 3053	0684
	if(iwat.eq.1) write(1,8004)	0685
	if(iwat.eq.2) write(1,8005)	0686
c	Listing of iron redox properties	0687
3053	continue	0688
	if(qiron.eq.0.0) go to 3055	0689
	if(xiron.gt.1.1) hfp=hfa	0690
	if(xiron.lt.1.1) write(hfp,8006) hff,xiron	0691
	write(1,8007) lt2,hfp	0692
c	Listing of components deleted from structural formula	0693
3055	continue	0694
	if(hdel.ne.0) go to 3057	0695
	write(1,8008) lt3	0696
	do 3056 i=1,nox	0697
3056	ndel(i)=0	0698
	go to 306	0699
3057	nq=0	0700
	do 3058 i=1,nox	0701
	if(i.lt.21) no(i)=40	0702
	if(ndel(i).eq.0) go to 3058	0703
	nq=nq+1	0704
	no(nq)=i	0705
3058	continue	0706
	if(nq.ne.0) go to 3059	0707
	hdel=0	0708
	go to 3055	0709
3059	write(1,8009) lt3,(lox(no(i)),i=1,5)	0710
	if(nq.gt.5) write(1,8010) (lox(no(i)),i=6,20)	0711
c	Listing of structural formula type	0712
306	continue	0713
	if(nl.eq.0) write(1,8031) lt4	0714
	if(nl.gt.10) write(1,8032) lt4	0715
	if(nl.gt.0.and.nl.lt.11) write(1,8012) lt4,1bt(nl)	0716
c	Listing of calculation parameters in structural formula	0717
	write(1,8025)	0718
	if(nform.eq.0) write(1,8013) lt5,nneg	0719
	if(nform.eq.1) write(1,8020) lt8,nneg	0720
	if(nform.eq.2.and.ncats.eq.0) write(1,8021) lt9,ntoc	0721
	if(nform.eq.2.and.ncats.eq.1) write(1,8022) lt11,ntoc	0722
	if(nl.ne.0) write(1,8014) lt6,ntet	0723

	if(iwat.ne.0.and.nnoh.ne.0) write(1,8026) ltl0,nwat	0724
c	Listing of cation site assignments in a phyllosilicate	0725
	if(nl.eq.0) go to 307	0726
	if(nzz.eq.0) lras=lraa(2)	0727
	if(nzz.ne.0) lras=lraa(1)	0728
	write(1,8015) lt7,lcas,lras	0729
	if(nzz.eq.0) go to 307	0730
	do 3069 i=1,3	0731
	do 3061 k=1,30	0732
3061	a4(k)=a4b	0733
	if(i.eq.1.and.itet.eq.0) go to 3069	0734
	if(i.ne.1) go to 3065	0735
	ltas=ltaa(1)	0736
	nq=0	0737
	do 3062 k=1,7	0738
	if(nt(k).eq.0) go to 3063	0739
	nq=nq+1	0740
3062	a4(nq)=1c(nt(k))	0741
3063	continue	0742
	go to 3068	0743
3065	nq=0	0744
	l=i-1	0745
	ltas=ltaa(i)	0746
	do 3066 k=1,nox	0747
	if(nz(k).eq.0.or.ny(k).ne.1) go to 3066	0748
	nq=nq+1	0749
	a4(nq)=1c(k)	0750
3066	continue	0751
	if(nq.eq.0) go to 3069	0752
3068	write(1,8018) ltas,(a4(k),k=1,10)	0753
	if(nq.gt.10) write(1,8019) (a4(k),k=11,20)	0754
	if(nq.gt.20) write(1,8019) (a4(k),k=21,30)	0755
3069	continue	0756
	if(notet.eq.0) go to 307	0757
	nq=0	0758
	do 3067 i=1,7	0759
	a4(i)=a4b	0760
	if(nt(i).eq.0) go to 3067	0761
	if(ny(nt(i)).ne.0) go to 3067	0762
	nq=nq+1	0763
	a4(nq)=1c(nt(i))	0764
3067	continue	0765
	if(nq.eq.0) a4(1)=none	0766
	write(1,8030) (a4(i),i=1,7)	0767
c		0768
c	Electing options for modifying structural formula parameters	0769
c	-----	0770
c		0771
307	continue	0772
	if(nbeg.eq.2) go to 308	0773
	if(nl.eq.0) write(1,8028)	0774
	if(nl.ne.0) write(1,8029)	0775
308	write(1,8017)	0776

read(1,991,err=307) n	0777
if(n.lt.0) go to 307	0778
if(n.gt.6) go to 305	0779
nbeg=0	0780
if(n.eq.0) go to 390	0781
if(n.eq.1) go to 310	0782
if(n.eq.2) go to 320	0783
if(n.eq.3) go to 330	0784
if(n.eq.4) go to 340	0785
if(n.eq.5) go to 350	0786
if(n.eq.6) go to 370	0787
c	0788
c Modifying water characteristics in structural formula	0789
c -----	0790
c	0791
310 continue	0792
if(pw(13).eq.0.0) go to 3101	0793
write(1,875)	0794
read(1,991,err=310) klo1	0795
3101 continue	0796
do 311 i=1,3	0797
311 iww(i)=0	0798
do 315 i=1,3	0799
ii=i+10	0800
if(pw(ii).eq.0.0) go to 315	0801
if(klo1.eq.0.and.i.eq.3) go to 315	0802
k1=1	0803
k2=3	0804
if(i.gt.1.and.iww(1).gt.1) k1=2	0805
if(i.gt.1.and.iww(1).gt.2) k1=3	0806
if(i.ne.3) go to 312	0807
if(iww(2).gt.1.and.k1.lt.2) k1=2	0808
if(iww(2).gt.2.and.k1.lt.3) k1=3	0809
312 write(1,8210) lox(ii)	0810
313 continue	0811
do 314 k=k1,k2	0812
314 write(1,8211) k,lwat(k)	0813
read(1,991,err=313) iww(i)	0814
if(iww(i).lt.1.or.iww(i).gt.3) go to 312	0815
315 continue	0816
meg=0	0817
if(iwat.ne.0) meg=1	0818
iwat=0	0819
do 3161 i=1,3	0820
if(iww(i).eq.1) go to 3162	0821
3161 continue	0822
go to 317	0823
3162 write(1,8212)	0824
read(1,991,err=3162) iwat	0825
if(iwat.lt.0.or.iwat.gt.2) go to 3162	0826
mug=0	0827
if(iwat.eq.0) go to 317	0828
if(meg.ne.0) go to 317	0829

if(nl.gt.0.and.nl.lt.11) go to 317	0830
mug=1	0831
nnoh=1	0832
go to 3571	0833
317 continue	0834
mug=0	0835
go to 307	0836
c	0837
c Modifying iron redox characteristics in structural formula	0838
c -----	0839
c	0840
320 xiron=2.0	0841
321 write(1,8220)	0842
call yesno(lr(3),nerr)	0843
if(nerr.ne.0) go to 321	0844
nfic=0	0845
nfus=0	0846
if(pw(4).gt.0.0) nfic=1	0847
if(pw(5).gt.0.0) nfus=1	0848
if(lr(3).eq.ln) go to 307	0849
322 write(1,8221)	0850
read(1,903,err=322) xiron	0851
if(xiron.lt.0.0.or.xiron.gt.1.0) go to 322	0852
323 write(1,8222) hff,xiron	0853
nfic=1	0854
nfus=1	0855
if(xiron.eq.0.0) nfus=0	0856
if(xiron.eq.1.0) xfic=0	0857
call yesno(la,nerr)	0858
if(nerr.ne.0) go to 323	0859
if(la.eq.ln) go to 320	0860
go to 307	0861
c	0862
c Modifying list of components for deletion from structural formula	0863
c -----	0864
c	0865
330 nq=0	0866
nng=0	0867
if(hdel.eq.0) go to 332	0868
331 write(1,8230)	0869
read(1,991,err=331) nng	0870
if(nng.lt.1.or.nng.gt.3) go to 331	0871
if(nng.eq.1) go to 3320	0872
332 write(1,8231)	0873
read(1,991,err=332) nm	0874
3320 hdel=0	0875
nq=0	0876
nnq=0	0877
do 333 i=1,nox	0878
if(nng.ne.3) ndel(i)=0	0879
if(nng.eq.1) go to 333	0880
if(nnq.ne.0) go to 3322	0881
do 3321 k=1,8	0882

	a8(k)=a8b	0883
3321	al(k)=id(5)	0884
3322	continue	0885
	if(nm.ne.0) go to 3326	0886
	if(i.ne.4.and.i.ne.5) go to 3325	0887
	if(xiron.gt.1.1.or.qiron.eq.0.0) go to 3325	0888
	go to 3326	0889
3325	continue	0890
	if(pw(i).eq.0.0.and.i.lt.nox) go to 333	0891
3326	nq=nq+1	0892
	if(ndel(i).ne.0) al(nq)=id(4)	0893
	if(pw(i).ne.0.0.or.nm.ne.0) write(a8(nq),8233) i,lox(i)	0894
	nnq=1	0895
	if(nq.ne.8.and.i.ne.nox) go to 333	0896
	write(1,8234) (al(k),a8(k),k=1,8)	0897
	nnq=0	0898
	nq=0	0899
333	continue	0900
	if(nng.eq.1) go to 307	0901
	write(1,8235)	0902
	if(nng.eq.3) write(1,8236)	0903
334	write(1,8237)	0904
	read(1,8662,err=334) (mo(i),i=1,20)	0905
	do 335 i=1,20	0906
	if(mo(i).eq.0) go to 336	0907
	if(ndel(mo(i)).eq.0) ndel(mo(i))=2	0908
	if(ndel(mo(i)).eq.1) ndel(mo(i))=0	0909
	if(ndel(mo(i)).eq.2) ndel(mo(i))=1	0910
335	continue	0911
336	continue	0912
	do 337 i=1,nox	0913
	if(hdel.ne.0) go to 338	0914
	if(ndel(i).eq.1) hdel=1	0915
337	continue	0916
338	continue	0917
	go to 307	0918
c		0919
c	Modifying structural formula type	0920
c	-----	0921
c		0922
340	continue	0923
	write(1,8240)	0924
	read(1,991,err=340) n	0925
	if(n.lt.1.or.n.gt.2) go to 340	0926
	if(n.eq.2) go to 341	0927
	n1=0	0928
	n11=0	0929
348	write(1,8242)	0930
	call yesno(la,nerr)	0931
	if(nerr.ne.0) go to 348	0932
	if(la.eq.ln) go to 350	0933
	lcas=lcaa(1)	0934
	lras=lraa(2)	0935

	nl=11	0936
	go to 350	0937
341	continue	0938
	do 342 i=1,20	0939
	ii=i+10	0940
342	mi(i)=ii	0941
343	write(1,824) (mi(i),lbt(i),i=1,10)	0942
	read(1,992,err=343) n	0943
	nl=n-10	0944
	if(nl.lt.1.or.nl.gt.10) go to 343	0945
	do 344 i=1,nox	0946
	if(nz(i).ne.0) go to 344	0947
	ny(i)=ky(i)	0948
	if(nl.eq.3.and.ky(i).eq.2) ny(i)=1	0949
344	continue	0950
	if(nl.gt.7.and.nz(9).eq.0) ny(9)=1	0951
	if(nl.gt.7.and.nz(19).eq.0) ny(19)=2	0952
	lcas=lcaa(1)	0953
	if(nl.eq.3) lcas=lcaa(2)	0954
	if(nl.gt.7) lcas=lcaa(3)	0955
	if(nform.eq.0) kneg=nd(nl)	0956
	if(nform.ne.0) kneg=na(nl)	0957
	ktoc=ndto(nl)	0958
	kwat=nw(nl)	0959
	ntet=jtet(nl)	0960
	nqq=1	0961
	go to 3501	0962
346	n11=1	0963
	if(nl.eq.3) n11=2	0964
	if(nl.gt.7) n11=3	0965
345	xneg=float(nneg)	0966
	xtet=float(ntet)	0967
	xtoc=float(ntoc)	0968
	xwat=float(nwat)	0969
	if(nbeg.eq.1) go to 305	0970
	if(mug.eq.1) go to 317	0971
	go to 307	0972
c		0973
c	Modifying parameters of structural formula	0974
c	-----	0975
c		0976
350	nqq=0	0977
c	Selecting method of calculation	0978
3501	write(1,8250)	0979
	read(1,991,err=3501) nform	0980
	if(nform.lt.0.or.nform.gt.2) go to 350	0981
	if(nl.lt.1.or.nl.gt.10) go to 359	0982
c	Assignment of default parameters	0983
	if(nform.eq.0) kneg=nd(nl)	0984
	if(nform.eq.1) kneg=na(nl)	0985
	ktoc = ndto(nl)	0986
	kwat=nw(nl)	0987
359	continue	0988

	if(nform.ne.2) go to 352	0989
	write(1,8244)	0990
	read(1,991,err=359) ncats	0991
	if(ncats.lt.0.or.ncats.gt.1) go to 359	0992
	if(nl.lt.1.or.nl.gt.10) go to 353	0993
	if(ncats.eq.1) go to 353	0994
351	write(1,8243)	0995
	call yesno(1a,nerr)	0996
	if(nerr.ne.0) go to 351	0997
	if(1a.eq.1n) ktoc=ntto(nl)	0998
352	continue	0999
	if(nl.lt.1.or.nl.gt.10) go to 353	1000
	write(1,8252)	1001
	call yesno(1a,nerr)	1002
	if(nerr.ne.0) go to 352	1003
	ndbl=1	1004
	if(1a.eq.1y) ndbl=2	1005
	nneg=ndbl*kneg	1006
	ntoc=ndbl*ktoc	1007
	nwat=ndbl*kwat	1008
	ntet=ndbl*jtet(nl)	1009
	if(nqq.eq.1) go to 346	1010
	go to 360	1011
c	Modifying basic calculation parameter	1012
353	continue	1013
	if(nform.eq.0) write(1,8253) 1t5,nneg	1014
	if(nform.eq.1) write(1,8254) 1t8,nneg	1015
	if(nform.eq.2.and.ncats.eq.0) write(1,8255) 1t9,ntoc	1016
	if(nform.eq.2.and.ncats.eq.1) write(1,8258) 1t11,ntoc	1017
	write(1,994)	1018
	call yesno(1a,nerr)	1019
	if(nerr.ne.0) go to 353	1020
	if(1a.eq.1y.and.nqq.eq.1) go to 352	1021
	if(1a.eq.1y) go to 355	1022
354	write(1,8251)	1023
	read(1,992,err=354) n	1024
	if(nform.lt.2) nneg=n	1025
	if(nform.eq.2) ntoc=n	1026
	if(nqq.eq.1) ktoc=ntoc	1027
	if(nqq.eq.1) go to 352	1028
	go to 353	1029
c	Modifying number of tetrahedral sites in formula unit	1030
355	continue	1031
	if(nl.eq.0) go to 357	1032
	write(1,8256) 1t6,ntet	1033
	write(1,994)	1034
	call yesno(1a,nerr)	1035
	if(nerr.ne.0) go to 355	1036
	if(1a.eq.1y) go to 357	1037
356	write(1,8251)	1038
	read(1,992,err=356) ntet	1039
	go to 355	1040
c	Modifying number of (OH,F,Cl) sites in formula unit	1041

357	continue	1042
	if(iwat.eq.0.or.nnoh.eq.0) go to 360	1043
3571	write(1,8257) 1t10,nwat	1044
	write(1,994)	1045
	call yesno(1a,nerr)	1046
	if(nerr.ne.0) go to 3571	1047
	if(1a.eq.ly.and.mug.eq.1) go to 345	1048
	if(1a.eq.ly) go to 360	1049
358	write(1,8251)	1050
	read(1,992,err=358) nwat	1051
	go to 357	1052
c	Check-list of formula parameters	1053
360	write(1,8260)	1054
	if(nform.eq.0) write(1,8253) 1t5,nneg	1055
	if(nform.eq.1) write(1,8254) 1t8,nneg	1056
	if(nform.eq.2.and.ncats.eq.0) write(1,8255) 1t9,ntoc	1057
	if(nform.eq.2.and.ncats.eq.1) write(1,8258) 1t11,ntoc	1058
	if(nl.gt.0) write(1,8256) 1t6,ntet	1059
	if(iwat.eq.0.or.nnoh.eq.0) go to 362	1060
	write(1,8257) 1t10,nwat	1061
362	write(1,994)	1062
	call yesno(1a,nerr)	1063
	if(nerr.ne.0) go to 362	1064
	if(1a.eq.ln) go to 353	1065
	go to 345	1066
c		1067
c	Modifying cation site assignments in clay-mineral formula	1068
c	-----	1069
c		1070
370	write(1,8270) (i,lcaa(i),i=1,3)	1071
	read(1,991,err=370) n	1072
	nm=2	1073
	if(n.lt.1.or.n.gt.3) go to 370	1074
c	Reassigning default-type	1075
	1a=ln	1076
	if(nzz.eq.0) go to 3703	1077
3701	write(1,8278)	1078
	call yesno(1a,nerr)	1079
	if(nerr.ne.0) go to 3701	1080
3703	continue	1081
	if(n.gt.0.and.n.lt.4) n11=n	1082
	do 371 i=1,nox	1083
	if(1a.eq.ly.and.nz(i).ne.0) go to 371	1084
	nz(i)=0	1085
	ny(i)=ky(i)	1086
	if(n11.eq.2.and.ky(i).eq.2) ny(i)=1	1087
371	continue	1088
	if(n11.ne.3) go to 372	1089
	if(nz(9).eq.0) ny(9)=1	1090
	if(nz(19).eq.0) ny(19)=2	1091
372	lcas=lcaa(n)	1092
	if(1a.eq.ly) go to 373	1093
	nt(1)=1	1094

nt(2)=3	1095
do 3725 i=3,7	1096
3725 nt(i)=0	1097
nzz=0	1098
itet=0	1099
notet=0	1100
373 write(1,8271)	1101
call yesno(1a,nerr)	1102
if(nerr.ne.0) go to 373	1103
if(1a.eq.1n) go to 307	1104
c Displaying listing of cation site assignments	1105
374 jt=0	1106
jo=0	1107
ji=0	1108
do 3742 k=1,30	1109
do 3741 i=1,3	1110
3741 llst(i,k)=a8b	1111
3742 continue	1112
if(nm.lt.2) go to 3744	1113
3743 write(1,8231)	1114
read(1,991,err=3743) nm	1115
3744 nzz=0	1116
itet=0	1117
do 375 i=1,nox	1118
if(ny(i).gt.2) go to 375	1119
if(nm.ne.0) go to 3746	1120
if(i.ne.4.and.i.ne.5) go to 3745	1121
if(i.eq.4.and.nfic.eq.1) go to 3746	1122
if(i.eq.5.and.nfus.eq.1) go to 3746	1123
go to 375	1124
3745 continue	1125
if(pw(i).eq.0.0) go to 375	1126
3746 ll=id(5)	1127
if(nz(i).ne.0) ll=id(4)	1128
if(ll.eq.id(4)) nzz=1	1129
if(ny(i).ne.1) go to 3747	1130
jo=jo+1	1131
write(llst(2,jo),8273) ll,i,lc(i)	1132
3747 continue	1133
if(ny(i).ne.2) go to 375	1134
ji=ji+1	1135
write(llst(3,ji),8273) ll,i,lc(i)	1136
375 continue	1137
hl=0	1138
h2=0	1139
do 376 i=1,7	1140
ll=id(5)	1141
if(nt(i).eq.0) go to 376	1142
jt=jt+1	1143
if(nt(i).ne.1.and.nt(i).ne.3) go to 3751	1144
if(nt(i).eq.1) hl=1	1145
if(nt(i).eq.1.and.h2.eq.1) go to 3751	1146
if(nt(i).eq.3) h2=1	1147

	if(nt(i).eq.3.and.h1.eq.0) go to 3751	1148
	go to 3752	1149
3751	itet=1	1150
	if(nt(i).ne.1.and.nt(i).ne.3) ll=id(4)	1151
3752	write(11st(1,jt),8273) ll,nt(i),lc(nt(i))	1152
376	continue	1153
	if(h1.eq.0.or.h2.eq.0) itet=1	1154
	if(itet.ne.0) nzz=1	1155
377	ii=jt	1156
	if(jo.gt.ii) ii=jo	1157
	if(ji.gt.ii) ii=ji	1158
	write(1,8274)	1159
	if(nl.gt.10) write(1,8288) (lcat(i),i=1,2)	1160
	if(nl.gt.0.and.nl.lt.11) write(1,8289) (lcat(i),i=1,3)	1161
	do 378 i=1,ii	1162
378	write(1,8275) (11st(j,i),j=1,3)	1163
	nq=0	1164
	do 3781 i=1,7	1165
	a4(i)=a4b	1166
	if(nt(i).eq.0) go to 3781	1167
	if(ny(nt(i)).ne.0) go to 3781	1168
	nq=nq+1	1169
	a4(nq)=lc(nt(i))	1170
3781	continue	1171
	if(nq.eq.0) a4(1)=none	1172
	write(1,8030) (a4(i),i=1,7)	1173
	if(nzz.ne.0.or.itet.ne.0) write(1,8276)	1174
c	Requesting reassignment option	1175
379	write(1,8277)	1176
	if(nzz.ne.0) write(1,8279)	1177
	read(1,991,err=379) n	1178
	if(n.lt.0.or.n.gt.4) go to 379	1179
	if(n.eq.0) go to 307	1180
	if(n.eq.3) go to 382	1181
c	Octahedral and interlayer site reassignments	1182
380	continue	1183
	if(n.lt.4) write(1,8280) lcat(n+1)	1184
	if(n.eq.4) write(1,8287)	1185
	read(1,992,err=380) nq	1186
	if(nq.lt.0.or.nq.gt.nox) go to 380	1187
	if(nq.eq.0) go to 374	1188
	if(n.eq.4) go to 381	1189
	nz(nq)=1	1190
	ny(nq)=n	1191
	go to 380	1192
381	nz(nq)=0	1193
	ny(nq)=ky(nq)	1194
	if(n11.eq.2.and.ny(nq).eq.2) ny(nq)=1	1195
	if(n11.eq.3.and.nq.eq.9) ny(nq)=1	1196
	if(n11.eq.3.and.nq.eq.19) ny(nq)=2	1197
	go to 380	1198
c	Reassigning order of filling tetrahedral sites	1199
382	write(1,8281)	1200

do 383 i=1,7	1201
if(nt(i).eq.0) go to 3831	1202
383 write(1,8282) i,nt(i),lc(nt(i))	1203
3831 write(1,994)	1204
call yesno(1a,nerr)	1205
if(nerr.eq.1) go to 3831	1206
if(1a.eq.ly) go to 389	1207
write(1,8283)	1208
do 384 i=1,10	1209
if(ntt(i).eq.0) go to 385	1210
384 write(1,8284) ntt(i),lc(ntt(i))	1211
385 write(1,8285)	1212
read(1,8286,err=385) (nt(i),i=1,7)	1213
go to 382	1214
389 continue	1215
write(1,8293)	1216
call yesno(1a,nerr)	1217
if(nerr.ne.0) go to 389	1218
if(1a.eq.ln) go to 3901	1219
3891 write(1,8294)	1220
read(1,8286,err=3891) (ktt(i),i=1,7)	1221
j=1	1222
jj=0	1223
do 3895 i=1,7	1224
if(ktt(i).eq.0) go to 3896	1225
if(ktt(i).eq.1) j=0	1226
if(ktt(i).gt.1) jj=1	1227
ny(ktt(i))=0	1228
do 3892 k=1,7	1229
if(ktt(i).eq.nt(k)) go to 3895	1230
if(nt(k).eq.0) go to 3893	1231
3892 continue	1232
3893 nt(k)=ktt(i)	1233
3895 continue	1234
3896 continue	1235
notet=0	1236
if(j.eq.1.or.jj.eq.1) notet=1	1237
k=0	1238
do 3897 i=1,7	1239
ktt(i)=0	1240
if(nt(i).eq.0) go to 3897	1241
if(ny(nt(i)).ne.0) go to 3897	1242
k=k+1	1243
ktt(k)=nt(i)	1244
nt(i)=0	1245
3897 continue	1246
do 3898 i=1,7	1247
if(nt(i).eq.0) go to 3898	1248
k=k+1	1249
ktt(k)=nt(i)	1250
3898 nt(i)=0	1251
do 3899 i=1,7	1252
nt(i)=ktt(i)	1253

3899	ktt(i)=0	1254
3901	continue	1255
	do 3906 i=1,nox	1256
	if(ny(i).ne.0) go to 3905	1257
	do 3903 k=1,7	1258
	if(nt(k).eq.i) go to 3905	1259
3903	continue	1260
	ny(i)=ky(i)	1261
	if(ny(i).eq.0) ny(i)=1	1262
	if(ny(i).ne.ky(i)) nz(i)=1	1263
3905	continue	1264
	if(i.gt.1.and.ny(i).eq.0) notet=1	1265
3906	continue	1266
	if(ny(1).ne.0) notet=1	1267
	go to 374	1268
c	Setting constants for calculating structural formula	1269
390	continue	1270
	if(nono.ne.2) go to 400	1271
391	write(1,8290)	1272
	call yesno(1a,nerr)	1273
	if(nerr.ne.0) go to 391	1274
	if(1a.eq.1n) nono=1	1275
c		1276

Structural formula calculation

c*****	1277	
c	STRUCTURAL FORMULA CALCULATION	1278
c*****	1279	
c		1280
c	Mole percentages and abundances in general silicate formula	1281
c	-----	1282
c		1283
400	continue	1284
	ibor=0	1285
	isil=0	1286
	kbor=0	1287
	if(pw(30).gt.0.0.and.ny(30).eq.1.and.n1.eq.11) kbor=1	1288
	do 401 i=1,50	1289
	xm(i)=0.0	1290
	xmm(i)=0.0	1291
	pm(i)=0.0	1292
	xc(i)=0.0	1293
	c(i)=0.0	1294
	xa(i)=0.0	1295
	a(i)=0.0	1296
401	continue	1297
	kkwat=0	1298
	tpw=0.0	1299
	txm=0.0	1300
	tpm=0.0	1301
	txa=0.0	1302
	toc=0.0	1303

	oxeq=0.0	1304
	anno=0.0	1305
c	Molar and atom proportions	1306
	do 405 i=1,nox	1307
	if(pw(i).eq.0.0) go to 405	1308
	tpw=tpw+pw(i)	1309
	xm(i)=pw(i)/wm(i)	1310
405	continue	1311
	if(kloi.eq.0) xm(13)=0.0	1312
c	F and Cl in molar proportions	1313
	ho=0	1314
	q=pw(28)+pw(29)	1315
	if(q.lt.1.e-6) go to 406	1316
	qho=(pw(28)*0.4210723)+(pw(29)*0.2256424)	1317
	qhot=tpw-qho	1318
	ho=1	1319
406	continue	1320
c	Structural formula molar proportions	1321
	do 412 i=1,nox	1322
	if(i.lt.14.and.i.gt.10) go to 412	1323
	if(ndel(i).eq.0) xmm(i)=xm(i)	1324
412	continue	1325
c	Iron redox redistribution	1326
	if(xm(4).eq.0.0.and.xm(5).eq.0.0) go to 413	1327
	if(xiron.gt.1.0) go to 413	1328
	x2=xm(5)	1329
	x3=xm(4)*2.0	1330
	q=x2+x3	1331
	xmm(5)=xiron*q	1332
	xmm(4)=0.0	1333
	if(xiron.lt.1.00) xmm(4)=(q-xmm(5))/2.0	1334
413	continue	1335
c	Redistribution of H2O+, H2O-, LOI as formula components	1336
	do 416 i=1,2	1337
	ii=i+10	1338
	do 415 k=1,3	1339
	kk=k+10	1340
	if(ndel(kk).ne.0) go to 415	1341
	if(iww(k).eq.i) xmm(ii)=xmm(ii)+xm(kk)	1342
415	continue	1343
416	continue	1344
c	Atom proportions in structural formula	1345
	do 417 i=1,nox	1346
	if(xm(i).eq.0.0.and.xmm(i).eq.0.0) go to 417	1347
	if(i.ne.28.and.i.ne.29) txm=txm+xm(i)	1348
	xc(i)=xmm(i)*xnc(i)	1349
417	continue	1350
	scat=0.0	1351
	swat=xc(11)+xc(28)+xc(29)	1352
	do 418 i=1,nox	1353
	if(xm(i).ne.0.0) pm(i)=(xm(i)/txm)*100.0	1354
	if(i.ne.28.and.i.ne.29) tpm=tpm+pm(i)	1355
	if(ny(i).gt.2) go to 418	1356

	if(xc(i).eq.0.0) go to 418	1357
	scat=scat+(xc(i)*(xnn(i)/(2.0*xnc(i))))	1358
	if(ny(i).lt.2) toc=toc+xc(i)	1359
	if(ny(i).eq.2.and.ncats.eq.1) toc=toc+xc(i)	1360
418	continue	1361
c	Atom abundances in structural formula	1362
	xsite=scat+(0.5*swat)	1363
	if(nform.eq.0) xsite=scat	1364
	if(nform.eq.2) xsite=toc	1365
	if(nform.ne.2) qq=xneg	1366
	if(nform.eq.2) qq=xtoc	1367
	do 419 i=1,nox	1368
	if(xc(i).eq.0.0) go to 419	1369
	c(i)=(xc(i)*qq)/xsite	1370
419	continue	1371
	if(xc(11).eq.0.0) go to 440	1372
	if(iwat.eq.0) go to 440	1373
	if(iwat.eq.2) go to 432	1374
c	Structural formula with H3O+	1375
	if(nform.ne.2) go to 421	1376
	if(ncats.eq.1) go to 421	1377
	toh=c(11)+c(28)+c(29)	1378
	if(toh.le.xwat) go to 440	1379
	q=toh-xwat	1380
	if(q.gt.c(11)) q=c(11)	1381
	c(13)=q/3.0	1382
	c(11)=c(11)-q	1383
	if(c(13).gt.0.0) kkwat=1	1384
	go to 440	1385
421	continue	1386
	if(xwat.ne.0.0) go to 422	1387
	xc(13)=xc(11)/3.0	1388
	xc(11)=0.0	1389
	xsite=xsite+(0.5*xc(13))	1390
	if(nform.eq.2) xsite=xsite+(0.5*xc(13))	1391
	if(xc(13).gt.0.0) kkwat=1	1392
	go to 440	1393
422	xrat=xneg/xwat	1394
	if(nform.eq.2) xrat=xtoc/xwat	1395
	toh=xc(11)+xc(28)+xc(29)	1396
	yrate=xsite/toh	1397
	if(yrate.ge.xrat) go to 440	1398
	do 425 i=1,3	1399
	ta(i)=0.0	1400
	do 424 k=1,3	1401
424	tx(k,i)=0.0	1402
425	continue	1403
	ta(1)=xsite	1404
	if(nform.eq.1) ta(1)=ta(1)-(0.5*xc(11))	1405
	ta(2)=xc(28)+xc(29)	1406
	ta(3)=xc(11)	1407
	tx(1,1)=1.0	1408
	tx(1,2)=-0.5	1409

	if(nform.eq.2) tx(1,2)=-1.0	1410
	if(nform.eq.1) tx(1,3)=-0.5	1411
	tx(2,1)=xwat/xneg	1412
	tx(2,3)=-1.0	1413
	tx(3,2)=3.0	1414
	tx(3,3)=1.0	1415
	call array(2,3,3,3,3,ty,tx)	1416
	do 426 i=1,9	1417
	if(i.lt.4) adp(i)=dble(ta(i))	1418
	xdp(i)=dble(ty(i))	1419
426	continue	1420
	call dgelg(adp,xdp,3,1,1.e-10,ier)	1421
	if(ier.ne.0) go to 427	1422
	go to 428	1423
427	write(1,955) ier	1424
	go to 305	1425
428	continue	1426
	do 429 i=1,3	1427
	ta(i)=sngl(adp(i))	1428
429	continue	1429
	q=(ta(2)*xneg)/ta(1)	1430
	if(q.lt.1.e-6.and.q.gt.-1.e-6) ta(2)=0.0	1431
	if(ta(2).lt.0.0) go to 427	1432
	if(ta(2).eq.0.0) go to 440	1433
	if(ta(3).ge.0.0) go to 431	1434
	xc(13)=xc(11)/3.0	1435
	xc(11)=0.0	1436
	if(nform.lt.2) xsite=xsite+(0.5*xc(13))	1437
	if(nform.eq.1) xsite=xsite-(0.5*3.0*xc(13))	1438
	if(nform.eq.2) xsite=xsite+xc(13)	1439
	if(xc(13).gt.0.0) kkwat=1	1440
	go to 420	1441
431	xsite=ta(1)	1442
	xc(13)=ta(2)	1443
	xc(11)=ta(3)	1444
	if(xc(13).gt.0.0) kkwat=1	1445
420	continue	1446
	do 430 i=1,nox	1447
	if(xc(i).ne.0.0) c(i)=(xc(i)*xneg)/xsite	1448
430	continue	1449
	go to 440	1450
c	Excess OH as interlayer H2O	1451
432	continue	1452
	toh=c(11)+c(28)+c(29)	1453
	if(toh.le.xwat) go to 440	1454
	if(nform.eq.1) go to 433	1455
	q=xwat-(c(28)+c(29))	1456
	if(q.lt.0.0) q=0.0	1457
	if(c(11).gt.q) c(12)=c(12)+(0.5*(c(11)-q))	1458
	if(c(11).gt.q) kkwat=1	1459
	if(c(11).gt.q) c(11)=q	1460
	go to 440	1461
433	toh=xc(11)+xc(28)+xc(29)	1462

	q=xwat/(xneg-(0.5*xwat))	1463
	qa=q*scat	1464
	q=toh-qa	1465
	if(q.le.0.0) go to 440	1466
	if(q.gt.xc(11)) q=xc(11)	1467
	if(q.gt.0.0) kkwat=1	1468
	xc(12)=xc(12)+(0.5*q)	1469
	xc(11)=xc(11)-q	1470
	xsite=scat+(0.5*(xc(11)+xc(28)+xc(29)))	1471
	qq=xneg	1472
	go to 420	1473
440	continue	1474
	if(ho.eq.0) go to 442	1475
	xfs=0.0	1476
	xcs=0.0	1477
	if(pm(28).gt.0.0) xfs=pm(28)/4.0	1478
	if(pm(29).gt.0.0) xcs=pm(29)/4.0	1479
442	continue	1480
	do 444 i=1,nox	1481
	if(ny(i).gt.2) go to 444	1482
	if(c(i).eq.0.0) go to 444	1483
	oxeq=oxeq+(c(i)*(xnn(i)/(2.0*xnc(i))))	1484
444	continue	1485
	q=c(11)+c(28)+c(29)	1486
	anno=oxeq+(0.5*q)	1487
	if(q.ne.0.0) go to 445	1488
	if(nfor.eq.0) anno=0.0	1489
	if(norm.gt.0) oxeq=0.0	1490
445	continue	1491
	if(nform.eq.0) oxeq=0.0	1492
	if(nform.eq.1) anno=0.0	1493
	if(lr(15).eq.ly) go to 600	1494
	if(nl.lt.1) go to 600	1495
c		1496
c	Cation distribution in clay-mineral structural formula	1497
c	-----	1498
c		1499
c	Tetrahedral layer cations	1500
	stot(1) = 0.0	1501
	n = 0	1502
	cz(1)=0.0	1503
	cz(2)=0.0	1504
	cz(3)=0.0	1505
	do 501 i=41,47	1506
	c(i)=0.0	1507
	n=n+1	1508
	if(nt(n).eq.0) go to 501	1509
	xnn(i)=xnn(nt(n))	1510
	xnc(i)=xnc(nt(n))	1511
	lc(i)=lc(nt(n))	1512
501	continue	1513
503	continue	1514
	do 508 i=1,7	1515

if(nt(i).eq.0) go to 508	1516
if(ny(nt(i)).ne.0) go to 508	1517
if(c(nt(i)).eq.0.0) go to 508	1518
m=i+40	1519
c(m)=c(nt(i))	1520
c(nt(i))=0.0	1521
stot(1)=stot(1)+c(m)	1522
cz(1)=cz(1)+((xnn(m)/xnc(m))*c(m))	1523
508 continue	1524
qq=xtet-stot(1)	1525
if(qq.lt.0.0) go to 511	1526
n=0	1527
do 507 i=41,47	1528
if(stot(1).ge.xtet) go to 511	1529
n=n+1	1530
if(nt(n).eq.0) go to 507	1531
if(ny(nt(n)).eq.0) go to 507	1532
c(i)=c(nt(n))	1533
q=qq-c(i)	1534
if(q.lt.0.0) go to 504	1535
c(nt(n))=0.0	1536
qq=qq-c(i)	1537
go to 505	1538
504 continue	1539
c(i)=qq	1540
c(nt(n))=c(nt(n))-c(i)	1541
505 continue	1542
cz(1)=cz(1)+(c(i)*(xnn(i)/xnc(i)))	1543
stot(1)=stot(1)+c(i)	1544
507 continue	1545
511 cz(1)=cz(1)-(4.0*xtet)	1546
c Octahedral and interlayer cations, anions, and molecules	1547
do 514 i=2,5	1548
514 stot(i)=0.0	1549
do 521 i=1,nox	1550
if(c(i).eq.0.0) go to 521	1551
if(ny(i).ne.1) go to 515	1552
if(i.eq.1.and.n1.eq.11) go to 521	1553
if(i.eq.30.and.kbor.eq.1) go to 521	1554
stot(2)=stot(2)+c(i)	1555
cz(2)=cz(2)+(c(i)*(xnn(i)/xnc(i)))	1556
go to 521	1557
515 continue	1558
if(ny(i).ne.2) go to 516	1559
stot(3)=stot(3)+c(i)	1560
cz(3)=cz(3)+(c(i)*(xnn(i)/xnc(i)))	1561
go to 521	1562
516 continue	1563
if(ny(i).ne.3) go to 517	1564
stot(4)=stot(4)+c(i)	1565
go to 521	1566
517 continue	1567
if(ny(i).ne.4) go to 521	1568

stot(5)=stot(5)+c(i)	1569
521 continue	1570
if(nform.eq.2) cz(2)=0.0-(cz(1)+cz(3))	1571
if(nform.eq.2) go to 600	1572
q=xneg*2.0	1573
if(nform.eq.1) q=q-(c(11)+c(28)+c(29))	1574
cz(2)=cz(2)-(q-(4.0*xtet))	1575
c	1576

Formatting results of calculation

c*****	1577
c FORMATTING RESULTS OF CALCULATIONS	1578
c*****	1579
c	1580
c Table of analytical data	1581
c -----	1582
c	1583
600 continue	1584
do 601 i=1,99	1585
ha(i)=hab1	1586
601 hf(i)=hfb1	1587
maxa=0	1588
max1=0	1589
mm=0	1590
c	1591
Chemical analysis data	1591
write(ha(2),9200)	1592
write(ha(4),9201) lcap(1),lcap(2)	1593
write(ha(5),9201) lcap(3),lcap(3)	1594
k=0	1595
do 603 i=6,99	1596
602 k=k+1	1597
iii=0	1598
if(k.gt.nox) go to 605	1599
if(pw(io(k)).eq.0.0.and.pm(io(k)).eq.0.0) go to 602	1600
if(io(k).ne.28.and.io(k).ne.29) write(ha(i),9203)	1601
-lox(io(k)),pw(io(k)),pm(io(k))	1602
if(io(k).eq.28.or.io(k).eq.29) write(ha(i),9204)	1603
-lox(io(k)),pw(io(k))	1604
603 iii=1	1605
605 i=i+iii	1606
write(ha(i),9201) lcap(3),lcap(3)	1607
i=i+1	1608
write(ha(i),9205) lcap(4),tpw,tpm	1609
i=i+1	1610
if(ho.eq.0) go to 607	1611
write(ha(i),9206) qho	1612
i=i+1	1613
write(ha(i),9207) qhot	1614
i=i+1	1615
607 maxa=i-1	1616
c	1617
Generalized structural formula	1617
if(lr(15).eq.1y) go to 645	1618

if(nl.gt.0) go to 620	1619
sc=0.0	1620
write(hf(1),9211)	1621
if(nform.eq.0) write(hf(2),9212) lneg,nneg	1622
if(nform.eq.1) write(hf(2),9210) lsit,nneg	1623
if(nform.eq.2.and.ncats.eq.0) write(hf(2),9216) ltoc,ntoc	1624
if(nform.eq.2.and.ncats.eq.1) write(hf(2),9217) lctn,ntoc	1625
i=5	1626
nq=0	1627
610 continue	1628
do 612 k=1,nox	1629
if(ny(io(k)).ne.nq) go to 612	1630
if(c(io(k)).eq.0.0) go to 612	1631
if(ny(io(k)).lt.3) sc=sc+c(io(k))	1632
write(hf(i),9213) lc(io(k)),c(io(k))	1633
i=i+1	1634
612 continue	1635
if(nq.ne.2) go to 615	1636
write(hf(i),9214) lcap(3)	1637
i=i+1	1638
write(hf(i),9215) sc	1639
i=i+2	1640
615 nq=nq+1	1641
if(nq.lt.5) go to 610	1642
maxl=i	1643
go to 645	1644
c	Phyllosilicate structural formula 1645
620 continue	1646
if(nl.gt.10) write(hf(1),9220)	1647
if(nl.lt.11) write(hf(1),9221)	1648
if(nl.lt.11) write(hf(2),9222) 1lt(nl)	1649
k=3	1650
if(nl.gt.10) k=2	1651
if(nform.eq.0) write(hf(k),9223) lneg,nneg	1652
if(nform.eq.1) write(hf(k),9224) lsit,nneg	1653
if(nform.eq.2.and.ncats.eq.0) write(hf(k),9228) ltoc,ntoc	1654
if(nform.eq.2.and.ncats.eq.1) write(hf(k),9229) lctn,ntoc	1655
ii=k+2	1656
if(nl.lt.11.or.ny(1).ne.1.or.c(1).eq.0.0) go to 622	1657
write(hf(ii),9225) 1b1,lc(1),c(1)	1658
isil=1	1659
c(1)=0.0	1660
ii=ii+2	1661
622 il=1	1662
kz=47	1663
iz=0	1664
k=40	1665
621 continue	1666
if(stot(il).eq.0.0) go to 630	1667
if(il.lt.5) 111=1cat(il)	1668
if(il.eq.3.and.nl.gt.7) 111=1big	1669
lines=0	1670
if(nl.gt.10) 111=1b1	1671

do 625 i=ii,99	1672
ic=0	1673
iii=0	1674
623 k=k+1	1675
if(k.gt.kz) go to 627	1676
if(k.gt.40) io(k)=k	1677
if(il.gt.1.and.ny(io(k)).ne.iz) go to 623	1678
if(c(io(k)).eq.0.0) go to 623	1679
if(il.eq.2.and.io(k).eq.30.and.kbor.eq.1) go to 623	1680
write(hf(i),9225) lll,lc(io(k)),c(io(k))	1681
lines=lines+1	1682
lll=lbl	1683
625 iii=1	1684
627 i=i+iii	1685
if(il.eq.5) go to 629	1686
if(lines.lt.2) go to 628	1687
write(hf(i),9226) lcap(4),stot(il)	1688
i=i+1	1689
628 continue	1690
if(nl.ne.ll.or.il.ne.1.or.kbor.ne.1) go to 629	1691
i=i+1	1692
write(hf(i),9225) lll,lc(30),c(30)	1693
ibor=1	1694
i=i+1	1695
629 i=i+1	1696
630 continue	1697
il=il+1	1698
k=0	1699
kz=nox	1700
iz=iz+1	1701
if(il.ne.4) go to 635	1702
if(nl.gt.7) go to 632	1703
ii=i	1704
do 631 i=ii,99	1705
k=k+1	1706
iii=0	1707
if(k.gt.3) go to 632	1708
if(k.eq.3.and.cz(k).eq.0.0) go to 632	1709
write(hf(i),9227) lcz(k),cz(k)	1710
631 iii=0	1711
632 k=0	1712
i=i+iii	1713
q=stot(4)+stot(5)	1714
if(q.eq.0.0) go to 640	1715
if(nl.lt.11) i=i+1	1716
635 continue	1717
if(il.eq.6) go to 640	1718
if(il.eq.5.and.stot(5).eq.0.0) go to 640	1719
ii=i	1720
go to 621	1721
640 maxl=i-1	1722
645 maxx=maxa	1723
if(maxl.gt.maxa) maxx=maxl	1724

650	continue		1725
c			1726
c		Footnotes	1727
c		-----	1728
c			1729
	nf=0		1730
c		Analytical data altered	1731
	if(nofile.eq.0) go to 652		1732
	nf=nf+1		1733
	mm=mm+1		1734
	write(a79(mm),9659) nf		1735
c		Halides in analysis	1736
652	continue		1737
	if(ho.eq.0) go to 655		1738
	if(xfs.lt.1.e-5) go to 654		1739
	nf=nf+1		1740
	mm=mm+1		1741
	write(a79(mm),9651) nf,xfs		1742
654	continue		1743
	if(xcs.lt.1.e-5) go to 655		1744
	if(mm.eq.0) nf=nf+1		1745
	mm=mm+1		1746
	if(mm.eq.1) write(a79(mm),9652) nf,xcs		1747
	if(mm.eq.2) write(a79(mm),9653) xcs		1748
c		Iron character	1749
655	continue		1750
	if(qiron.eq.0.0) go to 658		1751
	if(lr(7).eq.1n) go to 657		1752
	nf=nf+1		1753
	mm=mm+1		1754
	write(a79(mm),9655) nf		1755
657	continue		1756
	if(lr(15).eq.ly) go to 658		1757
	if(xiron.gt.1.1) go to 658		1758
	nf=nf+1		1759
	mm=mm+1		1760
	write(a79(mm),9656) nf,xiron		1761
c		Treatment of LOI in chemical analysis	1762
658	continue		1763
	if(pw(13).eq.0.0) go to 660		1764
	nf=nf+1		1765
	mm=mm+1		1766
	if(kloi.ne.0) write(a79(mm),9657) nf		1767
	if(kloi.eq.0) write(a79(mm),9658) nf		1768
c		Water character in structural formula	1769
660	continue		1770
	if(lr(15).eq.ly) go to 690		1771
	if(qwat.eq.0.0) go to 665		1772
	nf=nf+1		1773
	mm=mm+1		1774
	write(a79(mm),9661) nf		1775
	do 661 i=1,3		1776
	ii=i+10		1777

	if(pm(ii).eq.0.0) go to 661	1778
	mm=mm+1	1779
	write(a79(mm),9663) lox(ii),lwat(iww(i))	1780
661	continue	1781
	if(iwat.eq.0) go to 664	1782
	if(kkwat.eq.0) go to 664	1783
	mm=mm+1	1784
	if(iwat.eq.1) write(a79(mm),9664)	1785
	if(iwat.eq.2) write(a79(mm),9665)	1786
664	continue	1787
c	Components deleted from structural formula	1788
665	continue	1789
	if(hdel.eq.0) go to 670	1790
	nq=0	1791
	do 666 i=1,nox	1792
	if(i.lt.30) a5(i)=a5b	1793
	if(ndel(i).eq.0) go to 666	1794
	if(xm(i).eq.0.0) go to 666	1795
	nq=nq+1	1796
	a5(nq)=lox(i)	1797
666	continue	1798
	if(nq.eq.0) go to 670	1799
	nf=nf+1	1800
	mm=mm+1	1801
	write(a79(mm),9668) nf,(a5(k),k=1,5)	1802
	if(nq.lt.6) go to 670	1803
	mm=mm+1	1804
	write(a79(mm),9669) (a5(k),k=6,17)	1805
	if(nq.lt.18) go to 670	1806
	mm=mm+1	1807
	write(a79(mm),9669) (a5(k),k=18,29)	1808
670	continue	1809
c	Tetrahedral sites per formula unit	1810
	if(nl.eq.0) go to 672	1811
	nf=nf+1	1812
	mm=mm+1	1813
	write(a79(mm),9673) nf,ntet	1814
c	(OH,F,Cl) sites per formula unit	1815
672	continue	1816
	if(kkwat.eq.0) go to 675	1817
	mm=mm+1	1818
	if(nl.eq.0) nf=nf+1	1819
	if(nl.ne.0) write(a79(mm),9674) nwat	1820
	if(nl.eq.0) write(a79(mm),9678) nf,nwat	1821
675	continue	1822
c	Anion character in structural formula	1823
	n=0	1824
	if(oxeq.ne.0.0) n=n+1	1825
	if(anno.ne.0.0) n=n+1	1826
	if(n.eq.0) go to 680	1827
	if(anno.eq.0.0) go to 678	1828
	mm=mm+1	1829
	nf=nf+1	1830

	write(a79(mm),9675) nf,anno	1831
678	continue	1832
	if(oxeq.eq.0.0) go to 680	1833
	if(n.eq.1) nf=nf+1	1834
	mm=mm+1	1835
	if(n.eq.1) write(a79(mm),9676) nf,oxeq	1836
	if(n.eq.2) write(a79(mm),9677) oxeq	1837
c	Cation site assignment in clay formula	1838
680	nzzz=0	1839
	nitet=0	1840
	kotet=0	1841
	if(nl.eq.0) go to 690	1842
	nitet=0	1843
	i=0	1844
	k=-1	1845
6801	k=k+2	1846
	if(k.eq.2.or.k.gt.3) k=0	1847
6802	i=i+1	1848
	if(i.gt.7) go to 6803	1849
	if(nt(i).eq.k) go to 6801	1850
	if(nt(i).eq.0) go to 6802	1851
	if(xc(nt(i)).eq.0.0) go to 6802	1852
	nitet=1	1853
6803	continue	1854
	if(k.ne.0) nitet=1	1855
	do 6805 i=1,nox	1856
	if(xc(i).eq.0.0) go to 6805	1857
	if(nz(i).ne.0) nzzz=1	1858
6805	continue	1859
	do 6807 i=1,7	1860
	if(nt(i).eq.1) go to 6806	1861
	if(xc(nt(i)).gt.0.0.and.ny(nt(i)).eq.0) kotet=1	1862
	go to 6807	1863
6806	continue	1864
	if(ny(1).ne.0.and.xc(1).gt.0.0) kotet=1	1865
6807	continue	1866
	if(xc(1).gt.0.0.and.ny(1).ne.0) kotet=1	1867
6808	nzzt=nzzz+nitet+kotet	1868
	if(nzzt.eq.0) lras=lraa(2)	1869
	if(nzzt.ne.0) lras=lraa(1)	1870
	ii=1	1871
	iii=2	1872
	nf=nf+1	1873
	mm=mm+1	1874
	write(a79(mm),9681) nf,lt7,lcas,lras	1875
	if(nzzt.eq.0) go to 690	1876
	nq=0	1877
	if(kotet.eq.0) go to 683	1878
	do 6823 i=1,7	1879
	a4(i)=a4b	1880
	if(xc(nt(i)).eq.0.0.or.ny(nt(i)).ne.0) go to 6823	1881
	nq=nq+1	1882
	a4(nq)=1c(nt(i))	1883

6823	continue	1884
	mm=mm+1	1885
	if(nq.eq.0) a4(1)=none	1886
	write(a79(mm),9684) (a4(i),i=1,7)	1887
683	nq=0	1888
	if(itet.eq.0) go to 684	1889
	do 681 i=1,7	1890
	a4(i)=a4b	1891
	if(nt(i).eq.0) go to 681	1892
	if(xc(nt(i)).eq.0.0) go to 681	1893
	nq=nq+1	1894
	a4(nq)=1c(nt(i))	1895
681	continue	1896
682	mm=mm+1	1897
	write(a79(mm),9685) (a4(k),k=1,7)	1898
684	nq=0	1899
	if(nzzz.eq.0) go to 690	1900
	do 685 i=1,nox	1901
	if(i.lt.31) a4(i)=a4b	1902
	if(xc(i).eq.0.0) go to 685	1903
	if(nz(i).eq.0) go to 685	1904
	if(ny(i).ne.ii) go to 685	1905
	nq=nq+1	1906
	a4(nq)=1c(i)	1907
685	continue	1908
	if(nq.eq.0) go to 687	1909
	nqq=11	1910
	nqqq=20	1911
	mm=mm+1	1912
	write(a79(mm),9682) ltaa(iii),(a4(k),k=1,10)	1913
686	continue	1914
	if(nq.lt.nqq) go to 687	1915
	mm=mm+1	1916
	write(a79(mm),9683) (a4(k),k=nqq,nqqq)	1917
	nqq=nqq+10	1918
	nqqq=nqqq+10	1919
	if(nqq.lt.22) go to 686	1920
687	ii=ii+1	1921
	iii=iii+1	1922
	if(iii.eq.3.and.nl.gt.7) iii=4	1923
	if(ii.lt.3) go to 684	1924
	mm=mm+1	1925
	if(ibor.eq.1.and.isil.eq.0) write(a79(mm),9686)	1926
	if(ibor.eq.0.and.isil.eq.1) write(a79(mm),9687)	1927
	if(ibor.eq.1.and.isil.eq.1) write(a79(mm),9688)	1928
690	continue	1929
c		1930

Printing out sample analysis and structural formula

```

c***** 1931
c          PRINT-OUT OF SAMPLE ANALYSIS 1932
c***** 1933
c 1934
      if(lr(11).eq.1n.and.lr(12).eq.1n) lfile=inpt 1935
      if(kal.ne.0) go to 705 1936
701 write(1,951) 1937
      read(1,991,err=701) nouts 1938
      if(nouts.lt.0.or.nouts.gt.2) go to 701 1939
      if(nouts.lt.1) go to 705 1940
      do 702 i=1,999 1941
      write(lform,8541) i 1942
      inquire(file=lform,exist=mol) 1943
      if(mol) go to 702 1944
      open(15,status='new',file=lform,err=702) 1945
      write(1,8571) lform 1946
      go to 703 1947
702 continue 1948
703 write(1,8575) 1949
      call yesno(1a,nerr) 1950
      if(nerr.ne.0) go to 703 1951
      ncarr=0 1952
      if(1a.eq.1y) ncarr=1 1953
705 m=1 1954
      if(nouts.eq.1) m=15 1955
      if(nono.eq.0) go to 706 1956
      if(nono.eq.2) nono=0 1957
      if(m.eq.1) write(1,9511) 1958
      if(m.eq.1) read(1,990,err=705) 1a 1959
706 kal=1 1960
      kout=maxx+mm+5 1961
      if(m.eq.15.and.ncarr.eq.1) write(m,9512) 1962
      if(m.eq.1) ii=66 1963
      if(m.eq.15) ii=55 1964
      if(m.eq.15.and.ncarr.eq.1) ii=ii-2 1965
      nskip=ii-kout 1966
      if(nskip.le.0) go to 710 1967
      do 707 i=1,33 1968
      k=2*i 1969
      if(k.ge.nskip) go to 708 1970
707 continue 1971
708 kpre=i 1972
      kpost=i 1973
      if(k.gt.nskip) kpre=kpre-1 1974
      if(kpre.lt.1) go to 710 1975
      do 709 i=1,kpre 1976
709 write(m,997) 1977
710 continue 1978
      write(m,952) ltit,ldes 1979
712 continue 1980
      do 715 i=1,maxx 1981

```

write(m,9231) ha(i),hf(i)	1982
715 continue	1983
if(lfile.ne.inpt) write(m,960) nal,lfile	1984
if(lfile.eq.inpt) write(m,961) nal	1985
do 720 i=1,mm	1986
720 write(m,9235) a79(i)	1987
if(m.eq.15.and.ncarr.eq.1) go to 740	1988
if(nskip.le.0) go to 740	1989
do 730 i=1,kpost	1990
730 write(m,997)	1991
740 continue	1992
if(nouts.ne.2.or.m.ne.1.or.nono.eq.0) go to 745	1993
write(1,8577)	1994
call yesno(1a,nerr)	1995
if(nerr.ne.0) go to 740	1996
if(1a.eq.1n) go to 755	1997
745 continue	1998
if(nouts.lt.2.or.m.eq.15) go to 755	1999
m=15	2000
go to 706	2001
c	2002

Program continuation or termination

c*****	2003
c PROGRAM CONTINUATION OR TERMINATION	2004
c*****	2005
c	2006
755 continue	2007
nq=0	2008
if(lr(14).eq.ly.or.lr(15).eq.ly) go to 770	2009
if(nono.eq.0) go to 770	2010
go to 765	2011
761 write(1,989)	2012
762 nq=8	2013
if(lr(11).eq.ly) close(12)	2014
if(lr(12).eq.ly) close(13)	2015
lr(11)=1n	2016
go to 774	2017
c Other parameters for this analysis?	2018
765 write(1,985)	2019
call yesno(1a,nerr)	2020
if(nerr.ne.0) go to 765	2021
if(1a.eq.1n) go to 770	2022
nq=1	2023
c Change any component abundances in analysis?	2024
767 write(1,988)	2025
call yesno(1a,nerr)	2026
if(nerr.ne.0) go to 767	2027
if(1a.eq.ly) nq=2	2028
go to 780	2029
770 continue	2030
if(lr(12).eq.ly) nq=4	2031

if(lr(12).eq.ly) go to 780	2032
c	Another analysis in this group?
772 write(1,986)	2033
call yesno(1a,nerr)	2034
if(nerr.ne.0) go to 772	2035
if(1a.eq.1n) nq=8	2036
if(1a.eq.ly) go to 780	2037
c	Another group of analyses?
774 write(1,984)	2038
call yesno(1a,nerr)	2039
if(nerr.ne.0) go to 774	2040
if(1a.eq.1n) nq=16	2041
780 continue	2042
m=1	2043
nofile = 0	2044
if(nq.eq.2) nofile = 1	2045
if(nq.eq.0) go to 200	2046
if(nq.eq.1) go to 305	2047
if(nq.eq.2) go to 238	2048
if(nq.eq.4) go to 180	2049
if(nq.gt.7.and.lr(11).eq.ly) go to 270	2050
if(nq.eq.8) go to 100	2051
798 continue	2052
if(nouts.gt.0) close(15)	2053
799 continue	2054
stop	2055
c	2056
	2057
	2058

Format statements

c*****	2059
c	FORMAT STATEMENTS
c*****	2060
c	2061
8001 format(18x,'STRUCTURAL FORMULA SPECIFICATIONS')	2062
8002 format('1 - ',a40,':')	2063
8003 format(6x,a5,a34)	2064
8004 format(4x,'Excess OH recast as H3O+'.')	2065
8005 format(4x,'Excess OH recast as H2O+'.')	2066
8006 format(a28,f5.3, '.')	2067
8007 format('2 - ',a35,a35)	2068
8008 format('3 - ',a47,'None.')	2069
8009 format('3 - ',a47,4(a5,1x),a5)	2070
8010 format(4x,15a5)	2071
8012 format('4 - ',a29,a40)	2072
8013 format(6x,a37,i3, '.')	2073
8014 format(6x,a36,i2, '.')	2074
8015 format('6 - ',a25,a29,a20)	2075
8017 format(/,5x,'Enter ID (1-dig. int.) of specification to modify:',	2076
-/,10x,['0" = All specifications satisfactory.'],'/,10x,	2077
-['7" = Print summary of specifications.])'	2078
8018 format(6x,a22,10(a4,1x))	2079
8019 format(28x,10(a4,1x))	2080
	2081

8020	format(6x,a30,i2,'.')	2082
8021	format(6x,a49,i2,'.')	2083
8022	format(6x,a32,i2,'.')	2084
8025	format('5 - Parameters of structural formula:')	2085
8026	format(6x,a34,i2,'.')	2086
8028	format('1 - Water options',12x,'3 - Deletions',15x,	2087
	- '5 - Formula parameters',/, '2 - Iron redox',15x,	2088
	- '4 - Formula type')	2089
8029	format('1 - Water options',12x,'3 - Deletions',15x,	2090
	- '5 - Formula parameters',/, '2 - Iron redox',15x,	2091
	- '4 - Formula type',12x,'6 - Site assignments')	2092
8030	format(6x,'Must be tetrahedral: ',7(a4,1x))	2093
8031	format('4 - ',a29,'General (cation sites undesigned).')	2094
8032	format('4 - ',a29,'General (cation sites designated).')	2095
8210	format('Elect character ID (1-dig. int.) for ',a5)	2096
8211	format(5x,i1,' - ',a34)	2097
8212	format('Disposition of excess OH (enter 1-dig. int. ID):'	2098
	-/,5x,'0" - Retain as OH.',/,5x,	2099
	- "1" - Recast as H3O+ ion in interlayer sites.',/,5x,	2100
	- "2" - Recast as H2O (hydration waters).')	2101
8213	format('Enter # of (OH,F,Cl) sites in formula (2-dig. int.):')	2102
8220	format('Iron redox ratio other than in analysis? (y/n):')	2103
8221	format('Enter value for Fe2+/[Fe2+ + Fe3+] atom ratio (0.0-1.0):')	2104
8222	format(a28,f6.4,5x,'OK? (y/n):')	2105
8230	format('Select one option (1-dig. int.):',/,5x,	2106
	- '1 - Remove all previous deletions.',/,5x,	2107
	- '2 - Remove previous deletions and assign new ones.',/,5x,	2108
	- '3 - Retain previous deletions but modify the list.')	2109
8231	format('Selection of specific individual modifications '	2110
	-, '(enter 1-dig. int.)'	2111
	-/,5x,'0" = From list of only components in analysis.',/,5x,	2112
	- "1" = From list of all components in program.')	2113
8232	format(8(' '))	2114
8233	format(i2.2,1x,a5)	2115
8234	format(8(a1,a8,1x))	2116
8235	format('Enter ID (2-dig. int.) for components to be deleted:')	2117
8236	format(3x,'[If starred ("*") ID entered; previous deletion ',	2118
	- 'instruction removed.]')	2119
8237	format(20('-- '))	2120
	824 format('Clay-mineral formula type? Enter 2-dig. int. ID:',	2121
	-10(/,5x,i2,2x,a40))	2122
8240	format('Type of formula (enter 1-dig. int.):',/,5x,	2123
	- "1" = General silicate-mineral structural formula.',/,5x,	2124
	- "2" = Phyllosilicate (clay-mineral) formula.')	2125
8241	format('Enter type (1-dig. int.) of phyllosilicate formula:')	2126
8242	format('Component distribution into formula sites? (y/n):')	2127
8243	format('Mineral dioctahedral ("n"=trioctahedral)? (y/n):')	2128
8244	format('Tet-oct ("0") or all ("1") cations? ',	2129
	- '(Enter 1-dig. int.):')	2130
8250	format('Method for calculating structural formula (enter ',	2131
	- '1-dig. int.):',/,5x,'0" - Oxygen equivalents (anion ',	2132
	- 'components excluded) per formula unit.'	2133
	-/,5x,'1" - Number of anion sites (0,OH,F,Cl) per formula unit.'	2134

```

    -/,5x,"2" - Number of cations per formula unit.') 2135
8251 format('Enter (2-dig. int.) new value:') 2136
8252 format('Double the formula charge/sites? (y/n):') 2137
8253 format(a37,i2) 2138
8254 format(a30,i2) 2139
8255 format(a49,i2) 2140
8256 format(a36,i2) 2141
8257 format(a34,i2) 2142
8258 format(a32,i2) 2143
8260 format(10x,'STRUCTURAL FORMULA PARAMETERS') 2144
8270 format('Site assignment change desired (enter 1-dig. int.):', 2145
    -4(/,5x,i1,lx,a29)) 2146
8271 format('Any reassignment of specific cations? (y/n):') 2147
8272 format(' ') 2148
8273 format(a1,i2.2,lx,a4) 2149
8274 format(/,'LISTING OF CATION SITE ASSIGNMENTS',/, 2150
    -'(Tetrahedral listing is order of filling tetrahedral sites)') 2151
8275 format(20x,a8,8x,a8,8x,a8) 2152
8276 format('* Specifically user-reassigned to this site.') 2153
8277 format('Identify (1-dig. int.) type of modification desired:' 2154
    -/,5x,"0" = No change.',/,5x, 2155
    -"1" = Reassign to octahedral sites.',/,5x, 2156
    -"2" = Reassign to interlayer sites.',/,5x, 2157
    -"3" = Order of filling tetrahedral sites.') 2158
8279 format(5x,"4" = Remove individual non-default site assignments.') 2159
8278 format('Retain non-default cation site assignments? (y/n)', 2160
    -/,2x,['No" returns all cations to default assignment.])') 2161
8280 format('Enter ID (2-dig. int.) for reassignment to ',all, 2162
    -/,2x,'(An entry of "00" signifies no further changes)') 2163
8281 format('Order of filling tetrahedral sites') 2164
8282 format(5x,i1,'. ',i2.2,lx,a4) 2165
8283 format('Feasible tetrahedral cation ID''s') 2166
8284 format(5x,i2.2,lx,a4) 2167
8285 format('List ID''s (2-dig. int.) for order of filling tet. ' 2168
    -,'sites:',/,7('-- ')) 2169
8286 format(7(i2,lx)) 2170
8287 format('Enter ID (2-dig. int.) for return to default site ', 2171
    -'assignment.') 2172
8288 format(/,20x,2a16,'Big cations') 2173
8289 format(/,20x,3a16) 2174
8290 format('Same parameters for all samples in group? (y/n):') 2175
8293 format('Other ions that must be tetrahedral? (y/n):') 2176
8294 format('Enter ALL cation ID''s (2-dig. int.) that must ', 2177
    -'be tetrahedral:',/,7('-- ')) 2178
850 format(a1,a79) 2179
851 format(' ', 'Analysis on file? (y/n):') 2180
852 format(' ', 'A file of new analyses to be created? (y/n):') 2181
853 format(i3.3,2x,a75) 2182
8531 format(' ', 'Data entry ONLY (no output)? (y/n):') 2183
854 format('CLAY.',i3.3) 2184
8541 format('FORM.',i3.3) 2185
855 format(' ', 'Enter title of this group of analyses ', 2186
    -'(up to 75-char. string)') 2187

```

856	format(a8)	2188
857	format(' ', 'New file named: ', a8)	2189
8571	format(' ', 'Output file named: ', a8)	2190
8575	format('Output file with FORTRAN carriage control? (y/n):')	2191
8577	format('Into the output file? (y/n):')	2192
858	format(' ', 'Enter 3-dig. int. to identify desired CLAY.nnn file:',	2193
	-, 3x, '[An entry of 000 lists titles of CLAY.nnn files.]')	2194
859	format(' ', a8, ' NOT FOUND. Try again (y/n):')	2195
861	format(' ', 'A listing of samples in file? (y/n):')	2196
863	format(' ', 'FILE SAMPLE # ', i2.2, /, a80)	2197
864	format(' ', 'Want just designated samples in file? (y/n):')	2198
865	format(' ', 'Enter up to 10 integers (2-digit) ',	2199
	- 'designating samples desired: ', /, 10('--', 1x))	2200
866	format(10(i2, 1x))	2201
8662	format(20(i2, 1x))	2202
869	format(' ', 10x, 'SAMPLE DESCRIPTIONS: FILE ', a8, /)	2203
872	format(a1, 1x, 6(1x, i2.2, 1x, e9.4))	2204
873	format(a1, 1x, a5, 1x, a4, f9.5, 1x, 3i1)	2205
874	format(a1, 1x, a1)	2206
875	format('Disposition of LOI (loss-on-ignition)? (1-dig. int.)'	2207
	-, /, 5x, '"0" - LOI omitted from all calculations.', /, 5x,	2208
	- '"1" LOI calculated as molar H2O.')	2209
876	format(6x, 'LOI (loss-on-ignition) omitted from all calculations.')	2210
877	format(6x, 'LOI (loss-on-ignition) recast as molar H2O.')	2211
881	format(a5, a4, f9.5, 3i1)	2212
891	format('Tetrahedral Octahedral Interlayer ',	2213
	- 'Anions/waters Layer charge ')	2214
892	format(5a16)	2215
901	format(' ', 'Enter oxide wt. %:')	2216
902	format(' ', 2x, a5, ':')	2217
903	format(f9.6)	2218
904	format(' ', 'Any other oxides? (y/n):')	2219
905	format(' ', 'Enter oxide ID# and wt. %:')	2220
906	format(' ', 5(4x, i2.2, 2x, a5))	2221
907	format(2(' '), 1x, 9(' '))	2222
908	format(i2, 1x, f9.6)	2223
909	format(' ', 'Any other (unlisted) components? (y/n):')	2224
911	format(' ', 'Molecular formula (up to 5-char. string):')	2225
912	format(a5)	2226
913	format(' ', 'Formula atom (up to 4-char. string):')	2227
914	format(a4)	2228
915	format(' ', 'Molecular weight (with dec. pt.):')	2229
916	format(' ', 'Number of formula atoms per molecule (1 int.):')	2230
917	format(' ', 'Negative charges per molecule (1 int.):')	2231
918	format(' ', 'Site in clay structure (1 int. code ID)', /, 4x,	2232
	a '0=tet.', 4x, '1=oct.', 4x, '2=interlayer', 4x, '3=anion', 4x,	2233
	- '4=molecule:')	2234
919	format(' ', 'Is total Fe expressed as single oxide? (y/n):')	2235
920	format(' ', 'Sample description (up to 80-char. string):')	2236
9200	format(5x, 'CHEMICAL ANALYSIS', 6x)	2237
9201	format(1x, 8x, a8, 2x, a8, 1x)	2238
9203	format(1x, a5, 3x, f8.4, 2x, f8.4, 1x)	2239
9204	format(1x, a5, 3x, f8.4, 10x, 1x)	2240

9205	format(a8,1x,f8.4,2x,f8.4,1x)	2241
9206	format(1x,'O=F+Cl',2x,f8.4,10x,1x)	2242
9207	format('TOTAL',4x,f8.4,10x,1x)	2243
921	format(' ',//,10x,'CHECK OF DATA ENTRY')	2244
9210	format(14x,a32,i2,3x)	2245
9211	format(16x,'GENERALIZED STRUCTURAL FORMULA',5x)	2246
9212	format(11x,a38,i2)	2247
9213	format(27x,a4,2x,f8.5,10x)	2248
9214	format(33x,a8,10x)	2249
9215	format(17x,'Total cations',3x,f8.5,10x)	2250
9216	format(11x,a38,i2)	2251
9217	format(14x,a32,i2,3x)	2252
922	format(' ',24x,i2.2,1x,a5,2x,f8.5)	2253
9220	format(13x,'SILICATE MINERAL STRUCTURAL FORMULA',3x)	2254
9221	format(15x,'CLAY-MINERAL STRUCTURAL FORMULA',5x)	2255
9222	format(10x,a41)	2256
9223	format(10x,a39,i2)	2257
9224	format(13x,a33,i2,3x)	2258
9225	format(10x,a16,2x,a4,1x,f8.5,10x)	2259
9226	format(25x,a8,10x,f8.5)	2260
9227	format(13x,a26,f9.5,3x)	2261
9228	format(11x,a37,i2,1x)	2262
9229	format(14x,a31,i2,4x)	2263
923	format(' ', 'Analysis entry OK? (y/n):')	2264
9231	format(' ',a28,a51)	2265
9232	format(28(' '))	2266
9233	format(51(' '))	2267
9235	format(' ',a79)	2268
924	format(' ', 'Enter component ID (2-dig.int.) and corrected value',	2269
	a' (dec. pt.)',/,5x,'[ID: 00 = all OK; 98 = list ',	2270
	a'all ID's; 99 = re-enter whole analysis]')	2271
925	format(' ',/,a80,/,5x,'OK? (y/n):')	2272
926	format(' ', 'UNLISTED ANALYSIS MOLECULE CONSTANTS',/5x,	2273
	a'Molecule formula = ',a5,10x,'Atom/radical = ',a4,5x,	2274
	a/,5x,'Molecular weight = ',f9.5,10x,'Formula atoms in ',	2275
	a'molecule = ',i1,/,5x,'Molecule negative charge = ',i1,	2276
	a10x,'Assigned layer site = ',a16)	2277
944	format(' ', '0 = No additional changes',14x,'1 = Reassign to',	2278
	a' oct. sites',/,1x,'2 = Reassign to interlayer sites',7x,	2279
	a'3 = Revise order of filling tet. sites',/,	2280
	a'Enter desired integer code:')	2281
951	format(' ', 'Destination of output [one-digit entry]:',/,5x,	2282
	- '0 - Terminal only',/,5x,'1 - Data file for line printer',	2283
	-/,5x,'2 - Both of above')	2284
9511	format(' ',23x,'SET PAPER IN CARRAIGE AND RETURN')	2285
9512	format('1',10x)	2286
952	format(' ',4x,a75,/,a80,/,80('-'))	2287
955	format('Cannot compute H30+!! IER=',i2,'. Enter new parameters.')	2288
960	format(' ',53x,'Sample ',i2.2,' in file ',a8,/,80('-'))	2289
961	format(' ',46x,'Sample ',i2.2,' from direct data entry.'	2290
	-/,80('-'))	2291
9651	format(i1,'. Mol % SiO2 includes ',f7.4,' mol % SiF4.',37x)	2292
9652	format(i1,'. Mol % SiO2 includes ',f7.4,' mol % SiCl2.',36x)	2293

9653 format(3x,'Mol % SiO2 includes ',f7.4,' mol % SiCl2.',36x)	2294
9655 format(i1,'. Fe oxide in analysis is total Fe2+ + Fe3+.',32x)	2295
9656 format(i1,'. Iron redox in structural formula set at ',	2296
-'Fe2+/[Fe2+ + Fe3+] = ',f6.4,'.',8x)	2297
9657 format(i1,'. LOI (loss-on-ignition) calculated as molar H2O.',29x)	2298
9658 format(i1,'. LOI (loss-on-ignition) omitted from all calculations.	2299
-',23x)	2300
9659 format(i1,'. Chemical analysis altered from original data.')	2301
9661 format(i1,'. Water character in structural formula set at:',31x)	2302
9663 format(5x,a5,a34,35x)	2303
9664 format(3x,' Excess OH sites recast as interlayer H3O+ ion.',36x)	2304
9665 format(3x,' Excess OH sites recast as molecular H2O.',42x)	2305
9668 format(i1,'. Components deleted from strucutral formula: ',	2306
-5(a5,1x),2x)	2307
9669 format(5x,12(a5,1x),2x)	2308
9673 format(i1,'. Number of tetrahedral sites per formula set at '	2309
-,i2,'.',26x)	2310
9674 format(3x,'Number of (OH,F,Cl) sites per formula set at '	2311
-,i2,'.',28x)	2312
9675 format(i1,'. Number of anions per formula unit =',f9.5,'.')	2313
9676 format(i1,'. Number of oxygen equivalents per formula unit =',	2314
-f9.5,'.')	2315
9677 format(3x,'Number of oxygen equivalents per formula unit =',	2316
-f9.5,'.')	2317
9678 format(i1,'. Number of (OH,F,Cl) sites per formula unit set at '	2318
-,i2,'.')	2319
9681 format(i1,'. ',a25,a29,a20,2x)	2320
9682 format(5x,a22,10(a4,1x),2x)	2321
9683 format(27x,10(a4,1x),2x)	2322
9684 format(5x,'Must occupy tetrahedral sites: ',	2323
-7(a4,1x),7x)	2324
9685 format(5x,'Order of filling tetrahedral sites: ',	2325
-7(a4,1x),2x)	2326
9686 format(3x,'Default reassignment of B to unique sites.')	2327
9687 format(3x,'Default reassignment of Si to unique sites.')	2328
9688 format(3x,'Default reassignment of Si and B to unique sites.')	2329
984 format(' ', 'Another group (file) of analyses to enter? (y/n):')	2330
985 format(' ', '////,1x, 'Other parameters for this analysis? (y/n):')	2331
986 format(' ', 'Another analysis in this group to enter? (y/n):')	2332
9871 format(' ', 'Only wt.% and mol.% in output (no struct. form.)',	2333
-'? (y/n):')	2334
988 format(' ', 'Any modification of chemical analysis? (y/n):')	2335
989 format(' ', 20x, 'CANNOT READ DATA FILE!',/,16x,	2336
-'This file abandoned and closed.')	2337
990 format(a1)	2338
991 format(i1)	2339
992 format(i2)	2340
9921 format(i3)	2341
994 format(' ', 'OK? (y/n):')	2342
997 format(' ', 4x)	2343
998 format(a80)	2344
999 format(a75)	2345
end	2346

c 2347
c***** 2348

Subroutine YESNO

c*****	0001
c SUBROUTINE YESNO	0002
c*****	0003
c	0004
c Allows main program to accept either one of the "y" and "n" pair	0005
c or one of the "Y" and "N" pair in response to a yes-no question	0006
c in the host program; program also flags an error for any entry	0007
c other than "y", "Y", "n", or "N", and so informs the host program.	0008
c	0009
subroutine yesno(lam,nerr)	0010
character*1 lam,ly,ln,lyy,lnn	0011
data ly/'y'/,ln/'n'/,lyy/'Y'/,lnn/'N'/	0012
nerr=0	0013
read(1,990,err=190) lam	0014
if(lam.eq.lyy.or.lam.eq.lnn) go to 150	0015
if(lam.ne.ly.and.lam.ne.ln) go to 190	0016
go to 200	0017
150 continue	0018
if(lam.eq.lyy) lam=ly	0019
if(lam.eq.lnn) lam=ln	0020
go to 200	0021
190 nerr=1	0022
200 return	0023
990 format(a1)	0024
end	0025
c	0026
c*****	0027

Subroutine INSERT

c*****	0001
c SUBROUTINE INSERT	0002
c*****	0003
c	0004
c Inserts non-bank component into sequence for listing components	0005
c in output	0006
c	0007
subroutine insert(ny,nc,nn,io,n)	0008
dimension ny(50),nc(50),nn(50),io(50)	0009
if(ny(n).lt.4) na=nn(n)/nc(n)	0010
n1=n-1	0011
kk=0	0012
do 110 k=1,n1	0013
if(ny(n).gt.2) go to 105	0014
if(ny(io(k)).gt.2) go to 110	0015
nb=nn(io(k))/nc(io(k))	0016

if(nb.lt.na) go to 110	0017
kk=k	0018
go to 110	0019
105 continue	0020
if(io(k).eq.11.or.io(k).eq.12) go to 110	0021
if(ny(n).eq.ny(io(k))) kk=k	0022
if(ny(n).eq.4) kk=k-1	0023
110 continue	0024
kkk=kk+1	0025
do 210 k=n1,kkk,-1	0026
kq=k+1	0027
210 io(kq)=io(k)	0028
io(kkk)=n	0029
return	0030
end	0031
c	0032
c*****	0033

Subroutine LISTCLAYS

c*****	0001
c SUBROUTINE LISTCLAYS	0002
c*****	0003
c	0004
c Updates CLAY.ID and prints out listing clay analysis data files.	0005
c	0006
subroutine listclays	0007
character*80 180	0008
character*75 175	0009
character*8 18	0010
logical*4 mol	0011
open(17,file='CLAY.ID',status='old',err=890)	0012
read(17,901,err=890,end=890) 180	0013
open(18,status='scratch',err=890)	0014
write(18,901) 180	0015
do 100 i=1,999	0016
write(18,903) i	0017
inquire(file=18,exist=mol)	0018
if(mol) go to 90	0019
go to 100	0020
90 continue	0021
open(19,file=18,status='old',err=95)	0022
read(19,905,end=95,err=95) 175	0023
write(18,902) i,175	0024
write(1,902) i,175	0025
close(19)	0026
go to 100	0027
95 continue	0028
close(19,status='delete')	0029
100 continue	0030
110 continue	0031
close(17,status='delete')	0032

rewind(18)	0033
open(16,file='CLAY.ID',status='new',err=810)	0034
200 continue	0035
do 210 i=1,1000	0036
read(18,901,end=220) 180	0037
write(16,901) 180	0038
210 continue	0039
220 continue	0040
close(16)	0041
close(18)	0042
go to 890	0043
800 continue	0044
write(1,921)	0045
close(17)	0046
close(18)	0047
go to 890	0048
810 continue	0049
close(16)	0050
il=2	0051
811 continue	0052
if(il.gt.9) go to 813	0053
do 812 i=il,9	0054
il=il+1	0055
write(18,904) i	0056
inquire(file=18,exist=mol)	0057
if(mol) go to 812	0058
go to 814	0059
812 continue	0060
813 continue	0061
write(1,923)	0062
close(18)	0063
go to 890	0064
814 continue	0065
open(16,file=18,status='new',err=815)	0066
write(1,922) 18	0067
go to 200	0068
815 continue	0069
close(16)	0070
go to 811	0071
890 continue	0072
return	0073
901 format(a80)	0074
902 format(i3.3,2x,a75)	0075
903 format('CLAY.',i3.3)	0076
904 format('CLAY.ID',il)	0077
905 format(a75)	0078
921 format('Cannot read present CLAY.ID; file edit required!')	0079
922 format('CLAY.ID renamed ',a8)	0080
923 format('CLAY.ID destroyed - SORRY')	0081
end	0082
c	0083
c*****	0084

Subroutines ARRAY and DGELG

```

C***** 0001
C          SUBROUTINE ARRAY 0002
C***** 0003
C 0004
C    IBM software for converting two-dimensional to linear arrays. 0005
C 0006
C      subroutine array(mode,i,j,n,m,s,d) 0007
C      dimension s(1),d(1) 0008
C      ni=n-i 0009
C      if(mode-1)100,100,120 0010
100  ij=i*j+1 0011
C      nm=n*j+1 0012
C      do 110 k=1,j 0013
C      nm=nm-ni 0014
C      do 110 l=1,i 0015
C      ij=ij-1 0016
C      nm=nm-1 0017
C      d(nm)=s(ij) 0018
110  continue 0019
C      go to 140 0020
120  ij=0 0021
C      nm=0 0022
C      do 130 k=1,j 0023
C      do 125 l=1,i 0024
C      ij=ij+1 0025
C      nm=nm+1 0026
C      s(ij)=d(nm) 0027
125  continue 0028
C      nm=nm+ni 0029
130  continue 0030
140  return 0031
C      end 0032
C 0033
C***** 0034

C***** 0035
C          SUBROUTINE DGELG 0036
C***** 0037
C 0038
C    IBM software for solving simultaneous linear equations. 0039
C 0040
C      subroutine dgelg(r,a,m,n,eps,ier) 0041
C      dimension a(1),r(1) 0042
C      double precision r,a,piv,tb,tol,pivi 0043
C      if(m)23,23,1 0044
1  ier=0 0045
C      piv=0.d0 0046
C      mm=m*m 0047
C      nm=n*m 0048
C      do 3 l=1,mm 0049
C      tb=dabs(a(1)) 0050

```

if(tb-piv)3,3,2	0051
2 piv=tb	0052
i=1	0053
3 continue	0054
tol=eps*piv	0055
lst=1	0056
do 17 k=1,m	0057
if(piv)23,23,4	0058
4 if(ier)7,5,7	0059
5 if(piv-tol)6,6,7	0060
6 ier=k-1	0061
7 pivi=1.d0/a(i)	0062
j=(i-1)/m	0063
i=i-j*m-k	0064
j=j+1-k	0065
do 8 l=k,nm,m	0066
ll=1+i	0067
tb=pivi*r(ll)	0068
r(ll)=r(l)	0069
r(l)=tb	0070
8 continue	0071
if(k-m)9,18,18	0072
9 lend=lst+m-k	0073
if(j)12,12,10	0074
10 ii=j*m	0075
do 11 l=lst,lend	0076
tb=a(l)	0077
ll=1+ii	0078
a(l)=a(ll)	0079
a(ll)=tb	0080
11 continue	0081
12 do 13 l=lst,mm,m	0082
ll=1+i	0083
tb=pivi*a(ll)	0084
a(ll)=a(l)	0085
a(l)=tb	0086
13 continue	0087
a(lst)=j	0088
piv=0.d0	0089
lst=lst+1	0090
j=0	0091
do 16 ii=lst,lend	0092
pivi=-a(ii)	0093
ist=ii+m	0094
j=j+1	0095
do 15 l=ist,mm,m	0096
ll=1-j	0097
a(l)=a(l)+pivi*a(ll)	0098
tb=dabs(a(l))	0099
if(tb-piv)15,15,14	0100
14 piv=tb	0101
i=1	0102
15 continue	0103

```

do 16 l=k,nm,m                                0104
  ll=l+j                                        0105
  r(ll)=r(ll)+pivi*r(l)                        0106
16 continue                                    0107
  lst=lst+m                                    0108
17 continue                                    0109
18 if(m-1)23,22,19                             0110
19 ist=mm+m                                    0111
  lst=m+l                                      0112
  do 21 i=2,m                                  0113
    ii=lst-i                                  0114
    ist=ist-lst                              0115
    l=ist-m                                  0116
    l=a(l)+0.5d0                             0117
    do 21 j=ii,nm,m                          0118
      tb=r(j)                                0119
      ll=j                                    0120
    do 20 k=ist,mm,m                          0121
      ll=ll+l                                0122
      tb=tb-a(k)*r(ll)                      0123
20 continue                                    0124
  k=j+1                                        0125
  r(j)=r(k)                                    0126
  r(k)=tb                                      0127
21 continue                                    0128
22 return                                      0129
23 ier=-1                                       0130
  go to 22                                     0131
end                                              0132
c*****                                       0133

```

FORMDAT SOURCE CODE, FILE OF DATA CONSTANTS FOR CLAYFORM

SiO2	Si	60.0843	140	0001
TiO2	Ti	79.8988	141	0002
Al2O3	Al	101.96128261		0003
Fe2O3	Fe3+	159.6922	261	0004
FeO	Fe2+	71.8464	121	0005
MnO	Mn	70.9374	121	0006
MgO	Mg	40.3044	121	0007
CaO	Ca	56.0794	122	0008
Na2O	Na	61.97894	222	0009
K2O	K	94.196	222	0010
H2O+	OH	18.0152	223	0011
H2O-	H2O	18.0152	104	0012
LOI	H3O+	18.0152	112	0013
ZrO2	Zr	123.2188	141	0014
V2O3	V	149.8810	261	0015
Cr2O3	Cr	151.9902	261	0016
CoO	Co	74.9326	121	0017
NiO	Ni	74.6994	121	0018
CuO	Cu	79.5454	121	0019

ZnO	Zn	81.3794	121	0020
PbO	Pb	223.1994	122	0021
BeO	Be	25.01158	121	0022
SrO	Sr	103.6194	122	0023
BaO	Ba	153.3294	122	0024
Li2O	Li	29.8814	221	0025
Rb2O	Rb	186.935	222	0026
NH3	NH4	17.0304	112	0027
F	F	18.9984	113	0028
Cl	Cl	35.453	113	0029
B2O3	B	69.6182	261	0030
End		0.0		0031

REFERENCES

- Bodine, M. W., Jr., 1987, CLAYFORM, a FORTRAN 77 computer program apportioning the constituents in the chemical analysis of a clay or other silicate mineral into a structural formula: Computers & Geosciences, v. 13, p. 77-88.
- IBM Applications Program, 1970, System/360 scientific subroutine package, Version III, Programmer's manual (Program No. 360A-CM-03X): IBM Corporation, no. GH20-0205-4, 5th edition, 454 p.