UNIVERSITY COLLEGE LONDON

COMPUTER CENTRE

FORTRAN IN WASHINGTON

A REPORT ON THE MEETING
FROM A FORTRAN VIEWPOINT

BY

Dr A. C. DAY

TECHNICAL REPORT
No. 3

JANUARY 1973

TECHNICAL REPORT NO. 3

JANUARY 1973

MEETING OF ISO/TC 97/SC 5, WASHINGTON, DECEMBER 1972

FORTRAN in Washington:

A Report on the Meeting from a FORTRAN Viewpoint

by

Dr. A. C. Day.

SUMMARY

The ISO Subcommittee on Programming Languages met in Washington D.C. in December 1972. This report describes the work done at that meeting in relation to FORTRAN. In particular the proposals made by the ANSI committee X3J3 for revising the FORTRAN Standard are described and discussed.

Table of Contents

1. Introduction

This is a report on a meeting of ISO (International Organization for Standardization) TC 97 (Technical Committee 97 - Computers and Information Processing) SC 5 (Sub-Committee 5 - Programming Languages). The meeting was held in Washington D.C. from Tuesday, November 28th to Friday, December 1st, 1972, and was the first meeting of SC 5 for five years. The languages under discussion were ALGOL, COBOL, FORTRAN and PL/I, but this report only describes matters relevant to FORTRAN. Any enquiries regarding the other languages, or regarding the work of SC 5 as a whole, should be directed to Mr. Ewart L. Willey, Prudential Assurance Co. Ltd., Holborn Bars, London, E.C.1.

The countries represented were France, Germany, Japan, the Netherlands, Switzerland, the United Kingdom and the United States. The national standards bodies involved were AFNOR (Association Francaise de Normalisation), JISC (Japanese Industrial Standards Committee), ECMA (European Computer Manufacturers Association), BSI (British Standards Institution), and ANSI (American National Standards Institute). The list of delegates is given as Appendix A.

Tuesday morning was devoted to a plenary session in which reports were presented. From Tuesday afternoon to Thursday morning four ad hoc committees were meeting, one for each programming language. From Thursday afternoon onwards the plenary session reconvened to consider and vote on the resolutions.

In July 1972 ISO published a recommended standard for FORTRAN under the document number R1539. This has three levels of the language, corresponding closely with American Standard FORTRAN, ECMA FORTRAN, and American Basic Standard FORTRAN. In this report the recommended standard will be referred to as R1539.

2. First Plenary Session

Reports were presented by each of the delegations. The following points were mentioned which have relevance to FORTRAN standardization.

France. In 1965, SC 5 accepted a draft proposal on FORTRAN. The French and English texts of the draft Recommendation were prepared by AFNOR for submission to all ISO members. The French standard was issued in 1967. Three new editions have since been issued, and editorial improvements were made for each new edition.

Japan. The initial Japanese Industrial Standard (JIS) on FORTRAN was established in May, 1967, in three levels on the basis of ISO FORTRAN. In November 1970 the Programming Languages Technical Committee of the Japanese Industrial Standards Committee (JISC) began the revision of the initial standard with the following criteria: (a) The revision should be minimal. (b) The standard should be made more complete within the scope of the current. specifications. (c) Comments for the revision should be sought widely regardless of the above criteria. On March 1st, 1972, the revised proposals were made JIS's, defining three levels of FORTRAN (ievels 7000, 5000 and 3000).

These correspond to full, intermediate and basic FORTRAN of ISO respectively, with the exception of minor differences. Though there are no official activities at this time to extend the JIS FORTRAN, the FORTRAN Technical Committee of the Japan Electronic Industry Development Association (JEIDA) has studied the extension of JIS FORTRAN and published its comments in 1970 and 1972.

U.S. As a result of the work of ANSI X3J3, two sets of clarifications to the FORTRAN language have been published in the Communications of the ACM, the first in May 1969 and the second in October 1971. X3J3 is also in the process of reviewing extensions to FORTRAN standards. A final draft of the revised documents are expected to be available by mid-1973. In view of the proposed new standard, ANSI does not intend to re-affirm its previous standards (X3.9 and X3.10 - 1966). Such a re-affirmation became due five years after the standards were issued.

ECMA. ECMA TC 8 (FORTRAN) achieved its main objective with the specification and publication of the Standard ECMA-9 for FORTRAN, which was taken as the basis for the intermediate level of the ISO Recommendation on FORTRAN. TC 8 then worked during the period 1967 - 1970 in liaison with ANSI X3J3 to contribute to their activity of clarification of the FORTRAN standard.

Germany. A German version of R1539 has been produced simply by writing a German "envelope" for the English document. A technical dictionary for FORTRAN has also been produced.

U.K. A working party has been set up to review the ISO recommendation on FORTRAN and to prepare a comparable British Standard for FORTRAN. The FORTRAN Specialist Group of the British Computer Society have sent a number of proposed FORTRAN extensions to ANSI X3J3. They have also received the minutes of X3J3 meetings and have discussed and criticised the extensions favoured by X3J3.

3. FORTRAN Ad Hoc Committee

The Chairman of the committee meeting was Mr. Lloyd Campbell. Mrs. Carol Giammo was appointed Secretary. A list of those attending is given as Appendix B.

(a) Dr. Day presented a list of mechanical flaws (misprints etc.) discovered in R1539 by Mr. David Hill. Action on this matter was delayed until committee members had had opportunity to check these.

(b) M. Lanoix presented a note entitled "French comments on FORTRAN". This noted that ANSI are currently working on the revision of the FORTRAN standard, and continued:-

> "AFNOR is strongly of the opinion that, if the decision is made to undertake similar work in ISO to revise R1539, great care should be taken to avoid incompatibilities between the proposed revised International Standard and the current ISO Recommendation. The objective should be that the revised International Standard on FORTRAN be fully compatible with R1539".

The committee supported this opinion. Great care should be taken to avoid, if possible, incompatibilities between the current ISO Standard (R1539) and any future ISO Standard.

(c) Mr. Tokunaga presented a list of differences between JIS FORTRAN and ISO FORTRAN. This is given as Appendix C. Mr. Tokunaga explained that point (1) was included because some Japanese compilers will only accept two continuation lines, and yet in all other respects accept the full standard. Points (1), (2) and (4) therefore represent limitations, point (3) an extension, and point (5) a clarification.

(d) The relationship between the clarifications to ANSI X3.9-1966 and the ISO document R1539 was discussed. It was pointed out that the clarifications did not have the official approval of ANSI, but were merely the findings of X3J3. These clarifications have no official status with regard to R1539, and indeed should have none. However, the clarifications were being produced at the same time as R1539, and did influence the ISO document to some extent.

(e) Mr. Frank Engel, the Chairman of X3J3, spoke on the work of that subcommittee. Very shortly after the ANSI Standard appeared, questions were being asked concerning problems and ambiguities in it. From 1966 to 1970 X3J3 were concerned with clarifications. Two reports were published during that period. In one sense, it was not possible to provide clarifications without extending the Standard. More effort was put into the work of revising the Standard. ANSI rules require any Standard to be re-affirmed or revised every five years (unless the Standard is to be abolished). It was determined to revise X3.9 and X3.10, and not to re-affirm them in 1971. Most of the effort has been directed to revising X3.9. Rather than re-affirming X3.9 and X3.l0, and constructing a new Standard which is a superset of both, it was considered better to produce a new Standard language first, and then to standardise subsets of it, which may not correspond to X3.9 and X3.l0.

The criteria set down by X3J3 for acceptable extensions to FORTRAN are included as Appendix D. Approval for any proposal considered in an X5J3 meeting is by majority vote. The final document must be accepted by a two-thirds vote of a letter ballot of X3J3. Then the document will go out for comment, and may be changed at that stage.

Mr. Engel presented a list of the modifications to FORTRAN which have been initially approved by X3J3. Appendix E has been taken almost entirely from the document written and circulated by Mr. Engel, but some changes have been made to bring the list up to date and to add some more explanatory matter.

There was further discussion in the committee on a number of the items given in Appendix E.

III 4. X3J3 has since rejected this proposal.

6 & 7. The writer was assured that this did not mean that the evaluation must of necessity be from left to right, but merely that the same effect be produced. Type conversion in a mixed-mode expression will only be performed as it is needed for a particular operation.

17. The form EXTERNAL *name may be used to specify the user's own function (and not a Basic External Function of the same name).

28. This proposal was originally for Ew.dEe and Ew.dDe.

36. This will invalidate some programs which currently run. It does not, however, contradict the present standard.

46. Whether the entry name is a function or a subroutine depends entirely on the main entry point. The values of dummy arguments become undefined on return.

IV 1. The new character involved in this extension is not yet determined. All the new characters will be chosen together at one time when the extensions have been decided upon. (This also concerns III 24.)

2. Dr. Day suggested that the implied-DO in an I/O list should be subject to the same extensions as the DO loop, to preserve regularity (III 32-37).

3 & 4. Dr. Day pointed the irregularity of these extensions, in that the items could not be passed as arguments, and it was not certain that they could appear in I/O lists. Ad hoc restrictions were needed to prevent the definition of subarrays in a recursive manner. A straw vote by delegations was in favour of removing the SUBARRAY proposal by a majority of 3-1 (the U.S. voting against). This proposal had come close to being removed in a recent meeting of X3J3.

VI 1. The length feature of the character variables is to be specified by an asterisk (like /360 notation). Parentheses are needed when the length is an expression, and perhaps also a comma following. Dr. Day pointed out that this was highly irregular, and could be simplified greatly if parentheses were always used, without the comma or the asterisk, and if the length then applied to all variables in the statement.

The X3J3_reason for keeping the asterisk was that this kept the extension in line with the IBM method for indicating precision. One day precision of variables might be specified in the FORTRAN standard also. Dr. Day's reply was that the IBM method specified precision in terms of machine dependent bytes, whereas the standard would perhaps specify it in terms of decimal digits. For this reason it might be preferable to avoid similarity with the IBM asterisk notation. In addition, it was pointed out to X3J3 that the character type should be a full type, and should include functions, even though these will need to return a character string of predefined length.

As the latest proposals from the X3J3 working group on direct access I/O have not yet been considered by X3J3 as a whole, Mr. Richard Karp gave an informal report on their ideas to date. See Appendix F. Dr. Day asked how one would use the VARYING option, and was told that the I/O list in the READ or WRITE statement would need to be of the right length to match the record. He pointed out that this facility already exists in the present Standard, and the new keyword provides no advances.

(f) The following recommendations were adopted, to be sent to the plenary session of SO5:

Recommendation No. 1

Moved by Dr. Day (UK), seconded by Mr. Engel (USA):

ANSI is requested to forward to ISO/TC97/SC5 its Draft Proposed FORTRAN Standard at the earliest possible time. This document should be considered by ISO/TC97/SC5 as a possible basis for any ISO proposed revision to the current ISO FORTRAN, R1539.

The motion was passed unanimously (4~0).

Recommendation No. 2

(Discussion) The FORTRAN ad hoc Working Group has considered the recommendation from the Netherlands for the installation of a FORTRAN maintenance (working) group, and is of the opinion that:

a. If the intention of the recommendation is to provide clarifications to R1539, then there is no demonstrated need for such a group at the present time.

b. If the intention of the recommendation is to make revisions to R1539, then the matter is covered by the FORTRAN ad hoc Working Group Recommendation No. 1.

(Motion) Moved by Dr. Day (UK), seconded by Mr. Tokunaga (Japan):

SC5 should take no action at the present time towards forming a FORTRAN maintenance (working) group.

The motion was passed unanimously (4-0).

Recommendation No. 3

Moved by Dr. Day (UK), seconded by Mr. Engel (USA):

ISO/T097/SC5 strongly deplores the excessive price of ISO documents in that it limits the availability of ISO documents to ISO working groups and to the user community.

The motion was passed unanimously (4-0).

Recommendation No. 4

Moved by Mr. Noll (USA), seconded by Mr. Tokunaga (Japan):

Recommend that the attached list of mechanical flaws presented to the FORTRAN ad hoc Working Group by Dr. Day (UK) calling attention to mechanical flaws in ISO/R1539 be forwarded to the Secretariat for verification. Upon verification by the Secretariat, mechanical flaws shall be corrected in subsequent printings of ISO R1539.

The motion was passed unanimously (3~O).

(g) The Committee reviewed a document on numeric data for information interchange (N270), in order to comment on any incompatibilities with R1539. The following conflicts were found:

1. All forms of the exponent containing the character D are not acceptable to N270.

2. The exponent form with the character E omitted is not acceptable to N270.

3. The exponent form containing the character blank as a replacement for the character + is not acceptable to N270.

4. N270 insists on a blank character in the absence of an explicit sign.

5. A small negative number may be output according to R1539 as an apparent negative zero which N270 does not permit.

4. Second Plenary Session

Mr. Campbell reported on the FORTRAN ad hoc Committee meeting. The recommendations in 3(f) above were presented, to be voted on later in the plenary session. Mme. Chasles (France) then presented a further recommendation:

"Recommended that, if the decision is made to revise the current ISO recommendation on FORTRAN (R1539) great care be taken so that the revised standard is fully compatible with R1539”.

The discussion on this recommendation was particularly lively. The UK pointed out that the recommendation could be interpreted as requiring absolute compatibility, thus preventing necessary regularisations of the language which may affect only a tiny number of users. The French refused to countenance any compromise on the strength of the wording, such as adding the phrase "as far as possible". The USA revealed that there was one incompatibility which was thought necessary at the moment. This was that in the proposed new standard excess parentheses would not be permitted in I/O lists, as otherwise in a list-directed WRITE statement, two real constants in parentheses could not be distinguished from one complex constant. However, few compilers permit excess parentheses in an I/O list at the moment.

A vote was taken on whether this new recommendation should be accepted as a recommendation (i.e. whether it should be added to the

list for voting on later in the plenary session). It was accepted by a vote of 4-3, Switzerland, the Netherlands, Germany and France voted for and Japan, the UK and the USA against.

On Friday morning the recommendations were put to the vote. All the basic recommendations (in 3(f) above) were passed nem con. The UK had meanwhile provided an alternative wording for the French recommendation above. This read:

"Resolved that in the preparation of a new FORTRAN standard the greatest care be taken to avoid incompatibility with the present standard."

To this Switzerland added the following amendment:

"However, should incompatibilities arise, they should be documented in an appendix to the new standard".

The original French recommendation was defeated by a vote of 2-4-1 (Switzerland and France voting for, and Germany abstaining). The Swiss amendment to the UK wording was accepted (5-0-2), and the amended recommendation was accepted (5-1-1, France against and Germany abstaining).

The recommendations became, on acceptance, resolutions of ISOfiTC97/SC5.

5. BCS Action

In the FORTRAN ad hoc Committee, members of X3J3 had mentioned their high respect for the proposals of the British Computer Society FORTRAN Specialist Group. It had also been pointed out that X3J3 were reaching the stage when no further proposals could be considered, and the production of the new draft standard would begin. Comments and criticisms of the work done so far would be welcome, especially if they were submitted in time for consideration at the next meeting of X3J3, which was scheduled for January 10th.

In view of this, a meeting of the Extensions Working Party of the BCS FORTRAN Specialist Group was held on December 6th to consider proposals put forward by Dr. Day. The proposals were then submitted to the FORTRAN Specialist Group on December 7th, and were sent to X3J3 on December 11th. They are included here as Appendix G.

6. Acknowledgements

I must acknowledge my deep indebtedness to the National Computing Centre for providing my fare to Washington and back, and to my Director, Professor Paul A. Samet, of the Computer Centre of University College London, for freeing me from normal duties to be able to attend all the meetings. I am very grateful to the BCS FORTRAN Specialist Group for their help, and especially to the chairman, Mr. John Gatehouse, for inviting me to be the FORTRAN delegate in Washington.

Dr. A. Colin Day.

[page 8 is blank]

APPENDIX A

ATTENDANCE LIST
*Indicates Chief Delegate

Robert W. Bemer - Chairman

COUNTRY/NAME	ORGANIZATION
FRANCE	
Mrs. Francoise Chasles*	IBMé
Pierre Allené	Electricité de France
Albert Astières	STERIA
Jean Bourgain	Honeywe11-Bull
Frantz Lanoix	AFNOR
Dominique Truchetet	CELAR
GERMANY	
Prof. Dr. K. Samelson*	Technical University Munich
Dr. Jan Witt	Siemens AG, Munich
JAPAN	
Megumi Fujinaka*	Hitachi, Ltd.
Kohichi Aramaki	Hitachi, Ltd.
Hirohiko Nisimura	Electrotechnical Lab., MITI
Nobuo Suzuki	Nippon Electric Co., Ltd.
Eiji Tokunaga	IBM Japan Ltd.
NETHERLANDS	
W.B.C. Ebbinkhuijsen	Stichting Studiecentrum Voor Informatica
SWITZERLAND	
Jean Besse	European Computer Manufacturers Assn.
UNITED KINGDOM	
Ewart L. Willey*	Prudential Assurance Co., Ltd.
Maurice Batey	IBM - Europe
Angus Beatty	International Computers Ltd.
A. Colin Day	University College London.
I. David Hill	Medical Research Council
J.M. Sykes	ICI Limited
UNITED STATES	
Robert Kearney*	Bell Telephone Laboratories
George N. Baird	U.S. Navy
Peggy A. Beard	NCR
Richard P. Belmont	Honeywell
Robert F. Betscha	IBM
Lloyd W. Campbell	U.S. Army

COUNTRY/NAME	ORGANIZATION
U.S. contd.	
Lois C. Frampton	Food and Drug Administration
Marjorie L. Green	Equitable Life Assurance
Richard A. Karp	Burroughs
Michael Marcotty	General Motors Research Labs
Robert E. Rountree, Jr.	National Bureau of Standards
Richard L. Wexelblat	Bell Telephone Labs
David Beech**	IBM
Frances E. Holberton**	National Bureau of Standards
Marie E Hogsett	ANSI Secretariat
Maryse Charleston	ANSI Secretariat
Helene Bruns	Interpreter

** Observer

APPENDIX B

ATTENDANCE

FORTRAN AD HOC WORKING GROUP

Representative	Country	Affiliation
Campbell, L.W.	United States	U.S. Army
Day, A. Colin	United Kingdom	University College London
Engel, Frank, Jr.	United States	Self Employed
Giammo, Carol	United States	U.S. Government
Holberton, F.E.	United States	National Bureau of Standards
Karp, Richard A.	United States	Burroughs Corporation
Lanoix, Frantz	France	AFNOR
Noll, J.C.	United States	Bell Telephone Laboratories
Tokunaga, Eiji	Japan	IBM Japan Ltd.
Greenfield, Martin N.	United States	Honeywell

[page 12 is blank]

APPENDIX C

Differences between JIS FORTRAN and ISO FORTRAN

The following differences exist between JIS FORTRAN and ISO FORTRAN:

(1) In all levels of the JIS FORTRAN, the number of the continuation lines of a statement is not specified.

(2) In the JIS FORTRAN (level 7000), the extended range of a DO is defined as the extended range of a DO may not contain a DO statement and a READ/WRITE statement containing DO-implied list of the same program unit" instead of the definition of ISO full FORTRAN "the extended range of a DO may not contain a DO of the same program unit that has an extended range".

(3) In the JIS FORTRAN (level 7000), the array name is also permitted to a list of a DATA statement in addition to the names of variable and array element.

(4) In all levels of JIS FORTRAN, the magnitude of the values given for subscript expressions is defined as "the value given for the subscript expression in the array declarator subscript indicates the maximum value that the subscript expression may attain in any array element name" instead of the definition in ISO full FORTRAN "the magnitude of the values given for the subscript expressions indicates the maximum value that the subscript may attain in any array element name".

(5) In all levels of the JIS FORTRAN, the definition of ATAN in the basic external functions is added as follows

"The range of the value covered by arctan (a) is from $-\pi/2$ to $\pi/2$.

[page 14 is blank]

APPENDIX D

CRITERIA FOR 1971 DRAFT PROPOSAL

1. Interchangeability of FORTRAN programs between processors.

2. Compatibility with X3.9 - 1966, allied standards, and existing practices.

3. Consistency and simplicity to user.

4. Suitability for efficient processor operation for a wide range of computing equipment of varying structure and power.

5. Allowance for future growth in the language.

6. Achievement of capabilities not currently available, but needed for processes appropriately expressed in FORTRAN.

7. Acceptability by a significant portion of users.

8. Availability of Preliminary Draft Proposed Standard (PDPS) in 1971.

9. Improved ability to use FORTRAN programs and data in conjunction with other languages and environments.

[page 16 is blank]

APPENDIX E

X3J3 Proposed Extensions to FORTRAN

The following list of modifications to the ANS X3.9-1966 FORTRAN language have been approved initially by the X3J3 committee of the American National Standards Institute. This list has been prepared for informational purposes only. It is not to be construed that any or all of these modifications will be incorporated in the draft proposed American National Standard FORTRAN, or that if included, they will be in exactly the form described, inasmuch as development work on the dpANS FORTRAN is still in progress, and changes may also be introduced subsequently as the result of critical review during the ANSI approval cycle.

I. Minor revisions and certification of common practice:

1. Define REWIND to have no effect when the unit is at the initial position.

2. Require ASSIGN statement to be in same program unit as the assigned GO TO or FORMAT statement to which it refers.

3. Require consecutive slashes in format specification to cause printing of blank lines.

4. Allow an optional comma in assigned and computed GO TO statements and preceding the control variable in DO statement.

5. Allow in a comment line any characters capable of representation by the processor.

6. Include the apostrophe ‘ in the FORTRAN character set.

7. Allow Hollerith character strings to be delimited by apostrophes.

8. Allow quoted character strings in PAUSE statements.

9. Allow decimal digits in PAUSE and STOP statements.

10. Require characters in STOP statement be accessible.

11. Allow unlimited number of digits in real or double precision constant.

12. Require that the number of digits in an input datum under D, E, F or G type conversion is limited only by the field width w.

II. Minor extensions to the language:

1. Make optional the statement label list in the assigned GO TO statement.

2. Allow the IMPLICIT statement.

3. Allow as terminal statement of a DO-loop, any executable statement for which, upon completion of its execution, execution of the next following statement may take place.

4. Allow READ, PRINT and PUNCH statements, with respective units processor defined.

5. Allow many levels of parentheses in a format specification.

6. Require printing of asterisks in entire field when value exceeds field width in type Iw or Fw.d conversion.

7. Restrict scale factor nP in E & D output fields, such that -d < n < d+2.

8. Allow optional + sign in scale factor in FORMAT statement.

9. Allow ASSIGN statement to set the value of variable format reference in a formatted I/O statement.

1O. Specify units of arguments, ranges and results for the Basic External Functions.

11. Allow a PROGRAM statement in main program unit to permit the naming of a main program.

12. Allow naming of BLOCK DATA subprograms.

13. Allow more than one BLOCK DATA subprogram in an executable program.

14. Require END line to act as a STOP statement in a main program unit, and as a RETURN statement in a subprogram.

15. Remove concept of second level definition.

III. General extensions to the language:

1. Allow functions with no arguments to be defined and referenced, e.g. F()

2. Extend the rules for assignment of e to v to allow complex expressions with real, double precision or integer entities, and vice versa.

3. Require a complex entity to be equivalent to two type real entities, the first being the real part and the second being the imaginary part of the complex number.

4. Allow multiple assignment statements, e.g. v_1 , v_2 ,v_3 ,...=e [Note: This has now been removed.]

5. Specify right to left grouping of successive exponentiations, removing ambiguity. A**B**C**D means A**(B**(C**D)).

6. Require left to right expression evaluation for operations of same hierarchic level.

7. Allow mixed mode arithmetic expressions.

8. Allow array to have maximum of seven dimensions.

9. Allow implied-DO in DATA statement list.

1O. Allow array name (sans subscript) in DATA statement list.

11. Allow array name in EQUIVALENCE statement.

III. General extensions (cntd).

12. Allow an array element name in a statement function expression.

13. Allow generalized subscript expressions.

14. Allow variables used as adjustable dimensions in COMMON statement.

15. Allow redefinition and/or undefinition of adjustable dimensions within a subprogram unit without affecting the array dimensions.

16. Allow integer expressions as adjustable dimensions.

17. Require additional Basic External Functions:
ASIN. DASIN, ACOS, DACOS, SINH, DSINH, COSH, DCOSH,
DTANH, TAN, DTAN
and additional Intrinsic Functions:
DINT, DFLOAT, DDIM, DPROD

18. Allow ERR= in READ/WRITE statements.

19. Allow END= in READ statement.

20. Allow more than one file to appear on one sequential unit.

21. Allow list directed I/O.

22. Allow expressions in output lists.

23. Allow quoted character constant in output list of formatted WRITE statement, with a corresponding Aw format field descriptor required.

24. Allow a designated character to stop scan of a format specification at the end of an I/O list.

25. Allow -nX and +nX format field descriptors for relative tabbing, with implied initialization of record to blanks on output.

26. Allow T format field descriptor for absolute tabbing, with implied initialization of record to blanks on output.

27. Require processor to write + sign on positive or zero exponent in Ew.d and Dw.d fields.

28. Allow control of exponent size by Ew.d.e & Dw.d.e conversion on output.

29. Allow BACKFILE & SKIPFILE statements to position sequential unit.

29a. Allow BACKSPACE u,n to backspace n logical records.

30. Allow integer expressions in computed GO TO statement.

31. Require the execution of the next following statement, if the computed GO TO expression is out of range.

32. Allow real or double precision type control variable in DO statement.

33. Allow any arithmetic expression in DO-parameter with appropriate conversion.

34. Allow negative values as DO~parameters.

35. Require determination of incrementation value and number of range executions at start of execution of DO statement.

36. Require minimum number of DO iterations to be zero.

37. Allow redefinition and/or undefinition of DO parameter variables within range without affecting number of iterations.

38. Allow branching into the range of any active DO.

39. Allow a CONSTANT (PARAMETER) statement.

4O. Allow ENTRY statement in subprogram to provide additional entries and names.

IV. Array extensions to the language:

1. Allow non-unit lower bounds for an array dimension.

2. Allow block transfer as a list item in I/O list.

3. Allow * as a subscript to be an array cross-section reference and -* to be a reverse cross-section.

4. Allow a SUBARRAY statement.

5. Allow array assignment statement with any arithmetic expression, An array assignment statement is an assignment statement in which v is an array cross-section. The expression e may or may not contain array cross-section references.

V. Direct Access I/O extensions:

1. Allow OPEN and CLOSE statements.

2. Allow direct access READ and WRITE statements, both formatted and unformatted.

3. Allow direct access files to contain fixed length or variable length records.

VI. Character Data type extensions:

1. Allow CHARACTER type specifications statement.

2. Allow character type constants.

3. Allow character string references in both variables and array elements.

4. Allow character type expressions in assignment statement, argument of subprogram, and I/O lists. A concatenation operator is defined for character data.

5. Allow character expressions as I/O unit or as format reference in formatted READ/WRITE statements. When used as the I/O unit, this permits incore transmission by means of a FORMAT statement.

APPENDIX F

Proposals for Direct Access I/O

It is proposed to add three statements to FORTRAN, the OPEN, CLOSE and INQUIRE statements. Informally, the options available with these statements are shown below

```
                                      SCRATCH        DIR      FMT
OPEN( UNIT=exp, NAME=charexp,          OLD     ,     SEQ   ,  UNFMT ,
                                       NEW

            MAXREC = exp, READONLY, ERR=label, VARYING)

          KEEP
CLOSE(   DELETE , UNIT=exp)


          UNIT=exp
INQUIRE( NAME=charexp , MAXREC=X, EXISTS=B)
```

The order of the parameters is variable, and not all are obligatory. The use of MAXREC in the OPEN statement is to define the maximum record length for the file. In the INQUIRE statement, MAXREC=X will place the maximum record length into the variable X, and EXISTS=B will set the logical variable B to .TRUE. if the file exists, and to .FALSE. otherwise.

READ and WRITE statements will be extended so that the unit and format references may be written as UN1T= and FMT= in which case the order of the parameters is variable. For direct access READ and WRITE statements, the extra parameter REC= is needed.

REWIND, BACKSPACE etc. will have no effect on direct access files.

[page 22 is blank]

APPENDIX G

December 8th 1972

TO: ANSI X3J3

FROM: BCS Fortran Specialist Group

SUBJECT: Proposals concerning the new Fortran standard.

PROPOSAL 1. That the SUBARRAY statement be removed from the new standard.

Rationale: (a) Subarrays are irregular (compared with arrays) in terms of the places where they may appear. They cannot be used as arguments to subprograms. It is not certain whether they may appear in I/O lists. They may not be used for a FORMAT specification, as the storage may be non-contiguous. They cannot occur in DATA lists, as they may be controlled dynamically by means of variable subscripts.

(b) The subscript expressions permitted by the new standard in DIMENSION and type statements cannot be allowed for the subscripts in SUBARRAY statements, as this would imply deferred evaluation of expressions, causing very great inefficiencies for each subarray reference. Redefinition of adjustable dimensions will not have any effect in a DIMENSION or type statement, but the reverse is true for SUBARRAY.

(c) Ad hoc restrictions must be imposed to prevent recursive definitions. Although the X3J3 proposal does not say so, presumably array element references are not allowed in the subscripts, as otherwise the following statement would be possible:

```
SUBARRAY JIM(34) AT A(JlM(2))
```

Even without this problem, recursion can occur, as shown by:

```
SUBARRAY JIM(34) AT A(23), A(45) AT JiM(32)
```

(d) If the proposal for subarrays is to be compatible with the new rules for DIMENSION and type statements, then lower bounds should be included, introducing further complications.

(e) The syntax for the statement is peculiar. It has keywords in the middle of the statement, a situation matched only by the ASSIGN statement.

(f) The ‘principle of least astonishment' is violated in such examples as:

```
      DO 13 I = 1,15
   13 A(4) - B(4)
```

Here the loop can only be redeemed from the realm of nonsense (or from being optimised out of existence) by being preceded by statements such as:

```
DIMENSION G(55), D(55)
SUBARRAY A(6) AT C(I), B(6) AT D(I)
```

(g) The SUBARRAY statement is not likely to be of help to many.

PROPOSAL 2. That reverse cross~sections of arrays be removed from the new standard.

PROPOSAL 3. That array cross-sections be removed from the new standard.

Rationale: (a) As with the SUBARRAY proposal, the problem of non-contiguous storage is introduced. Cross-sections may therefore not be passed as arguments to a subprogram, and may not be used for a FORMAT specification.

(b) If cross—sections are nevertheless passed as arguments, the irregular situation arises whereby correct results may be obtained only if the cross—section is forward and is based on the first subscript.

(c) Ad hoc rules need to be added so that an array assignment statement using cross sections does not cause overlapping to occur.

(d) For reasons of regularity, many repercussions of this proposal must be considered. Are cross-sections to be permitted in I/O lists? Or block transfer list items in an assignment statement? These are apparently two proposals to do very similar things, but have quite different limitations (one cannot step backwards, and the other cannot refer to less than a complete cross-section). Are cross~sections to be permitted in DATA statements? Or implied DO loops in assignments? Have cross-sections of subarrays been considered? This whole area seems rather ill thought out, with partial extensions criss-crossing one another.

PROPOSAL 4. That the order of specification statements must not be more restrictive than in the present standard.

Rationale: (a) As the meeting of ISO/TC/97/SC 5 showed, there is very great pressure for the new standard to be compatible with the old. To restrict the order of specification statements would be an incompatibility of the first magnitude, which would invalidate a considerable proportion of the programs which currently conform to the standard.

(b) If the justification for this restriction is in order to make things simpler for a one—pass compiler, then the restriction could well be made a feature of one of the lower subsets of the new standard. However, as far as we can see, this should not present problems even to a one-pass compiler. We are not sure whether the proposed restriction would involve DATA statements, and we would appreciate further information on this point.

PROPOSAL 5. That the length of variables defined in a CHARACTER statement should be given.in parentheses instead of following an asterisk and preceding a comma. The length should only occur following the keyword CHARACTER, and then applies to every variable defined in that statement.

A comparison is made here of Boswell's ‘Compromise Syntax‘ and 'BCS Syntax' to show how they correspond.

Compromise Syntax	BCS Syntax
CHARACTER*5, A(1O),B(2,2),C	CHARACTER(5) A(10),B(2,2),C
CHARACTER*(N), D	CHARACTER(N) D
CHARACTER*(*), E(11)	CHARACTER(*) E(11)
CHARACTER*(N*M), F(13)	CHARACTER(N*M) F(13)
	CHARACTER(3) G, J
CHARACTER*3, G, H*5, J	CHARACTER(5) H

Rationale: (a) According to the X3J3 proposal, parentheses would be necessary only if the length was not a constant. The ;s asterisk and comma are always needed. If parentheses are always obligatory the rule is perfectly simple and regular, and both the comma and the asterisk may be dispensed with.

(b) The X3J3 proposal is unsatisfactory in such cases as the declaration for G, H and J above. The inexperienced user is liable to wonder whether the length of J is 3 or 5, as he will have learnt from the P factor in a FORMAT statement and from the COMMON statement that a change made in the middle of a statement may apply to succeeding items.

(c) It seems very probable that future Fortran standards will include precision of numeric variables. This should be declared in terms of decimal digits, and for economy of syntax should parallel the length feature of character variables. However, the asterisk notation is used by IBM for precision in gems terms of bytes. We should therefore avoid this notation for representing the length of character variables.

PROPOSAL 6. That the new standard should allow programmer~defined functions of CHARACTER type, returning a fixed length character value whose length has been declared in both the calling program and the function itself.

Rationale: (a) If CHARACTER type is introduced, it should be a fully regular type, matching in every way possible the types already available.

(b) Although a variable length function may well be preferable, this seems out of the question at the present time. It is nevertheless better to be able to write fixed length character functions rather than none at all.

(c) It should be easy to implement such functions, seeing that the amount of storage required for the value is known to both calling and called segments, just as for functions requiring one or two words for the result. If character arguments may be transferred, then character functions can be implemented, as the value returned can be treated as an additional argument.

PROPOSAL 7. That the implied DO in an I/O list should be extended in the same way as the DO statement, allowing expressions as the parameters, and real or double precision variables as the control variables. Function references in the expressions must not cause I/O operations to be performed.

Rationale: (a) The two types of loop-controlling features should be kept in step for reasons of regularity.

(b) A real or double precision control variable could in certain circumstances be very useful. The I/O list may contain a function reference for which one argument should be the control variable, or it may be necessary to write out a regular series of real values. The same extensions cannot, of course, be applied to the DATA statement, as any implied DO there must be interpreted at compilation time.

PROPOSAL 8. That an alternative keyword should be found for the PARAMETER statement.

We agreed in our dislike for "PARAMETER", but we were not able to reach full agreement on any other term. "CONSTANT" won most approval. Other keywords suggested were "HOLD", "SYNONYM", "EQUALITY‘" and “IDENTITY".

Rationale: (a) The term PARAMETER is used widely to describe an argument. This will cause much confusion, as the items defined by the statement are certainly not arguments.

(b) Although i11-phrased references to CONSTANT variables may be confusing, it is true that the items declared by means of this statement are symbolic representations for constants, and some term should be chosen which reflects this fact.

(c) It may well be true that the only implementation of this feature uses the term PARAMETER, but the fact that the first implementor got the term wrong should not be used as a reason to foist this unfortunate error on the whole Fortran community.

PROPOSAL 9. That the BACKSPACE, RENIND, ENDFILE, SKIPFILE, BACKFILE and any other statement referring to I/O operations should permit the unit reference to be:

(UNIT=value)

Rationale: This is in order to maintain and increase the regularity between these statements and the READ, WRITE, OPEN and CLOSE statements (see Proposal 13).

PROPOSAL 10. That the INQUIRE statement not be added to the new standard.

PROPOSAL 11. That direct access READ and WRITE statements be permitted in Fortran, differing from the corresponding sequential READ and WRITE statements by the addition of the parameter

REC=value

within the parentheses enclosing the unit designation.

PROPOSAL 12. That wherever the new standard includes as keywords abbreviations of English words (e.g. ERR, REC), the full word also be permitted as an alternative (e.g. ERROR, RECORD).

PROPOSAL 13. That OPEN and CLOSE statements be permitted of the following form

```
OPEN (UNIT=value, NAME=charvalue, ERR=label)
CLOSE (UNIT=value)
```

In the OPEN statement, the specification of NAME and ERR are both optional. If the unit designation comes first within the parentheses, then "UNIT=" may be omitted. If the unit is specified by an integer constant or an integer variable, and no other information is given by the OPEN statement, the parentheses may be omitted, and the statement takes the form:

OPEN value

The OPEN statement is executable, causing the program to become associated with the file. The charvalue above is a character value which provides a way of referring to the file in terms external to the Fortran program. If it is not possible to associate the program with the file, execution continues at the statement specified by the label. All other attributes of the file (old or new, device type, record length, label type, protection, direct or sequential, keyed or unkeyed, space, disposition on closing, blocking factor etc.) will (if they are necessary) be given outside the Fortran program. If an OPEN statement for a particular unit has not been executed, the first READ or WRITE statement for that unit will implicitly open the file.

The CLOSE statement is also an executable statement, causing the program to become disassociated with the file. It this statement "UNIT=" may be omitted, in which case, if the unit designation is an integer constant or an integer variable, the parentheses may also be omitted, and the statement takes the form:

CLOSE value

Rationale: (The X3J3 proposals referred to here are those which were described at the Washington meeting of ISO/TC 97/SC5.)

(a) As "keyword parameters" have already been included in the proposed new standard through "END=" and "ERR=" it seems good to use this syntactic technique elsewhere. However, the X3J3 proposals have parameters which are in a sense "keyword" and yet have no equals sign (e.g. VARYING). These have no parallel in Fortran. If OPEN is to be made regular with READ and WRITE, so that

"UNIT=" may be omitted if the unit is given first, then ambiguity results, since a keyword such as NEW cannot be distinguished from a variable unit number if it is in the first position.

(b) Some of the proposed features of the OPEN statement are incompatible with the present Fortran standard. For instance, specifying whether a file is formatted or unformatted contradicts the present possibility of writing on one unit, with a mixture of formatted and unformatted WRITE statements. The VARYING option does not do anything, as at the moment one can read records of different lengths provided that the I/O list plus the format tally; with the new option one must still provide an I/O list (and format) of the correct length. The new standard should allow a mixture of direct access and sequential I/O statements to be used with the same file.

(c) The proposed INQUIRE statement uses keyword parameters to specify variables which are to be given values. The result is similar to an assignment statement in which the assignment is from left to right, and is highly undesirable.

(d) It appears that the aim of the X3J3 proposals has been to include all features of the file, so that non-Fortran control statements may be eliminated, a state of affairs which may be especially helpful to users of terminals. This has not been successful. Some features which will be of vital necessity have not been included, e.g. the number of records for a new direct access file. Some features place an intolerable burden on the operating system, e.g. using INQUIRE to find the maximum record length of an unlabelled magnetic tape file, which involves reading every record. It is impossible to specify the device type required for the file without including in the Fortran language a list of obsolescent devices which will rapidly become incomplete, and yet it may be necessary for the user to inform the system cf the device, even for scratch purposes. For these reasons it seems clear that Fortran should not be used to give information about the file which is not needed by the Fortran program.

(e) The BCS proposals given above are both simple and effective in providing new I/O facilities for Fortran. By means of the “ERR=" parameter in the OPEN statement an enquiry can be made as to whether an old file exists, or whether there is sufficient space available for a scratch file. By the use of OPEN and CLOSE statements different files may become associated with the same unit number at different times during execution. External names can be read in from the data and used in the NAME parameter of the OPEN statement. By means of CLOSE, devices may be unloaded and freed, or scratch space may be released.

Colin Day.