NAT'L INST. OF STAND & TECH R.I.C.

A11103 812058

NIST Special Publication 500-203

Computer
Systems
Technology

U.S. DEPARTMENT OF
COMMERCE
Technology Administration
National Institute of
Standards and Technology

NIST

Conformance Test Specifications for COBOL Intrinsic Function Module

Carmelo Montanez-Rivera
L. Arnold Johnson

NIST
PUBLICATIONS

QC
100
.U57
500-203
1992
C.2

AUTHENTICATED
U.S. GOVERNMENT
INFORMATION
GPO

The National Institute of Standards and Technology was established in 1988 by Congress to "assist industry in the development of technology . . . needed to improve product quality, to modernize manufacturing processes, to ensure product reliability . . . and to facilitate rapid commercialization . . . of products based on new scientific discoveries."

NIST, originally founded as the National Bureau of Standards in 1901, works to strengthen U.S. industry's competitiveness; advance science and engineering; and improve public health, safety, and the environment. One of the agency's basic functions is to develop, maintain, and retain custody of the national standards of measurement, and provide the means and methods for comparing standards used in science, engineering, manufacturing, commerce, industry, and education with the standards adopted or recognized by the Federal Government.

As an agency of the U.S. Commerce Department's Technology Administration, NIST conducts basic and applied research in the physical sciences and engineering and performs related services. The Institute does generic and precompetitive work on new and advanced technologies. NIST's research facilities are located at Gaithersburg, MD 20899, and at Boulder, CO 80303. Major technical operating units and their principal activities are listed below. For more information contact the Public Inquiries Desk, 301-975-3058.

Technology Services

- Manufacturing Technology Centers Program
- Standards Services
- Technology Commercialization
- Measurement Services
- Technology Evaluation and Assessment
- Information Services

Electronics and Electrical Engineering Laboratory

- Microelectronics
- Law Enforcement Standards
- Electricity
- Semiconductor Electronics
- Electromagnetic Fields[1]
- Electromagnetic Technology[1]

Chemical Science and Technology Laboratory

- Biotechnology
- Chemical Engineering[1]
- Chemical Kinetics and Thermodynamics
- Inorganic Analytical Research
- Organic Analytical Research
- Process Measurements
- Surface and Microanalysis Science
- Thermophysics[2]

Physics Laboratory

- Electron and Optical Physics
- Atomic Physics
- Molecular Physics
- Radiometric Physics
- Quantum Metrology
- Ionizing Radiation
- Time and Frequency[1]
- Quantum Physics[1]

Manufacturing Engineering Laboratory

- Precision Engineering
- Automated Production Technology
- Robot Systems
- Factory Automation
- Fabrication Technology

Materials Science and Engineering Laboratory

- Intelligent Processing of Materials
- Ceramics
- Materials Reliability[1]
- Polymers
- Metallurgy
- Reactor Radiation

Building and Fire Research Laboratory

- Structures
- Building Materials
- Building Environment
- Fire Science and Engineering
- Fire Measurement and Research

Computer Systems Laboratory

- Information Systems Engineering
- Systems and Software Technology
- Computer Security
- Systems and Network Architecture
- Advanced Systems

Computing and Applied Mathematics Laboratory

- Applied and Computational Mathematics[2]
- Statistical Engineering[2]
- Scientific Computing Environments[2]
- Computer Services[2]
- Computer Systems and Communications[2]
- Information Systems

[1] At Boulder, CO 80303.
[2] Some elements at Boulder, CO 80303.

QC100
.U57
500-202
1992
C2

Conformance Test Specifications for COBOL Intrinsic Function Module

Carmelo Montanez-Rivera
L. Arnold Johnson

Computer Systems Laboratory
National Institute of Standards and Technology
Technology Administration
Gaithersburg, MD 20899

July 1992

U.S. DEPARTMENT OF COMMERCE
Barbara Hackman Franklin, Secretary

TECHNOLOGY ADMINISTRATION
Robert M. White, Under Secretary for Technology

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
John W. Lyons, Director

Reports on Computer Systems Technology

The National Institute of Standards and Technology (NIST) has a unique responsibility for computer systems technology within the Federal government. NIST's Computer Systems Laboratory (CSL) develops standards and guidelines, provides technical assistance, and conducts research for computers and related telecommunications systems to achieve more effective utilization of Federal information technology resources. CSL's responsibilities include development of technical, management, physical, and administrative standards and guidelines for the cost-effective security and privacy of sensitive unclassified information processed in Federal computers. CSL assists agencies in developing security plans and in improving computer security awareness training. This Special Publication 500 series reports CSL research and guidelines to Federal agencies as well as to organizations in industry, government, and academia.

National Institute of Standards and Technology Special Publication 500-203
Natl. Inst. Stand. Technol. Spec. Publ. 500-203, 111 pages (July 1992)
CODEN: NSPUE2

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1992

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, DC 20402

Table of Contents

1 INTRODUCTION

1.1 Background

This document contains test specifications for the COBOL Intrinsic Functions Module of the Federal Information Processing Standard (FIPS) Programming Language COBOL, FIPS PUB 21-3 (ANSI X3.23-1985, and Addendum ANSI X3.23A-1989)

The testing of language processors to determine the degree to which they conform to FIPS may be required by the Government departments and agencies in accordance with the FIPS, the Federal Information Resources Management Regulation (FIRMR) 201.13 and 201.39, and the associated Federal ADP and Telecommunications Standards Index. As part of its mission, the Computer Systems Laboratory (CSL) is responsible for providing language processor validations for FIPS in support of Government departments' and agencies' procurement requirements.

A validation service has been established in fulfillment of this responsibility. Results of validations, or validation certificates from other validation organizations may be accepted as the basis for CSL Certificate of Validation provided that all CSL requirements are met. For Ada validations, CSL coordinates its validation responsibility with the Department of Defense, Ada Joint Program Office (AJPO).

The CSL presently provides validation services for the following languages:

- Ada
- C
- COBOL
- Fortran
- MUMPS
- Pascal
- SQL

CSL is also working on future validation services for:

- BASIC

1.2 Purpose

This document serves as a reference manual and as a user's guide for the COBOL Intrinsic Function Module Tests in the 1985 COBOL Compiler Validation System (CCVS). The tests are used by the National Institute of Standards and Technology (NIST) to test COBOL implementations for conformance to FIPS PUB 21-3, COBOL (ANSI X3.23-1985, and Addendum ANSI X3.23A-1989).

ANSI document number X3.23A-1989, "INTRINSIC FUNCTION ADDENDUM TO AMERICAN NATIONAL STANDARD COBOL X3.23-1985" proposed the incorporation of 42 new library functions into the standard. The functions are:

1) ACOS
2) ANNUITY
3) ASIN
4) ATAN
5) CHAR
6) COS
7) CURRENT-DATE
8) DATE-OF-INTEGER
9) DAY-OF-INTEGER
10) FACTORIAL
11) INTEGER
12) INTEGER-OF-DATE
13) INTEGER-OF-DAY
14) INTEGER-PART
15) LENGTH
16) LOG
17) LOG10
18) LOWER-CASE
19) MAX
20) MEAN
21) MEDIAN
22) MIDRANGE
23) MIN
24) MOD
25) NUMVAL
26) NUMVAL-C
27) ORD
28) ORD-MAX
29) ORD-MIN
30) PRESENT-VALUE
31) RANDOM
32) RANGE
33) REM
34) REVERSE
35) SIN
36) SQRT
37) STANDARD-DEVIATION
38) SUM
39) TAN
40) UPPER-CASE
41) VARIANCE
42) WHEN-COMPILED

This document is based on FIPS PUB 21-3. About 99% of the specifications have been translated into COBOL code by the National Computer Center (NCC) in England.

Tests are divided into two major categories' Simple Tests and Complex Tests. A Simple test uses a single entity as the argument, i.e., a constant or a literal by itself. Complex tests take as an argument entities such as expressions or other Intrinsic Functions. The nature of arguments for the Complex tests will most likely have an effect on the accuracy of the expected value. This effect may be compensated by allowing a greater error margin for such tests.

2 GLOSSARY of TERMS

Following is a list of terms used throughout the specifications.

2.1 **Function** A temporary data item whose value is determined by invoking a mechanism provided by the implementor at the time the function is referenced during the execution of the statement.

2.2 **Variable** A data item whose value may be changed by execution of the object program. A variable used in an arithmetic-expression must be a numeric elementary item.

2.3 **Literal** A Character-string whose value is implied by an ordered set of characters of which the literal is composed or by specification of a reserved word which references a figurative constant.

2.4 **Numeric Literal** A Character-string whose characters are selected from the digits '0' through '9', the plus sign, the minus sign, and the decimal point.

2.5 **Nonnumeric Literal** A Character-string delimited at the beginning and at the end by the separator quotation mark.

2.6 **Domain** The set of values that the function uses as input.

2.7 **Range** The set of values that the function returns based on the domain values.

2.8 **Integer**

A numeric literal or a numeric data item that does not include any digit position to the right of the assumed decimal point. When the term 'integer' appears in general formats, integer must not be a numeric data item, and must not be signed, nor zero unless explicitly allowed by the rules of that format.

2.9 **Noninteger**

A numeric literal or numeric data item that includes one or more digits to the right of the assumed decimal point.

2.10 **FXN**

Function Name

2.11 **Verb**

A word that expresses an action to be taken by a COBOL compiler or object program.

2.12 **Standard Position**

The position of an angle with its vertex at the origin of a rectangular-coordinate system and its initial side coinciding with the positive x-axes.

3 CONVENTIONS

3.1 Numbering System

Naming conventions in these specifications follow the conventions used in the CCVS. A routine name consists of 6 characters whose meaning is as follows:

First two Characters --- "IF" which identifies the Intrinsic Functions module.

Third Character --- 1 which identifies level 1 of the IF module.

Fourth & Fifth Characters --- XX, A unique sequential number that identifies each one of the test programs.

Sixth Character --- "A", which indicates that this is an Automatic test.

For example IF102A is the second of a series of several programs which tests level 1 of the Intrinsic Functions module.

3.2 Number of tests

The section labeled "Specific Features to be tested" (see sec. 5) specifies which particular feature is to be tested. Only one test for each specific feature is expected to be coded. The total number of tests is 756.

3.3 References

All page and section references in section 5 refers to ANSI X3.23A-1989.

4 REQUIREMENTS

4.1 Functionality Tests

All tests in these specifications are of a functional character. The main purpose of the tests is to determine whether or not language processors accept the standard syntax of Intrinsic Function-identifiers, and with the specified arguments, produce implementation-defined return values representative of correct implementation of the Intrinsic Function item. To allow for differences in implementation-defined techniques in computing the return values, a relative error of .00002 or .00004 in each direction from the theoretically correct value is specified for most numeric or integer functions. The following formula illustrates that schema.

```
max-range = (return-value * 1.00002)
min-range = (return-value * 0.99998)
```

In some areas where the function is very unstable and it is increasing in the vicinity of the argument the relative error is applied to the arguments instead, as described by the following formula:

```
max-range = function(argument * 1.00002)
min-range = function(argument * 0.99998)
```

The test report should list the range (maximum and minimum values) for the expected answer. The test should be marked as "FAIL" if the computed value is outside the specified error range of the expected answer.

In some cases, where relative error is not practical, an absolute error schema is used to compute the ranges as illustrated by the following formula:

```
max-range = (return-value + 0.00002)
min-range = (return-value - 0.00002)
```

The formulas presented above assume the value returned from the function is positive. When the value is negative, the max-range and the min-range are switched.

4.2 Intermediate Values

All intermediate values carry a precision of 10 decimal places for those tests in which the argument is: 1) Not a single entity; 2) The function is used with other functions (for numeric and integer functions only); and 3) The function is used as part of an expression (for numeric and integer functions only). It is recommended that implementors use 10 decimal digits although it is not required by the standard.

4.3 Expected Value

The expected value holds a precision of 6 decimal places, with the following exceptions:

1) IF136A - Simple Tests, subtest I, L: precision = 7 decimal places
2) IF139A - Complex Tests, subtest K: precision = 7 decimal places

Implementors are free to use a higher precision for the computed value on all integer/numeric functions.

4.4 Error Margin

The expected relative error for the functions for which the standard refers to its return value as an "approximation" is .00002 for Simple Tests and .00004 for Complex Tests. A .00002 relative error is also allowed for other numeric functions whose arguments involve the calculation/manipulation of intermediate results and tests involving fractional digits.

Those tests are:

1) ACOS
2) ANNUITY
3) ASIN
4) ATAN
5) COS
6) INTEGER (some tests)
7) INTEGER PART (some tests)
8) LOG
9) LOG10
10) MAX (Complex tests and some Simple tests)
11) MEAN (Complex tests and some Simple tests)
12) MEDIAN (Complex tests and some Simple tests)
13) MIDRANGE (Complex tests and some Simple tests)
14) MIN (Complex tests and some Simple tests)

15) MOD (Complex tests)
16) NUMVAL (some tests)
17) NUMVAL-C (some tests)
18) PRESENT-VALUE
19) RANGE (Complex tests and some Simple tests)
20) REM (Complex tests and some Simple tests)
21) SIN
22) SQRT
23) STANDARD-DEVIATION
24) SUM (Complex tests and some Simple tests)
25) TAN
26) VARIANCE

Regardless of error considerations no error margin is allowed beyond the range of the function.

The following functions do not allow any error margin for the expected value:

1) DATE-OF-INTEGER
2) DAY-OF-INTEGER
3) FACTORIAL
4) INTEGER (Tests not involving fractional digits)
5) INTEGER-OF-DATE
6) INTEGER-OF-DAY
7) INTEGER-PART (Tests not involving intermediate arithmetic operations)
8) LENGTH
9) MAX (Simple Tests not involving fractional digits)
10) MEAN (Simple Tests not involving fractional digits)
11) MEDIAN (Simple Tests not involving fractional digits)
12) MIDRANGE (Simple Tests not involving fractional digits)
13) MIN (Simple Tests not involving fractional digits)
14) MOD (Simple Tests not involving fractional digits)
15) NUMVAL (Tests not involving fractional digits)
16) NUMVAL-C (Tests not involving fractional digits)
17) ORD
18) ORD-MAX
19) ORD-MIN
20) RANGE (Simple Tests not involving fractional digits)
21) REM (Simple Tests not involving fractional digits)
22) SUM (Simple Tests not involving fractional digits)

The computed value for the following functions should be equal to, or lie between, the specific limits given for that particular test.

1) CURRENT-DATE
2) CHAR
3) LOWER-CASE
4) RANDOM
5) REVERSE

6) UPPER-CASE
7) WHEN-COMPILED

4.5 Angles Measurement

All angle measures are given in radians, and all angles are assumed to be in standard position.

4.6 Statements Structure

The PERFORM statement, where specified, should be used only as described in section "E" of each test set as the statement involves looping, which is different for each function depending on its range and domain.

4.7 Collating Sequence

The PROGRAM COLLATING SEQUENCE clause should be specified with an alphabet-name defined with the STANDARD-1 option for the MAX, MIN, ORD-MAX, ORD-MIN, CHAR, ORD function test programs.

4.8 Additional Information

Additional information regarding arguments and returned values for the functions described below can be found in document # X3.23A-1989.

5 TESTS DESCRIPTION

5.1 IF101A

a. Features Tested

This program tests the Intrinsic Function ACOS, which returns a numeric value in radians that approximates the arcosine of argument-1. The type of this function is numeric. The valid domain is -1 <= argl <= 1 and valid range is >=0 and <= Pi. The type of argument-1 must be of class numeric. The returned value is the approximation of the arccosine of argument-1.

FUNCTION ACOS (argl)

b. Reference

Page A-33
Section 2.5

c. Number of tests

26

d. Variables

```
A PICTURE S9(5)V9(5)    VALUE  -0.00004
B PICTURE S9(10)        VALUE  4
C PICTURE S9(10)        VALUE  100000
D PICTURE S9(10)        VALUE  1000
PI PICTURE S9V9(17)     VALUE  3.141592654
ARG1 PICTURE S9V9(17)   VALUE  0.00
ARR                     VALUE  "40537"
      IND OCCURS 5 TIMES PICTURE 9
TEMP PICTURE S9(5)V9(5)
```

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, must be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL ACOS(arg1) < 1

      procedure-name-1
            ...
            arg1 = arg1 + .25
```

Identifier-2 must never be used as a function invocation.
Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function.
Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Simple Tests (relative error = .00002)

Specific features to be tested	Arguments (arg1)	Expected Answer
a) Z = 1	(1.0)	>= 0.000000 <= 0.000020
b) Z = 1/2	(0.5)	>= 1.04718 <= 1.04722
c) Z = 0	(0)	>= 1.57076 <= 1.57082

d) Z = -1	(-1)	>= 3.14153 <= 3.14165
e) Z = values close to 1	(.999)	>= 0.044276 <= 0.045170
f) Z = values close to 1/2	(.49)	>= 1.05868 <= 1.05872
g) Z = values close to 0	(.001)	>= 1.56976 <= 1.56982
h) Z = values close to -1	(-.999)	>= 3.09680 <= 3.09692
i) Z = a low magnitude non-integer variable	(A)	>= 1.57080 <= 1.57086
j) Z = a low magnitude non-integer constant	(.00002)	>= 1.57074 <= 1.57080

Complex Tests (relative error = .00004)

Specific features to be tested	Arguments (arg1)	Expected Answer
a) Z = 1/sqrt(2)	(1/sqrt(2))	>= 0.785367 <= 0.785429
b) Z = sqrt(3)/2	(sqrt(3)/2)	>= 0.523577 <= 0.523619
c) Z = expr. with value close to or equal to 0	(1 - 1.01)	>= 1.58073 <= 1.58085
d) Z = expr. with value close to or equal to 1	(1.98 / 2)	>= 0.141533 <= 0.141545
e) Z = expr. with value close to or equal to 1/2	(0.2 + 0.29)	>= 1.05866 <= 1.05874
f) Z = expr. with value close to or equal to -1	(0.99 * -1)	>= 2.99993 <= 3.00017
g) Z = a subscripted variable	(IND(B)-2)	>=-0.00004 <= 0.00004
h) Z = a subscripted constant	(IND(5) / 9)	>= 0.679646 <= 0.679700
i) Z = an integer expression using constants only	(4 - 3)	>= 0.000000 <= 0.000040
j) Z = an integer expression using variables only	(C / C)	>= 0.000000 <= 0.000004
k) Z = a non-integer expression using constants only	(0.25 * 1)	>= 1.31806 <= 1.31816
l) Z = an integer expression using constants and and variables	((D / D) - 1)	>= 1.57073 <= 1.57085
m) Z = a non-integer expression using variables and constants	(PI - 4)	>= 2.60285 <= 2.60305

n) ACOS function used recursively i.e., ACOS(ACOS(X))), where X may be a variable and/or an expression	TEMP=ACOS(ACOS(D/D))	TEMP: >= 1.57073 <= 1.57085
o) The function ACOS applied twice within an expression	TEMP=ACOS(D/D)+ACOS(D/D)	TEMP: >= 0.000000 <= 0.000040

5.2 IF102A

a. Features Tested

This program tests the Intrinsic Function ANNUITY, which returns a numeric value that approximates the ratio of an annuity paid at the end of each period for the number of periods specified by argument-2 to an initial investment of one. Interest is earned at the rate specified by argument-1 and it is applied at the end of the period, before the payment. The type of this function is numeric. Argument-1 must be of class numeric. The returned value depends on the value of argument-1 as follows:

```
if arg1 = 0:
   1 / arg2
if arg1 <> 0:
   arg1 / (1 - (1 + arg1) ** (- arg2))
```

Arg1 = interest rate and must be a value >= 0.
Arg2 = number of periods and must be a positive integer.

FUNCTION ANNUITY (arg1 arg2)

b. Reference

Page A-34
Section 2.6

c. Number of tests

13

d. Variables

```
A PICTURE S9(10)          VALUE  4
B PICTURE S9(5)V9(5)      VALUE  .25
C PICTURE S9(10)          VALUE  10
D PICTURE S9(10)          VALUE  100
```

```
ARG2 PICTURE S9(10)     VALUE  1
ARR                     VALUE  "40537"
      IND OCCURS 5 TIMES PICTURE 9
TEMP PICTURE S9(5)V9(5)
```

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL FUNCTION ANNUITY(0, arg2)
   < .25

      procedure-name-1
            ...
            arg2 = arg2 + 1
```

Identifier-2 must never be used as a function invocation.
Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function.
Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Simple Tests (relative error = .00002)

Specific features to be tested	Arguments (arg1)	Expected Answer
a) arg1 = 0; arg2 = const	(0, 4)	>= 0.249995 <= 0.250005
b) arg1 = non-integer const	(2.9, 4)	>= 2.91252 <= 2.91264
c) arg1 = non-integer const	(.09, A)	>= 0.308663 <= 0.308675
d) arg1 = non-integer var	(B, 2)	>= 0.694430 <= 0.694458
e) arg1 = non-integer var	(B, 4)	>= 0.423434 <= 0.423450
f) arg1 = integer var	(A, 9)	>= 3.99992 <= 4.00008

g) argl = integer const	(5, 5)	>= 5.00054 <= 5.00074
h) argl, arg2 subscripted values	(IND(1), IND(A))	>= 4.03217 <= 4.03233

Complex Tests (relative error = .00004)

Specific features to be tested	Arguments (argl)	Expected Answer
a) argl = expr with vars, consts arg2 = constant	(B / 2, 8)	>= 0.204824 <= 0.204840
b) ANNUITY function that invokes itself	TEMP=ANNUITY(ANNUITY(0, 3), 3))	TEMP: >= 0.576553 <= 0.576599
c) ANNUITY function used as part of an expression	TEMP=ANNUITY(0,2) + 5	TEMP: >= 4.49978 <= 5.50022
d) ANNUITY function used twice within an expression	TEMP=ANNUITY(0,2)+ ANNUITY(0,2)	TEMP: >= 0.999960 <= 1.00004

5.3 IF103A

a. Features Tested

This program tests the Intrinsic Function ASIN, which returns a numeric value in radians that approximates the arcsine of argumentl. The type of this function is numeric. The valid domain is -1 <= argl <= 1. and range is >= -Pi/2 and <= +Pi/2. Argument-1 must be of class numeric. The returned value is the approximation of the arcsine of argument-1.

FUNCTION ASIN (argl)

b. Reference

Page A-35
Section 2.7

b. Number of tests

27

c. Variables

```
A PICTURE S9(5)V9(5)     VALUE  -0.00004
B PICTURE S9(10)         VALUE  2
```

```
C PICTURE S9(10)         VALUE  100000
D PICTURE S9(10)         VALUE  1000
PI PICTURE S9V9(17)      VALUE  3.141592654
ARG1 PICTURE S9(10)      VALUE  1
ARR                      VALUE  "40537"
      IND OCCURS 5 TIMES PICTURE 9
TEMP PICTURE S9(5)V9(5)
```

d. Statements structure

At least one of the following COBOL statements, excluding PERFORM, be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL FUNCTION ASIN(argl) < 0

      procedure-name-1
            ...
            argl = argl - .25
```

Identifier-2 must never be used as a function invocation.
Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function.
Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Simple Tests (relative error = .00002)

Specific features to be tested	Arguments (argl)	Expected Answer
a) Z = 1	(1.0)	>= 1.57076 <= 1.57080
b) Z = 1/2	(0.5)	>= 0.523588 <= 0.523609
c) Z = 0	(0)	>=-0.000020 <= 0.000020
d) Z = -1	(-1)	>=-1.57080 <=-1.57076
e) Z = values close to 1	(.999)	>= 1.52563 <= 1.52652

f) Z = values close to 1/2	(.49)	>= 0.512079 <= 0.512099
g) Z = -1/2	(-0.5)	>=-0.523609 <=-0.523588
h) Z = values close to -1	(-.999)	>=-1.52652 <=-1.52563
i) Z = PI/4	(PI/4)	>= 0.903321 <= 0.903357
j) Z = -PI/4	(-PI/4)	>=-0.903357 <=-0.903321
k) Z = a variable subscripted variable	(IND(B))	>=-0.000020 <= 0.000020

Complex Tests (relative error = .00004)

Specific features to be tested	Arguments (argl)	Expected Answer
a) Z = 1/sqrt(2)	(1/sqrt(2))	>= 0.785367 <= 0.785429
b) Z = sqrt(3)/2	(sqrt(3)/2)	>= 1.04715 <= 1.04723
c Z = expr. with value close to or equal to -1/2	(-0.2 + -0.29)	>=-0.512110 <=-0.512069
d) Z = expr. with value close to or equal to 1	(1.98 / 2)	>= 1.42919 <= 1.42931
e) Z = expr. with value close to or equal to 1/2	(0.2 + 0.29)	>= 0.512069 <= 0.512110
f) Z = expr. with value close to or equal to -1	(0.99 * -1)	>=-1.42931 <=-1.42919
g) Z = a constant subscripted variable	(IND(3) / 8)	>= 0.675104 <= 0.675158
h) Z = an integer expression using constants only	(4 - 3)	>= 1.57073 <= 1.57080
i) Z = an integer expression using variables only	(C - C)	>=-0.000040 <= 0.000040
j) Z = a non-integer expression using constants only	(0.25 * 1)	>= 0.252670 <= 0.252690
k) Z = a non-integer expression using variables only	(1 / PI)	>= 0.323933 <= 0.323959
l) Z = an integer expression using constants and variables	((D / D) - 1)	>=-0.000040 <= 0.000040
m) Z = a non-integer expression using variables and constants	(PI - 4)	>=-1.03219 <=-1.03211

n) ASIN function used recursively i.e., ASIN(ASIN(X))), where X may be a variable and/or expression	TEMP = ASIN(ASIN(PI-3))	TEMP: >= 0.142546 <= 0.142558
o) The function ASIN used twice within an expression	TEMP = ASIN(.6) + ASIN(.6)	TEMP: >= 1.28695 <= 1.28705

5.4 IF104A

a. Features Tested

This program tests the Intrinsic Function ATAN, which returns a numeric value in radians that approximates the arctangent of argument-1. The type of this function is numeric. The valid range is > -Pi/2 and < +Pi/2. Argument-1 must be of numeric class. The returned value is the approximation of the arctangent of argument-1.

FUNCTION ATAN (arg1)

b. Reference

Page A-36
Section 2.8

c. Number of tests

27

d. Variables

```
A PICTURE S9(5)V9(5)     VALUE  -0.00004
B PICTURE S9(10)         VALUE  2
C PICTURE S9(10)         VALUE  100000
D PICTURE S9(10)         VALUE  1000
PI PICTURE S9V9(17)      VALUE  3.141592654
ARG1 PICTURE S9V9(17)    VALUE  1.00
SQRT3 PICTURE S9V9(17)   VALUE  1.732050808
ARR                      VALUE  "40537"
      IND OCCURS 5 TIMES PICTURE 9
TEMP PICTURE S9(5)V9(5)
```

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL FUNCTION ATAN(argl) < 0

      procedure-name-1
            ...
            argl = argl -.25
```

Identifier-2 must never be used as a function invocation.
Expression-1 and expression-2 refers to expressions in which one of the operands is an Intrinsic Function.
Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Simple Tests (relative error = .00002)

Specific features to be tested	Arguments (argl)	Expected Answer
a) Z = 1	(1.0)	>= 0.785382 <= 0.785414
b) Z = 1/2	(0.5)	>= 0.463638 <= 0.463656
c) Z = 0	(0)	>=-0.000020 <= 0.000020
d) Z = -1	(-1)	>=-0.785414 <=-0.785382
e) Z = values close to 1	(.999)	>= 0.784881 <= 0.784913
f) Z = values close to 0	(.049)	>= 0.048959 <= 0.048961
g) Z = a low magnitude non-integer variable	(A)	>=-0.000040 <=-0.000039
h) Z = a low magnitude non-integer constant	(.00002)	>= 0.000019 <= 0.000020
i) Z = a subscripted constant	(IND(B))	>=-0.000020 <= 0.000020

Complex Tests (relative error = .00004)

Specific features to be tested	Arguments (arg1)	Expected Answer
a) Z = 1/sqrt(3)	(1/sqrt(3))	>= 0.523577 <= 0.523619
b) Z = sqrt(3)	(sqrt(3))	>= 1.04715 <= 1.04723
c) Z = values close to sqrt(3)	(SQRT3 - .001)	>= 1.04690 <= 1.04698
d) Z = values close to 1/sqrt(3)	((1 / SQRT3) - .001)	>= 0.522827 <= 0.522869
e) Z = expr. with value close to or equal to 0	(1 - 1.01)	>=-0.010000 <=-0.009998
f) Z = expr. with value close to or equal to 1	(1.98 / 2)	>= 0.780342 <= 0.780404
g) Z = expr. with value close to or equal to sqrt(3)	(SQRT3 + .01)	>= 1.04964 <= 1.04972
h) Z = expr. with value close to or equal to 1/sqrt(3)	((1 / SQRT3) + .01)	>= 0.531045 <= 0.531087
i) Z = a subscripted variable	(IND(3)/B)	>= 1.19023 <= 1.19033
j) Z = an integer expression using constants only	(4 - 3)	>= 0.785367 <= 0.785429
k) Z = an integer expression using variables only	(C - C)	>=-0.000040 <= 0.000040
l) Z = a non-integer expression using constants only	(0.25 * 1)	>= 0.244968 <= 0.244988
m) Z = a non-integer expression using variables only	(1 / PI)	>= 0.308157 <= 0.308181
n) Z = an integer expression using constants and and variables	((D / D)- 1)	>=-0.000040 <= 0.000040
o) Z = a non-integer expression using variables and constants	(PI - 4)	>=-0.709382 <=-0.709326
p) ATAN function used recusively i.e., ATAN(ATAN(X)), where X may be a variable and/or an expression	TEMP=ATAN(ATAN(PI/5))	TEMP: >= 0.511215 <= 0.511255
q) The function ATAN used twice within an expression	TEMP = ATAN(.6) + ATAN (-.6)	TEMP: >=-0.000040 <= 0.000040

5.5 IF105A

a. Features Tested

This program tests the Intrinsic Function CHAR, which returns a one-character alphanumeric value that is a character in the program collating sequence having the ordinal position equal to the value of argument-1. The type of this function is alphanumeric. Argument-1 must be an integer, whose value must be greater than zero and less than or equal to the number of positions in the collating sequence.

FUNCTION CHAR (argl)

b. Reference

Page A-37
Section 2.9

c. Number of tests

8

d. Variables

```
B PICTURE S9(10)        VALUE  37
C PICTURE S9(10)        VALUE  2
D PICTURE S9(10)        VALUE  100
ARR                     VALUE  "066037100070044"
      IND OCCURS 5 TIMES PICTURE 9(3)
TEMP PICTURE S9(5)V9(5)
```

e. Statements structure

At least one of the following COBOL statements must be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the MOVE statement where possible.

```
1) MOVE identifier-1 TO identifier-2
2) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
```

Identifier-1 refers to a function invocation.
Identifier-2 must never be used as a function invocation.
Condition-1 refers to a conditional expression for which one of its operands is an Intrinsic Function.

Specific features to be tested	Arguments (argl)	Expected Answer
a) Z = an integer constant	(37)	$
b) Z = an integer variable	(B)	$
c) Z = a subscripted integer constant	(IND(5))	+
d) Z = a subscripted integer variable	(IND(C))	$
e) Z = an integer constant	(87)	V
f) Z = an integer variable	(D)	c
g) a CHAR function used with another function	TEMP = ORD(CHAR(2))	TEMP = 2
h) a CHAR function used twice within an expression	TEMP = ORD(CHAR(4))+ ORD(CHAR(7))	TEMP = 11

5.6 IF106A

a. Features Tested

This program tests the Intrinsic Function <u>COS</u>, which returns a numeric value that approximates the cosine of an angle or arc expressed in radians, that is specified by argument-1. The type of this function is numeric. The valid range is: -1<= COS(argl)< =1. Argument-1 must be class numeric. The returned value is the approximation of the cosine of argument-1.

FUNCTION COS (argl)

b. Reference

Page A-38
Section 2.10

c. Number of tests

32

d. Variables

A PICTURE S9(5)V9(5)	VALUE	-0.00004
B PICTURE S9(5)V9(5)	VALUE	14000.105
C PICTURE S9(10)	VALUE	100000
D PICTURE S9(10)	VALUE	1000
E PICTURE S9(10)	VALUE	3
PI PICTURE S9V9(17)	VALUE	3.141592654
ARG1 PICTURE S9V9(17)	VALUE	1.00

```
ARR                     VALUE  "40537"
      IND OCCURS 5 TIMES PICTURE 9
TEMP PICTURE S9(5)V9(5)
```

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL FUNCTION COS(arg1) < 0

      procedure-name-1
            ...
            arg1 = arg1 - .25
```

Identifier-2 must never be used as a function invocation. Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function. Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Simple Tests (relative error = .00002)

Specific features to be tested	Arguments (arg1)	Expected Answer
a) Z = 0	(0)	>= 0.999980 <= 1.00000
b) Z = PI	(PI)	>=-1.00000 <=-0.999980
c) Z = -PI	(-PI)	>=-1.00000 <=-0.999980
d) Z = Values close to 0	(0.001)	>= 0.999980 <= 1.000000
e) Z = a low magnitude non-integer constant (const < .0001)	(.00009)	>= 0.999980 <= 1.000000
f) Z = a low magnitude non-integer variable (var < .0001)	(A)	>= 0.99998 <= 1.00000

g) Z = a variable subscripted variable	(IND(E))	>= 0.283656 <= 0.283668
h) Z = a constant subscripted variable	(IND(5))	>= 0.753887 <= 0.753917

Complex Tests (relative error = .00004)

Specific features to be tested	Arguments (argl)	Expected Answer
a) Z = PI/3	(PI / 3)	>= 0.499980 <= 0.500020
b) Z = PI/2	(PI / 2)	>=-0.000040 <= 0.000040
c) Z = 3PI/2	(3 * PI)/2	>=-0.000040 <= 0.000040
d) Z = -PI/3	(-PI / 3)	>= 0.499980 <= 0.500002
e) Z = -PI/2	(-PI / 2)	>=-0.000040 <= 0.000040
f) Z = -3PI/2	(3 * -PI)/ 2	>=-0.000040 <= 0.000040
g) Z = Values close to PI/2	(PI/2)-0.001	>= 0.000937 <= 0.001063
h) Z = Values close to PI/3	((PI/3) + 0.001)	>= 0.499113 <= 0.499153
i) Z = Values close to 3PI/2	((3 * PI)/ 2) + 0.001	>= 0.000811 <= 0.001189
j) Z = Expr. with value close to or equal to 0	(PI * (4 - 2) / 180)	>= 0.999350 <= 0.999430
k) Z = Expr. with value close to or equal to PI/2	(PI / 2) - (PI / 180)	>= 0.017451 <= 0.017453
l) Z = Expr. with value close to or equal to PI/3	(PI / 3) - (PI / 180)	>= 0.515017 <= 0.515059
m) Z = Expr. with value close to or equal to PI	(PI + (PI / 180))	>=-0.999887 <=-0.999807
n) Z = Expr. with value close to or equal to 3PI/2	((PI * 272) / 180)	>= 0.034898 <= 0.034900
o) Z = an integer expression using constants only	(4 / 2)	>=-0.416163 <=-0.416129
p) Z = a non-integer expression using constants only	(3 / 2)	>= 0.070734 <= 0.070740
q) Z = a non-integer expression using variables only	(PI - A)	>=-1.000000 <=-0.999960
r) Z = an integer expression using variables and constants	(D / 100)	>=-0.839105 <=-0.839037

s) Z = a non-integer expression using variables & constants	(PI / 180)	>= 0.999807 <= 0.999887
t) Z = Values close to PI	(PI - 0.001)	>=-1.000000 <=-0.999960
u) COS used as part of an expression	TEMP = COS(PI) + 1	TEMP: >=-0.000040 <= 0.000040
v) COS function used recursively i.e., COS(COS(X)), where x may be a variable or an expression	TEMP = COS(COS(2))	TEMP: >= 0.914616 <= 0.914690
w) The COS function used twice within an expression	TEMP = COS(PI)+COS(PI)	TEMP: >=-2.00008 <=-1.99992

5.7 IF107A

a. Features Tested

This program tests the Intrinsic Function <u>CURRENT-DATE</u>, which returns 21-character alphanumeric value that represents the calendar date, time of day and local time differential factor provided by the system on which the function is evaluated. The type of this function is alphanumeric. For additional information related to the returned values see pages A-39 & A-40 of X3.23A-1989, "INTRINSIC FUNCTION MODULE ADDENDUM TO AMERICAN NATIONAL STANDARD COBOL, X3.23-1985."

FUNCTION CURRENT-DATE

b. Reference

Page A-39
Section 2.11

c. Number of tests

2

d. Variables

TEMP1 PICTURE X(21)
TEMP2 PICTURE X(21)

e. Statement structure

At least one the following COBOL statements must be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the MOVE statement where possible.

```
1) MOVE identifier-1 TO identifier-2
2) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
```

Identifier-1 refers to a function invocation.
Identifier-2 must never be used as a function invocation.
Condition-1 refers to a conditional expression for which one of its operands is an Intrinsic Function.

Specific features to be tested	Arguments	Expected Answer
a) check that the range is valid	TEMP1 = CURRENT-DATE	
b) check again and make sure values are close, with a small time differential between function calls and second call returns a time later than first call	TEMP2 = CURRENT-DATE	TEMP2=>TEMP1

5.8 IF108A

a. Features Tested

This program tests the Intrinsic Function DATE-OF-INTEGER, which converts a date in the Gregorian calendar from integer date form to standard date form (YYYYMMDD). The type of this function is integer. The argument given must be a positive integer that corresponds to the number of days past December 31, 1600 in the Gregorian calendar. The returned value represents the ISO standard date equivalent of the integer specified in argument-1. The returned value is in the form (YYYYMMDD), where YYYY represents a year in the Gregorian calendar, MM represents the month of that year and DD represents the day of that month.

FUNCTION DATE-OF-INTEGER (arg1)

b. Reference

Page A-41
Section 2.12

c. Number of tests

10

d. Variables

```
A PICTURE S9(10)        VALUE  400
C PICTURE S9(10)        VALUE  300
D PICTURE S9(10)        VALUE  1
ARG1 PICTURE S9(10)     VALUE  1
ARR                     VALUE  "40537"
      IND OCCURS 5 TIMES PICTURE 9
TEMP PICTURE S9(5)V9(5)
```

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL FUNCTION
   DATE-OF-INTEGER(arg1) > 16010110

      procedure-name-1
            ...
            arg1 = arg1 + 1
```

Identifier-2 must never be used as a function invocation. Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function. Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Specific features to be tested	Arguments (arg1)	Expected Answer
a) Z = an integer constant	(1)	16010101
b) Z = an integer variable	(A)	16020204
c) Z = a constant subscripted integer variable	(IND 1))	16010104
d) Z = a variable subscripted integer variable	(IND(D))	16010104

e) Z = an integer that is exactly equal to 365*2, two years	(730)	16021231
f) Z = an integer variable	(C)	16011027
g) Z = an integer that is exactly equal to 365	(365)	16011231
h) DATE-OF-INTEGER function used as part of an expression	TEMP = DATE-OF-INTEGER (D) + 10	TEMP = 16010111
i) DATE-OF-INTEGER function used twice within an expression	TEMP = DATE-OF-INTEGER (D) + DATE-OF-INTEGER (D)	TEMP = 32020202

5.9 IF109A

a. Features tested

This program tests the Intrinsic Function DAY-OF-INTEGER, which converts a date in the Gregorian calendar from integer date form to Julian date form (YYYYDDD). The type of this function is integer. The argument must be a positive integer that corresponds to the number of days past December 31, 1600 in the Gregorian calendar. The returned value represents the Julian equivalent of the integer specified in argument-1. The returned value is in the form (YYYYDDD) where YYYY represents a year in the Gregorian calendar and DDD represents the day of that year.

FUNCTION DAY-OF-INTEGER (argl)

b. Reference

Page A-42
Section 2.13

c. Number of tests

8

d. Variables

```
A PICTURE S9(10)          VALUE  400
C PICTURE S9(10)          VALUE  365
D PICTURE S9(10)          VALUE  1
ARG1 PICTURE S9(10)       VALUE  1
ARR                       VALUE  "40537"
       IND OCCURS 5 TIMES PICTURE 9
TEMP PICTURE S9(5)V9(5)
```

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL FUNCTION
   DAY-OF-INTEGER(arg1) > 1601010

      procedure-name-1
            ...
            arg1 = arg1 + 1
```

Identifier-2 must never be used as a function invocation.
Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function.
Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Specific features to be tested	Arguments (arg1)	Expected Answer
a) Z = an integer constant	(1)	1601001
b) Z = an integer variable	(A)	1602035
c) Z = a subscripted integer constant	(IND(1))	1601004
d) Z = a subscripted integer variable	(IND(D))	1601004
e) Z = an integer equals to the number of days in one year	(C)	1601365
f) DAY-OF-INTEGER function used as part of an expression	TEMP = DAY-OF-INTEGER(D) + 10	TEMP = 1601011
g) DAY-OF-INTEGER function used twice within an expression	TEMP = DAY-OF-INTEGER(D) +DAY-OF-INTEGER(D)	TEMP = 3202002

5.10 IF110A

a. Features Tested

This program tests the Intrinsic Function <u>FACTORIAL</u>, which returns an integer that is the factorial of argument-1. The type of this function is integer. Argument-1 must be an integer greater than or equal to zero. If the value of argument-1 is zero, the value 1 is returned. If the value of argument-1 is positive, its factorial is returned.

FUNCTION FACTORIAL (arg1)

b. Reference

Page A-43
Section 2.14

c. Number of tests

9

d. Variables

```
A PICTURE S9(10)         VALUE  5
B PICTURE S9(10)         VALUE  7
ARG1 PICTURE S9(10)      VALUE  1
ARR                      VALUE  "40537"
      IND OCCURS 5 TIMES PICTURE 9
TEMP PICTURE S9(5)V9(5)
```

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL FUNCTION FACTORIAL(arg1) >
   120

      procedure-name-1
            ...
            arg1 = arg1 + 1
```

Identifier-2 must never be used as a function invocation. Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function. Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Specific features to be tested	Arguments (arg1)	Expected Answer
a) Z = 0	(0)	1
b) Z = an integer constant	(3)	6
c) Z = an integer variable	(A)	120
d) Z = a subscripted integer constant	(IND(4))	6
e) Z = a subscripted integer variable	(IND(A))	5040
f) a FACTORIAL function that invokes itself	TEMP = FACTORIAL(FACTORIAL(3))	TEMP = 720
g) a FACTORIAL function used as part of an expression	TEMP = FACTORIAL(1)+B	TEMP = 8
h) a FACTORIAL function used twice within an expression	TEMP = FACTORIAL(4) + FACTORIAL(2)	TEMP = 26

5.11 IF111A

a. Features Tested

This program tests the Intrinsic Function INTEGER, which returns the greatest integer value that is less than or equal to the argument. The type of this function is integer. Argument-1 must be of class numeric.

FUNCTION INTEGER (arg1)

b. Reference

Page A-44
Section 2.15

c. Number of tests

24

d. Variables

A PICTURE S9(10)	VALUE 500000
B PICTURE S9(10)	VALUE 1

```
E PICTURE S9(6)V9(5)    VALUE  399999.122
F PICTURE S9(5)V9(5)    VALUE  0.00032
G PICTURE S9(5)V9(5)    VALUE  4.08
H PICTURE S9(5)V9(5)    VALUE -5
I PICTURE S9(5)V9(5)    VALUE  3.4
ARG1 PICTURE S9(5)V9(5)  VALUE  4.4
ARR                     VALUE  "40537"
      IND OCCURS 5 TIMES PICTURE 9
TEMP PICTURE S9(5)V9(5)
```

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL FUNCTION INTEGER(arg1) < 0

      procedure-name-1
            ...
            arg1 = arg1 - 1
```

Identifier-2 must never be used as a function invocation.
Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function.
Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Specific features to be tested	Arguments (arg1)	Expected Answer
a) Z = 0	(0)	0
b) Z = positive integer constant	(3)	3
c) Z = positive non-integer constant	(4.578)	4
d) Z = negative integer constant	(-58)	-58
e) Z = negative non-integer constant	(-9.763)	-10
f) Z = a large magnitude integer constant	(320485)	320485
g) Z = a large magnitude non-integer constant	(230492.4828)	230492

h) Z = a low magnitude non-integer constant	(0.00032)	0
i) Z = a large magnitude integer variable	(A)	500000
j) Z = a large magnitude non-integer variable	(E)	399999
k) Z = a low magnitude integer variable	(B)	1
l) Z = a low magnitude non-integer variable	(F)	0
m) Z = a constant subscripted variable	(IND(2))	0
n) Z = a variable subscripted variable	(IND(B))	4
o) Z = an integer expression using constants only	((6/3) + 9)	11
p) Z = an integer expression using variables only	(H + B)	-4
q) Z = a non-integer expression using constants only	(6.3 - 4.2/2)	4
r) Z = a non-integer expression using variables only	((H + G) * I)	-4
s) Z = an integer expression using variables and constants	(H / 5)	-1
t) Z = low-magnitude negative non-integer constant	(-0.0000001)	-1
u) INTEGER used as part of an expression	TEMP = INTEGER(3.2)+I	TEMP: >= 6.39987 <= 6.40013
v) INTEGER function that invokes itself	TEMP = INTEGER(INTEGER (1.6))	TEMP = 1
w) The INTEGER function applied twice on an expression	TEMP = INTEGER(1.2) + INTEGER(1.6)	TEMP = 2

5.12 IF112A

a. Features Tested

This program tests the Intrinsic Function INTEGER-OF-DATE, which converts a date in the Gregorian calendar from standard date form (YYYYMMDD) to integer date form. The type of this function is integer. Argument-1 must be an integer of the form YYYYMMDD, where:

a) YYYY - represents the year in the Gregorian calendar. It must be an integer greater than 1600.
b) MM - represents a month and must be a positive integer less than 13.

c) DD - represents a day and must be a positive integer less than 32 provided that it is valid for the specified month & year combination.

The returned value is an integer that is the number of days the date represented by argument-1 succeeds December 31, 1600, in the Gregorian calendar.

FUNCTION INTEGER-OF-DATE (argl)

b. Reference

Page A-45
Section 2.16

c. Number of tests

8

d. Variables

```
A PICTURE S9(10)         VALUE  16020204
D PICTURE S9(10)         VALUE  2
ARG1 PICTURE S9(10)      VALUE  16010101
ARR                      VALUE  "1601010116020210"
      IND OCCURS 2 TIMES PICTURE 9(8)
TEMP PICTURE S9(10)
```

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL FUNCTION
   INTEGER-OF-DATE(argl) > 10

      procedure name-1
             ...
             argl = argl + 1
```

Identifier-2 must never be used as a function invocation. Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function.

Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Specific features to be tested	Arguments (arg1)	Expected Answer
a) Z = an integer constant	(16010101)	1
b) Z = an integer variable	(A)	400
c) Z = a subscripted integer constant	(IND(1))	1
d) Z = a subscripted integer variable	(IND(D))	406
e) Z = an integer equals to one year after December 31 1600	(16011231)	365
f) INTEGER-OF-DATE function used as part of an expression	TEMP=INTEGER-OF-DATE(A) + 10	TEMP=410
g) INTEGER-OF-DATE function used twice within an expression	TEMP=INTEGER-OF-DATE(A) + INTEGER-OF-DATE(A)	TEMP=800

5.13 IF113A

a. Features Tested

This program tests the Intrinsic Function INTEGER-OF-DAY, which converts a date in the Gregorian calendar year from Julian date form (YYYYDDD) to integer date form. The type of this function is integer. Argument-1 must be an integer of the form YYYYDDD where:

a) YYYY - represents the year in the Gregorian calendar. It must be an integer greater than 1600.
b) DDD - represents the day of the year. It must be an integer less than 367 provided that is valid for the year specified.

The returned value is an integer that is the number of days the date represented by argument-1 succeeds December 31, 1600, in the Gregorian calendar.

FUNCTION INTEGER-OF-DAY (arg1)

b. Reference

Page A-46
Section 2.17

c. Number of tests

8

d. Variables

```
A PICTURE S9(10)        VALUE  1602035
D PICTURE S9(10)        VALUE  2
ARG1 PICTURE S9(10)     VALUE  1601001
ARR                     VALUE  "16010011602035"
      IND OCCURS 2 TIMES PICTURE 9(7)
TEMP PICTURE S9(10)
```

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL FUNCTION
   INTEGER-OF-DAY(arg1) > 10

      procedure-name-1
            ...
            arg1 = arg1 + 1
```

Identifier-2 must never be used as a function invocation. Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function. Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Specific features to be tested	Arguments (arg1)	Expected Answer
a) Z = an integer constant	(1601001)	1
b) Z = an integer variable	(A)	400
c) Z = a constant subscripted integer variable	(IND(1))	1
d) Z = a variable subscripted integer variable	(IND(D))	400

e) Z = an integer that is equal to the number of days after one year past December 31, 1600	(1601365)	365
f) INTEGER-OF-DAY function used as part of an expression	TEMP= INTEGER-OF-DAY(A) + 10	TEMP = 410
g) INTEGER-OF-DAY function used twice within an expression	TEMP = INTEGER-OF-DAY(A) + INTEGER-OF-DAY(A)	TEMP = 800

5.14 IF114A

a. Features Tested

This program tests the Intrinsic Function INTEGER-PART, which returns an integer that is the integer portion of argument-1. The type of this function is integer. Argument-1 must be class numeric. If the value of argument-1 is zero, the returned value is zero. If the value of argument-1 is positive, the returned value is the greatest integer less than or equal to the value of argument-1. If the value of argument-1 is negative, the returned value is the least integer greater than or equal to the value of argument-1.

FUNCTION INTEGER-PART (arg1)

b. Reference

Page A-47
Section 2.18

c. Number of tests

24

d. Variables

```
A PICTURE S9(10)         VALUE  500000
B PICTURE S9(10)         VALUE  1
E PICTURE S9(5)V9(5)     VALUE  399999.122
F PICTURE S9(5)V9(5)     VALUE  0.00032
G PICTURE S9(5)V9(5)     VALUE  4.08
H PICTURE S9(5)V9(5)     VALUE -5
I PICTURE S9(5)V9(5)     VALUE  3.4
ARG1 PICTURE S9(5)V9(5)  VALUE  4.4
ARR                      VALUE  "40537"
      IND OCCURS 5 TIMES PICTURE 9
TEMP PICTURE S9(5)V9(5)
```

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL FUNCTION
   INTEGER-PART(arg1) < 0

      procedure-name-1
             ...
             arg1 = arg1 - 1
```

Identifier-2 must never be used as a function invocation. Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function. Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Specific features to be tested	Arguments (arg1)	Expected Answer
a) Z = 0	(0)	0
b) Z = positive integer constant	(3)	3
c) Z = positive non-integer constant	(4.578)	4
d) Z = negative integer constant	(-58)	-58
e) Z = negative non-integer constant	(-9.763)	-9
f) Z = a large magnitude integer constant	(320485)	320485
g) Z = a large magnitude non-integer constant	(230492.4828)	230492
h) Z = a low magnitude non-integer constant	(0.00032)	0
i) Z = a large magnitude integer variable	(A)	500000
j) Z = a large magnitude non-integer variable	(E)	399999
k) Z = a low magnitude integer variable	(B)	1

l) Z = a low magnitude non-integer variable	(F)	0
m) Z = a subscripted constant	(IND(1))	4
n) Z = a subscripted variable	(IND(B))	4
o) Z = an integer expression using constants only	((6/3) + 9)	11
p) Z = an integer expression using variables only	(H + B)	-4
q) Z = a non-integer expression using constants only	(6.3 - (4.2/2))	4
r) Z = a non-integer expression using variables only	((H + G) * I)	-3
s) Z = an integer expression using variables and constants	(H / 5)	-1
t) Z = a low-magnitude negative non-integer constant	(-0.0001)	0
u) INTEGER-PART used as part of an expression	TEMP=INTEGER-PART(3.2) + I	TEMP >= 6.39987 <= 6.40013
v) INTEGER-PART function that invokes itself	TEMP=INTEGER-PART(INTEGER-PART(3.2))	TEMP = 3
w) The INTEGER-PART function applied twice on an expression	TEMP=INTEGER-PART(3.2)+ INTEGER-PART(1.3)	TEMP = 4

5.15 IF115A

a. Features Tested

This program tests the Intrinsic Function LENGTH, which returns an integer equal to the length of the argument in character positions. the type of the function is integer. Argument-1 may be a nonnumerical literal or a data item of any class or category. For additional information related to arguments and returned values see page A-48 of X3.23A-1989, "INTRINSIC FUNCTION MODULE ADDENDUM TO AMERICAN NATIONAL STANDARD COBOL, X3.23-1985."

FUNCTION LENGTH (argl)

b. Reference

Page A-48
Section 2.19

c. Number of tests

8

d. Variables

```
K PICTURE A(1)          VALUE "D"
M PICTURE A(17)         VALUE "longstringofchars"
N PICTURE A(3)          VALUE "abc"
C PICTURE S9(10)
```

e. Statements structure

At least one of the following COBOL statements must be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
```

Identifier-2 must never be used as a function invocation.
Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function.
Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Specific features to be tested	Arguments (arg1)	Expected Answer
a) Z = constant string of 1 character	("A")	1
b) Z = constant string with a large number of characters	("ABCDEFGHIJKLMNOPQRST")	20
c) Z = constant string with a low number of characters	("ABCD")	4
d) Z = variable string of 1 character	(K)	1
e) Z = variable string with a large number of characters	(M)	17
f) Z = variable string with a low number of characters	(N)	3
g) the LENGTH function used as part of an expression	C = LENGTH(N) + 2*	C = 5
h) the LENGTH function used twice within an expression	C = LENGTH(N) + LENGTH(N)	C = 6

5.16 IF116A

a. Features Tested

This program tests the Intrinsic Function LOG, which returns a numeric value that approximates the logarithm to the base e (natural log) of argument-1. The type of this function is numeric. Argument-1 must be class numeric and must be greater than zero. The returned value is the approximation of the logarithm to the base e of argument-1.

FUNCTION LOG (arg1)

b. Reference

Page A-49
Section 2.20

c. Number of tests

26

d. Variables

```
A PICTURE S9(10)         VALUE  600000
B PICTURE S9(10)         VALUE  7
C PICTURE S9(10)         VALUE -4
D PICTURE S9(10)         VALUE 10
E PICTURE S9(1)V9(9)     VALUE 2.718281828
F PICTURE S9(5)V9(5)     VALUE 32000.8
G PICTURE S9(5)V9(5)     VALUE  .00002
H PICTURE S9(5)V9(5)     VALUE -5.3
ARG1 PICTURE S9(5)V9(5)  VALUE  1.00
ARR                      VALUE  "40537"
      IND OCCURS 5 TIMES PICTURE 9
TEMP PICTURE S9(10)
```

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2

```
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL FUNCTION LOG(arg1) > 1

      procedure-name-1
            ...
            arg1 = arg1 + .2
```

Identifier-2 must never be used as a function invocation. Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function. Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Simple Tests (relative error = .00002)

Specific features to be tested	Arguments (arg1)	Expected Answer
a) Z = e	(E)	>= 0.999980 <= 1.00002
b) Z = 1	(1)	>=-0.000020 <= 0.000020
c) Z = values close to 1	(.999)	>=-0.001020 <=-0.000980
d) Z = values close to 0	(.001)	>=-6.90789 <=-6.90761
e) Z = a large magnitude integer constant	(10000)	>= 9.21015 <= 9.21524
f) Z = a large magnitude non-integer constant	(3029.48)	>= 8.01598 <= 8.01630
g) Z = a low magnitude non-integer constant	(.00005)	>=-9.90368 <=-9.90328
h) Z = a large magnitude integer variable	(A)	>= 13.3044 <= 13.3050
i) Z = a large magnitude non-integer variable	(F)	>= 10.3733 <= 10.3737
j) Z = a low magnitude non-integer variable	(G)	>=-10.8199 <=-10.8195
k) Z = a subscripted constant	(IND(4))	>= 1.09859 <= 1.09863

Complex Tests (relative error = .00004)

Specific features to be tested	Arguments (arg1)	Expected Answer
a) Z = values close to e	(E + .001)	>= 1.00032 <= 1.00040
b) Z = expr with value close to 0	(1 / 10)	>=-2.30267 <=-2.30249
c) Z = expr. with value close to or equal to e	(E - .1)	>= 0.962479 <= 0.962556
d) Z = expr. with value close to or equal to 1	(1 - .1)	>=-0.105401 <=-0.105321
e) Z = a subscripted variable	(IND(D - 5))	>= 1.94583 <= 1.94599
f) Z = an integer expression using constants only	(2 * 10)	>= 2.99561 <= 2.99585
g) Z = an integer expression using variables only	(B + C)	>= 1.09857 <= 1.09865
h) Z = a non-integer expression using constants only	(3.2 / 1.7)	>= 0.632497 <= 0.632547
i) Z = a non-integer expression using variables only	(E - H)	>= 2.08164 <= 2.08180
j) Z = an integer expression using variables and constants	(B - 2)	>= 1.60937 <= 1.60949
k) Z = a non-integer expression using variables and constants	(E + 1.7)	>= 1.48569 <= 1.48581
l) LOG used as part of an expression	TEMP = LOG(E)+4	TEMP: >= 4.99980 <= 5.00002
m) LOG function used recursively with variables & expressions	TEMP = LOG(LOG(B))	TEMP: >= 0.665702 <= 0.665756
n) The LOG function used twice on an expression	TEMP = LOG(E)+LOG(2)	TEMP: >= 1.69307 <= 1.69321

5.17 IF117A

a. Features Tested

This program tests the Intrinsic Function LOG10, which returns a numeric value that approximates the logarithm to the base 10 of argument-1. The type of this function is numeric. Argument-1 must be class numeric and must be greater than zero. The returned value is the approximation of the logarithm to the base 10 of argument-1.

FUNCTION LOG10(arg1)

b. Reference

Page A-50
Section 2.21

c. Number of tests

33

d. Variables

```
A PICTURE S9(10)        VALUE  600000
B PICTURE S9(10)        VALUE  7
C PICTURE S9(10)        VALUE -4
D PICTURE S9(10)        VALUE 10
E PICTURE S9(1)V9(9)    VALUE 2.718281828
F PICTURE S9(5)V9(5)    VALUE 32000.8
G PICTURE S9(5)V9(5)    VALUE  .00002
H PICTURE S9(5)V9(5)    VALUE -5.3
ARG1 PICTURE S9(5)V9(5) VALUE 10.00
ARR                     VALUE  "40537"
      IND OCCURS 5 TIMES PICTURE 9
TEMP PICTURE S9(10)
```

e. Statements structure

At least one of the following COBOL statements must be usedto test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL function LOG10(arg1) < .30

      procedure-name-1
            ...
            arg1 = arg1 - 1.00
```

Identifier-2 must never be used as a function invocation.
Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function.
Condition-1 refers to a conditional expression containing

arithmetic expressions for which one of its operands is an Intrinsic Function.

Simple Tests (relative error = .00002)

Specific features to be tested	Arguments (arg1)	Expected Answer
a) Z = 1	(1)	>=-0.000020 <= 0.000020
b) Z = 10	(10)	>= 0.999980 <= 1.000020
c) Z = .01	(.01)	>=-2.00004 <=-1.99996
d) Z =.001	(.001)	>=-3.00006 <=-2.99994
e) Z = 100	(100)	>= 1.99996 <= 2.00004
f) Z = values close to 10	(9.999)	>= 0.999936 <= 0.999976
g) Z = values close to 1	(1.001)	>= 0.000425 <= 0.000443
h) Z = values close to .01	(.009)	>=-2.04579 <=-2.04571
i) Z = values close to 100	(100.1)	>= 2.00039 <= 2.00047
j) Z = a large magnitude integer constant	(10000)	>= 3.99992 <= 4.00008
k) Z = a large magnitude non-integer constant	(3029.48)	>= 3.48129 <= 3.48143
l) Z = a low magnitude non-integer constant	(.00005)	>=-4.30111 <=-4.30093
m) Z = a large magnitude integer variable	(A)	>= 5.77803 <= 5.77826
n) Z = a large magnitude non-integer variable	(F)	>= 4.50507 <= 4.50525
o) Z = a low magnitude non-integer variable	(G)	>=-4.69906 <=-4.69888
p) Z = a subscripted constant	(IND(4))	>= 0.477111 <= 0.477131

Complex Tests (relative error = .00004)

Specific features to be tested	Arguments (arg1)	Expected Answer
a) Z = values close to e	(E + .001)	>= 0.434437 <= 0.434471

b) Z = expr with value close to	(1 / 10)	>=-1.00004 <=-0.999960
c) Z = expr. with value close to or equal to e	(E - .1)	>= 0.417999 <= 0.418033
d) Z = expr. with value close to or equal to 1	(1 - .1)	>=-0.045775 <=-0.045740
e) Z = expr. with value close to or equal to 10	(10 * 1.1)	>= 1.04135 <= 1.04143
f) Z = expr with value close to or equal to .01	((A * G)/ 1000)	>=-1.92090 <=-1.92074
g) Z = a subscripted variable	(IND(D - 5))	>= 0.845064 <= 0.845132
h) Z = an integer expression using constants only	(2 * 10)	>= 1.30097 <= 1.30107
i) Z = an integer expression using variables only	(B + C)	>= 0.477102 <= 0.477140
j) Z = a non-integer expression using constants only	(3.2 / 1.7)	>= 0.274690 <= 0.274712
k) Z = a non-integer expression using variables only	(E - H)	>= 0.904045 <= 0.904117
l) Z = an integer expression using variables and constants	(B - 2)	>= 0.698942 <= 0.698998
m) Z = a non-integer expression using variables and constants	(E + 1.7)	>= 0.645227 <= 0.645279
n) LOG10 used as part of an expression	TEMP = LOG10(B) + 4	TEMP: >= 4.84490 <= 4.84529
o) LOG10 function that invokes itself, using variables & expressions	TEMP=LOG10(LOG10(2))	TEMP: >=-0.521411 <=-0.521369
p) LOG10 function used twice within an expression	TEMP=LOG10(1)+LOG10(1)	TEMP: >=-0.000040 <= 0.000040

5.18 IF118A

a. Features Tested

This program tests the Intrinsic Function LOWER-CASE, which returns a character string that is the same length as argument-1 with each uppercase letter replaced by the corresponding lowercase letter. The type of this function is alphanumeric. Argument-1 must be class alphabetic or alphanumeric and must be at least one character in length. The character string returned has the same length as argument-1. If the computer character set does not include lower case letters, no changes take place in the character string.

FUNCTION LOWER-CASE (arg1)

b. Reference

Page A-51
Section 2.22

c. Number of tests

13

d. Variables

```
A PICTURE A(10)          VALUE "tumble"
B PICTURE A(10)          VALUE "WEED"
C PICTURE X(10)          VALUE "Was"
D PICTURE X(10)          VALUE "4"
E PICTURE X(10)          VALUE "And4"
TEMP PICTURE S9(10)
```

e. Statements structure

At least one of the following COBOL statements must be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the MOVE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) MOVE identifier-1 TO identifier-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
```

Identifier-1 refers to a function invocation.
Identifier-2 must never be used as function invocation.
Condition-1 refers to a conditional expression for which one of its operands is an Intrinsic Function.

Specific features to be tested	Arguments (arg1)	Expected Answer
a) Z = constant string of only lower case alphabetic characters	("figure")	figure
b) Z = constant string of only upper case alphabetic characters	("CAPS")	caps
c) Z = constant string of mixed case alphabetic chars	("highnLOW")	highnlow

d) Z = constant string of non-alphabetic characters	("95")	95
e) Z = constant string of alphabetic and non-alphabetic characters	("8isaNUMBER")	8isanumber
f) Z = variable string of all lower case alphabetic characters	(A)	tumble
g) Z = variable string of all upper case alphabetic characters	(B)	weed
h) Z = variable string of mixed case alphabetic chars	(C)	was
i) Z = variable string of non-alphabetic characters	(D)	4
j) Z = variable string of alphabetic and non-alphabetic characters	(E)	and4
k) LOWER-CASE used as part of an expression	TEMP=LENGTH(LOWER-CASE ("GIZZARD"))+2	TEMP = 9
l) LOWER-CASE used to invoke itself	LOWER-CASE(LOWER-CASE ("GIZZARD"))	gizzard
m) LOWER-CASE used twice in an expression	TEMP=LENGTH(LOWER-CASE ("HOME"))+LENGTH(LOWER-CASE("HOME"))	TEMP = 8

5.19 IF119A

a. Features Tested

This program tests the Intrinsic Function MAX, which returns the content of the argument-1 that contains the maximum value. The type of this function depends upon the argument types as follows:

Argument type	Function Type
Alphabetic	Alphanumeric
Alphanumeric	Alphanumeric
All arguments integer	Integer
Numeric (some args. may be integer)	Numeric

If more than one argument-1 is specified, all arguments must be of the same class. If more than one argument-1 has the same greatest value, the content of the argument-1 returned is the leftmost argument-1 having that value. If the type of the function is alphanumeric, the size of the returned value is the same as the size of the selected argument-1.

FUNCTION MAX (arg1 ...)

b. Reference

Page A-52
Section 2.23

c. Number of tests

24

d. Variables

```
A PICTURE S9(10)          VALUE  5
B PICTURE S9(10)          VALUE  7
C PICTURE S9(10)          VALUE -4
D PICTURE S9(10)          VALUE 10
E PICTURE S9(5)V9(5)      VALUE 34.26
F PICTURE S9(5)V9(5)      VALUE -8.32
G PICTURE S9(5)V9(5)      VALUE  4.08
H PICTURE S9(5)V9(5)      VALUE -5.3
I PICTURE X(1)            VALUE "R"
J PICTURE X(1)            VALUE "U"
M PICTURE S9(10)          VALUE  1
N PICTURE S9(10)          VALUE  3
O PICTURE S9(10)          VALUE  5
ARG1 PICTURE S9(10)       VALUE  1
ARR                       VALUE  "40537"
      IND OCCURS 5 TIMES PICTURE 9
TEMP PICTURE S9(10)
```

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, must be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) MOVE identifier-1 TO identifier-2 (*)
2) COMPUTE identifier-2 = arithmetic-expression-1
3) EVALUATE expression-1 ALSO expression-2
4) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
5) PERFORM procedure-name-1 UNTIL FUNCTION MAX(arg1,1) > 5

      procedure-name-1
            ...
            arg1 = arg1 + 1
```

Identifier-1 refers to a function invocation.
Identifier-2 must never be used as a function invocation.
Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function.
Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

* The MOVE statement should only be used on those cases where the function type is alphanumeric.

Simple Tests (relative error = 0 or .00002)

Specific features to be tested	Arguments (arg1 ...)	Expected Answer
a) Z = multiple integer constants (all > 0)	(5, 6, 10, 3, 7)	10
b) Z = multiple integer consts (with numbers < 0)	(-4, 7, 304, 3, -8)	304
c) Z = multiple non-integer constants (all > 0)	(4.3, 2.6, 7.3, 9.1)	>= 9.09982 <= 9.10018
d) Z = multiple non-integer constants (with nums < 0)	(-4.3, 10.2, -0.7, 3.9)	>= 10.1998 <= 10.2002
e) Z = multiple integer variables (all > 0)	(A, B, D)	10
f) Z = multiple integer vars (with values < 0)	(A, B, C)	7
g) Z = multiple non-integer variables (all > 0)	(E, G)	>= 34.2593 <= 34.2607
h) Z = multiple non-integer variables (with vars < 0)	(F, G, H)	>= 4.07992 <= 4.08008
i) Z = multiple variables and and constants (all int)	(A, 4, 8, -10, C, 0)	8
j) Z = multiple variables and constants (some non-int)	(4, D, H, 6.3, -2.0)	>= 9.9998 <= 10.0002
k) Z = multiple alphanumeric characters	("R", I, "I", "a")	a
l) Z = multiple alphabetic characters	("A", J, "J")	U
m) Z = a series of subscripted variables	(IND(M), IND(N), IND(O))	7
n) Z = a series of subscripted constants	(IND(1), IND(2), IND(3))	5
o) Z = ALL used as a subscript to reference a table	(IND(ALL))	7
p) Z = a series of medium to low magnitude constants	(0.03, 0.029, 0.031 0.011)	>=0.030999 <=0.031001
q) Z = a series of large magnitude constants	(31000, 310001, 78000, 29000, 12000)	310001

Complex Tests (relative error = .00002)

Specific features to be tested	Arguments (arg1 ...)	Expected Answer
a) Z = multiple integer expressions	(A*B, (C+1)/2, 3+4)	>= 34.9993 <= 35.0007
b) Z = multiple non-integer expressions	(E+4, H*2, 5+A)	>= 38.2592 <= 38.2608
c) Z = multiple integer values (all the same value)	(-7, -9+2, -7)	>=-7.00014 <=-6.99986
d) MAX function that invokes itself	TEMP=MAX(MAX(14,A), E, 50)	TEMP: >= 49.9990 <= 50.0001
e) MAX function used as part of an expression	TEMP=MAX(4,B,E)+2	TEMP: >= 36.2593 <= 36.2607
f) MAX function used twice within an expression	TEMP=MAX(A,G)+MAX(B,0)	TEMP: >= 11.9998 <= 12.0002

5.20 IF120A

a. Features Tested

This program tests the Intrinsic Function MEAN, which returns a numeric value that is the arithmetic mean (average) of its arguments. The type of this function is numeric. Argument-1 must be class numeric. The returned value is defined as the sum of the argument-1 series divided by the number of occurrences referenced by argument-1.

FUNCTION MEAN (arg1 ...)

b. Reference

Page A-53
Section 2.24

c. Number of tests

18

d. Variables

```
A PICTURE S9(10)        VALUE  5
B PICTURE S9(10)        VALUE  7
C PICTURE S9(10)        VALUE -4
```

```
D PICTURE S9(10)        VALUE 10
E PICTURE S9(5)V9(5)    VALUE 34.26
F PICTURE S9(5)V9(5)    VALUE -8.32
G PICTURE S9(5)V9(5)    VALUE  4.08
H PICTURE S9(5)V9(5)    VALUE -5.3
M PICTURE S9(10)        VALUE 320000
N PICTURE S9(10)        VALUE 650000
O PICTURE S9(10)        VALUE -430000
P PICTURE S9(10)        VALUE  1
Q PICTURE S9(10)        VALUE  3
R PICTURE S9(10)        VALUE  5
ARG1 PICTURE S9(10)     VALUE  1
ARG2 PICTURE S9(10)     VALUE  1
ARR                     VALUE  "40537"
      IND OCCURS 5 TIMES PICTURE 9
TEMP PICTURE S9(10)V9(5)
```

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL FUNCTION MEAN(arg1,arg2) >
   8

      procedure-name-1
            ...
            arg1 = arg1 + 1
            arg2 = arg2 + 1
```

Identifier-2 must never be used as a function invocation.
Expression-1 and expression-2 refers to expressions in which one of the operands is an Intrinsic Function.
Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Simple Tests (relative error = 0 or .00002)

Specific features to be tested	Arguments (argl ...)	Expected Answer
a) Z = series of integer constants	(5, -2, -14, 0)	>=-2.75006 <=-2.74995
b) Z = series of non-integer constants	(3.9, -0.3, 8.7, 100.2)	>= 28.1244 <= 28.1256
c) Z = series of integer variables	(A, B, C, D)	>= 4.49991 <= 4.50009
d) Z = series of non integer variables	(E, F, G, H)	>= 6.17988 <= 6.18012
e) Z = series of integer and non-integer constants	(10.2, -0.2, 5.6, -15.6)	>=-0.000020 <= 0.000020
f) Z = series of integer and non-integer variables	(A, B, C, D, E, F, G, H)	>= 5.33989 <= 5.34011
g) Z = a series of constants subscripted variables	(IND(2), IND(1), IND(3))	>= 2.99994 <= 3.00006
h) Z = a series of variable subscripted variables	(IND(P), IND(Q), IND(R))	>= 5.33323 <= 5.33344
i) Z = ALL used as a subscript to reference a table	(IND(ALL))	>= 3.79992 <= 3.80008
j) Z = a series of medium to low magnitude constants	(0.032, 0.019, 0.014, 0.065)	>= 0.032499 <= 0.032501
k) Z = a series of large magnitude variables	(M, N, O)	180000
l) Z = series of same values	(A, 5, A)	5

Complex Tests (relative error = .00002)

Specific features to be tested	Arguments (argl ...)	Expected Answer
a) Z = mixture of expressions variables and consts	(E, 9 * A, 0, B / 2)	>= 20.6896 <= 20.6904
b) MEAN used as part of an expression	TEMP=MEAN(A,B)+78	TEMP: >= 83.9983 <= 84.0017
c) MEAN used twice within an expression	TEMP=MEAN(A,B)+ MEAN(-2.6,-4.4)	TEMP: >= 2.49995 <= 2.50005

d) MEAN function that invokes itself	TEMP=MEAN(MEAN(4,2),6)	TEMP: >= 4.49991 <= 4.50009
e) Z = mixture of expressions and constants	(2.6 + 30, 4.5 * 2)	>= 20.7996 <= 20.8004

5.21 IF121A

a. Features Tested

This Program tests the Intrinsic Function MEDIAN, which returns the content of the argument whose value is the middle value in the list formed by arranging the arguments in sorted order. The type of this function is numeric. Argument-1 must be class numeric. If the number of occurrences referenced by argument-1 is odd, the returned value is such that at least half of the occurrences referenced by argument-1 are greater than or equal to the returned value and at least half are less than or equal. If the number of occurrences referenced by argument-1 is even, the returned value is the arithmetic mean of the values referenced by the two middle occurrences.

FUNCTION MEDIAN (arg1 ...)

b. Reference

Page A-54
Section 2.25

c. Number of tests

18

d. Variables

A PICTURE S9(10)	VALUE 5
B PICTURE S9(10)	VALUE 7
C PICTURE S9(10)	VALUE -4
D PICTURE S9(10)	VALUE 10
E PICTURE S9(5)V9(5)	VALUE 34.26
F PICTURE S9(5)V9(5)	VALUE -8.32
G PICTURE S9(5)V9(5)	VALUE 4.08
H PICTURE S9(5)V9(5)	VALUE -5.3
M PICTURE S9(10)	VALUE 320000
N PICTURE S9(10)	VALUE 650000
O PICTURE S9(10)	VALUE -430000
P PICTURE S9(10)	VALUE 1
Q PICTURE S9(10)	VALUE 3
R PICTURE S9(10)	VALUE 5
ARG1 PICTURE S9(10)	VALUE 2

```
ARG2 PICTURE S9(10)     VALUE  2
ARR                     VALUE  "40537"
     IND OCCURS 5 TIMES PICTURE 9
TEMP PICTURE S9(10)V9(5)
```

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COBOL COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL FUNCTION MEDIAN
   (1,arg1,arg2,20) > 10

      procedure-name-1
          ...
          arg1 = arg1 + 1
          arg2 = arg2 + 1
```

Identifier-2 must never be used as a function invocation.
Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function.
Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Simple Tests (relative error = 0 or .00002)

Specific features to be tested	Arguments (arg1 ...)	Expected Answer
a) Z = series of integer constants	(5, -2, -14, 0)	-1
b) Z = series of non-integer constants	(3.9, -0.3, 8.7, 100.2)	>= 6.29987 <= 6.30013
c) Z = series of integer variables	(A, B, C, D)	6
d) Z = series of non-integer variables	(E, F, G)	>= 4.07992 <= 4.08008
e) Z = series of integer and non-integer constants	(10.2, -0.2, 5.6, -15.6)	>= 2.69995 <= 2.70005
f) Z = series of integer and non-integer variables	(A, B, C, D, E, F, G)	>= 4.99990 <= 5.00010

g) Z = a series of subscripted constants	(IND(1), IND(2), IND(3))	4
h) Z = a series of subscripted variables	(IND(P), IND(Q), IND(R))	5
i) Z = ALL used as a subscript to reference a table	(IND(ALL))	4
j) Z = a series of medium to low magnitude constants	(0.065, 0.030, 0.021, 0.014)	>= 0.025499 <= 0.025501
k) Z = a series of large magnitude variables	(M, N, O)	320000
l) Z = series of same values	(A, 5, A)	5

Complex Tests (relative error = .00002)

Specific features to be tested	Arguments (arg1 ...)	Expected Answer
a) Z = mixture of expressions and constants	(2.6 + 30, 4.5 * 2)	>= 20.7996 <= 20.8004
b) Z = mixture of expressions variables and consts	(E, 9 * A, B / 2)	>= 34.2593 <= 34.2607
c) MEDIAN used as part of an expression	TEMP=MEDIAN(A, B)+78	TEMP: <= 83.9983 >= 84.0017
d) MEDIAN used twice within an expression	TEMP=MEDIAN(A,B)+ MEDIAN(-2.6, -4.4, 1)	TEMP: >= 3.39932 <= 3.40007
e) MEDIAN function that invokes itself	TEMP=MEDIAN(MEDIAN (1, 2), 3)	TEMP: >= 2.24995 <= 2.25004

5.22 IF122A

a. Features Tested

This program tests the Intrinsic Function MIDRANGE, which returns a numeric value that is the arithmetic mean (average) of the values of the minimum argument and the maximum argument. The type of this function is numeric. Argument-1 must be class numeric. The returned value is the arithmetic mean of the greatest argument-1 value and the least argument-1 value.

FUNCTION MIDRANGE (arg1 ...)

b. Reference

Page A-55
Section 2.26

c. Number of tests

18

d. Variables

```
A PICTURE S9(10)        VALUE  5
B PICTURE S9(10)        VALUE  7
C PICTURE S9(10)        VALUE -4
D PICTURE S9(10)        VALUE 10
E PICTURE S9(5)V9(5)    VALUE 34.26
F PICTURE S9(5)V9(5)    VALUE -8.32
G PICTURE S9(5)V9(5)    VALUE  4.08
H PICTURE S9(5)V9(5)    VALUE -5.3
M PICTURE S9(10)        VALUE 320000
N PICTURE S9(10)        VALUE 650000
O PICTURE S9(10)        VALUE -430000
P PICTURE S9(10)        VALUE  1
Q PICTURE S9(10)        VALUE  3
R PICTURE S9(10)        VALUE  5
ARG1 PICTURE S9(10)     VALUE  2
ARR                     VALUE  "40537"
      IND OCCURS 5 TIMES PICTURE 9
TEMP PICTURE S9(10)V9(5)
```

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL FUNCTION
   MIDRANGE(1,arg1) > 10

      procedure-name-1
            ...
            arg1 = arg1 + 1
```

Identifier-2 must never be used as a function invocation. Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function. Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Simple Tests (relative error = 0 or .00002)

Specific features to be tested	Arguments (argl ...)	Expected Answer
a) Z = series of integer constants	(5, -2, -14, 0)	>=-4.50009 <=-4.49991
b) Z = series of non-integer constants	(3.9, -0.3, 8.7, 100.2)	>= 49.9490 <= 49.9510
c) Z = series of integer variables	(A, B, C, D)	3
d) Z = series of non integer variables	(E, F, G, H)	>= 12.9697 <= 12.9703
e) Z = series of integer and non-integer constants	(10.2, -0.2, 5.6, -15.6)	>=-2.70005 <=-2.69995
f) Z = series of integer and non-integer variables	(A, B, C, D, E, F, G, H)	>= 12.9697 <= 12.9703
g) Z = mixture of expressions and constants	(2.6 + 30, 4.5 * 2)	>= 20.7996 <= 20.8004
h) Z = a series of constant subscripted variables	(IND(1), IND(2), IND(3))	>= 2.49995 <= 2.50005
i) Z = a series of variable subscripted variables	(IND(P), IND(Q), IND(R))	>= 5.49989 <= 5.50011
j) Z = ALL used as a subscript to reference a table	(IND(ALL))	>= 3.49993 <= 3.50007
k) Z = a series of medium to low magnitude constants	(0.065, 0.030, 0.020, 0.014)	>= 0.039499 <= 0.039501
l) Z = a series of large magnitude variables	(M, N, O)	110000
m) Z = series of same values	(A, 5, A)	5

Complex Tests (relative error = .00002)

Specific features to be tested	Arguments (argl ...)	Expected Answer
a) Z = mixture of expressions variables and consts	(E, 9 * A, 0, B / 2)	>= 22.4995 <= 22.5004

b) MIDRANGE used as part of an expression	TEMP=MIDRANGE(A,B)+78	TEMP: >= 83.9983 <= 84.0017
c) MIDRANGE used twice within an expression	TEMP=MIDRANGE(A,B)+ MIDRANGE(-2.6, -4.4)	TEMP: >= 2.49995 <= 2.50005
d) MIDRANGE function used recursively	TEMP=MIDRANGE(MIDRANGE (1, 3), 5)	TEMP: >= 3.49993 <= 3.50007

5.23 IF123A

a. Features Tested

This program tests the Intrinsic Function MIN, which returns the content of argument-1 that contains the minimum value. The type of this function depends upon the argument types as follows:

Argument type	Function Type
Alphabetic	Alphanumeric
Alphanumeric	Alphanumeric
All arguments integer	Integer
Numeric (some args. may be integer)	Numeric

If more than one argument is specified, all arguments must be of the same class. The returned value is the content of the argument-1 having the least value. If more than one argument-1 has the same least value, the content of argument-1 returned is the leftmost argument-1 having that value. If the type of the function is alphanumeric, the size of the returned value is the same as the size of the selected argument-1.

FUNCTION MIN (argl ...)

b. Reference

Page A-56
Section 2.27

c. Number of tests

24

d. Variables

A PICTURE S9(10)	VALUE 5
B PICTURE S9(10)	VALUE 7
C PICTURE S9(10)	VALUE -4
D PICTURE S9(10)	VALUE 10

```
E PICTURE S9(5)V9(5)    VALUE 34.26
F PICTURE S9(5)V9(5)    VALUE -8.32
G PICTURE S9(5)V9(5)    VALUE  4.08
H PICTURE S9(5)V9(5)    VALUE -5.3
I PICTURE X(1)          VALUE "R"
J PICTURE X(1)          VALUE "U"
M PICTURE S9(10)        VALUE  1
N PICTURE S9(10)        VALUE  3
O PICTURE S9(10)        VALUE  5
ARG1 PICTURE S9(10)     VALUE  15
ARR                     VALUE  "40537"
      IND OCCURS 5 TIMES PICTURE 9
TEMP PICTURE S9(10)
```

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, must be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) MOVE identifier-1 TO identifier-2 (*)
2) COMPUTE identifier-2 = arithmetic-expression-1
3) EVALUATE expression-1 ALSO expression-2
4) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
5) PERFORM procedure-name-1 UNTIL FUNCTION MIN(arg1,20) < 10

      procedure-name-1
            ...
            arg1 = arg1 - 1
```

Identifier-2 must never be used as a function invocation.
Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function.
Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

* The MOVE statement should only be used on those cases where the function type is alphanumeric.

Simple Tests (relative error = 0 .00002)

Specific features to be tested	Arguments (argl ...)	Expected Answer
a) Z = multiple integer constants (all > 0)	(5, 6, 10, 3, 7)	3
b) Z = multiple integer consts (with numbers > 0)	(-4, 7, 2304, 3, -8)	-8
c) Z = multiple non-integer constants (all > 0)	(4.3, 2.6, 7.3, 9.1)	>= 2.59995 <= 2.60005
d) Z = multiple non-integer constants (with nums < 0)	(-4.3, 10.2, -0.7, 3.9)	>=-4.30009 <=-4.29991
e) Z = multiple integer variables (all > 0)	(A, B, D)	5
f) Z = multiple integer vars (with values < 0)	(A, B, C, D)	-4
g) Z = multiple non-integer variables (all > 0)	(E, G)	>= 4.07992 <= 4.08008
h) Z = multiple non-integer variables (with vars < 0)	(E, F, G, H)	>=-8.32017 <=-8.31983
i) Z = multiple variables and and constants (all int)	(A, 4, 8, -10, C, 0)	-10
j) Z = multiple variables and constants (some non-int)	(4, D, E, 6.3, -2.0)	>=-2.00004 <=-1.99996
k) Z = multiple alphanumeric characters	("R", I, "I", "a")	I
l) Z = multiple alphabetic characters	("a", J, "J")	J
m) Z = a series of constant subscripted variables	(IND(1), IND(2), IND(3))	0
n) Z = a series of variable subscripted variables	(IND(M), IND(N), IND(O))	4
o) Z = ALL used as a subscript to reference a table	(IND(ALL))	0
p) Z = a series of medium to low magnitude constants	(0.13, 0.14, 0.15, 0.16)	>=0.129997 <=0.130003
q) Z = a series of large magnitude constants	(31000, 310001, 78000, 29000, 12000)	12000

Complex Tests (relative error = .00002)

Specific features to be tested	Arguments (argl ...)	Expected Answer
a) Z = multiple integer expressions	(A*B, (3+1)/2, 3+4)	>= 1.99996 <= 2.00004
b) Z = multiple non-integer expressions	(E+4, H*2, 5+A)	>=-10.6002 <=-10.5998

c) Z = multiple integer values (all the same value)	(-7, -9+2, (-B))	>=-7.00014 <=-6.99986
d) a MIN function that invokes itself	TEMP=MIN(MIN(14,A), E,50)*	TEMP: >= 4.99990 <= 5.00010
e) a MIN function used as part of an expression	TEMP=MIN(4,B,E)+A	TEMP: >= 8.99982 <= 9.00018
f) a MIN function used twice within an expression	TEMP=MIN(A,E)+MIN(B,0)	TEMP: >= 4.99990 <= 5.00010

5.24 IF124A

a. Features Tested

This program tests the Intrinsic Function MOD, which returns an integer value that is argument-1 modulo argument-2. The type of this function is integer. Argument-1 and Argument-2 must be integers and the value of argument-2 must not be zero. The returned value is defined as:

```
arg1-(arg2 * FUNCTION INTEGER (arg1/arg2))
```

```
FUNCTION MOD (arg1 arg2)
```

b. Reference

Page A-57
Section 2.28

c. Number of tests

21

d. Variables

A PICTURE S9(10)	VALUE 5
B PICTURE S9(10)	VALUE 7
C PICTURE S9(10)	VALUE -4
ARG2 PICTURE S9(10)	VALUE 1
TEMP PICTURE S9(10)	

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL FUNCTION MOD(5,arg2) >= 2

      procedure-name-1
              ...
              arg2 = arg2 + 1
```

Identifier-2 must never be used as a function invocation.
Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function.
Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Simple Tests (relative error = 0 or .00002)

Specific features to be tested	Arguments (arg1, arg2)	Expected Answer
a) arg1 = arg2 (both constants)	(6, 6)	0
b) arg1 > arg2 (both constants)	(11, 5)	1
c) arg1 < arg2 (both constants)	(10, 20)	10
d) arg1 = var, arg2 = var	(A, B)	5
e) arg1 = var, arg2 = const	(A, -3)	-1
f) arg1 = const, arg2 = var	(23, B)	2
g) arg1 < 0, arg2 < 0	(-11, -5)	-1
h) arg1 > 0, arg2 < 0	(11, -5)	-4
i) arg1 < 0, arg2 > 0	(-11, 5)	4

Complex Tests (relative error = .00002)

Specific features to be tested	Arguments (arg1, arg2)	Expected Answer
a) arg1 = const, arg2 = function invocation	(35, INTEGER(A*B))	>=-0.000020 <= 0.000020
b) arg1 = var, arg2 = function invocation	(A, INTEGER(B-5))	>= 0.999980 <= 1.00002
c) arg1 = function invocation arg2 = constant	(INTEGER(A-B), 9)	>= 6.99986 <= 7.00014
d) arg1 = function invocation arg2 = constant	(INTEGER((A+B)/-2), -4)	>=-2.00004 <=-1.99996

e) arg1 = function invocation arg2 = function invocation	(INTEGER(A*B),INTEGER(B-A))	>= 0.999980 <= 1.00002
f) arg1 = MOD Fxn, arg2 = var	(MOD(B, A), A)	>= 1.99996 <= 2.00004
g) arg1 = var, arg2 = MOD Fxn	(C,MOD(C, B))	>= 1.99996 <= 2.00004
h) arg1 = MOD Fxn, arg2 MOD Fxn	(MOD(9, 5),MOD(B,4))	>= 0.999980 <= 1.00002
i) The MOD function used within an expression	TEMP = MOD(23,B)+A	TEMP: >= 6.99986 <= 7.00014
j) a MOD function that invokes itself	TEMP =MOD(MOD(5,2),1)	TEMP: >=-0.000020 <= 0.000020

k) a MOD function used twice within an expression	TEMP = MOD(25,C)+
	MOD(-11,5)
		TEMP:
		>= 0.999980
		<= 1.00002

5.25 IF125A

a. Features Tested

This program tests the Intrinsic Function NUMVAL, which returns the numeric value represented by the character string specified by argument-1. Leading and trailing spaces are ignored. The type of this function is numeric. The returned value is the numeric value represented by argument-1. If the DECIMAL-POINT IS COMMA clause is specified in the SPECIAL-NAMES paragraph, a comma must be used in argument-1 rather than a decimal point.

FUNCTION NUMVAL (arg1)

Valid formats of argument-1:

```
[space] [+/-] [space] digit [ . [digit]] [space]
[space] [+/-] [space] . digit            [space]
              or
[space] digit [ . [digit]] [space] [x] [space]
[space] . digit            [space] [x] [space]
```

where x is one of +, -, CR, or DB

The total number of digits of argument-1 must not exceed 18.

b. Reference

Page A-58
Section 2.29

c. Number of tests

20

d. Variables

```
A PICTURE X(1)          VALUE "4"
B PICTURE X(5)          VALUE "203"
C PICTURE X(4)          VALUE ".429"
D PICTURE X(7)          VALUE "928.344"
E PICTURE X(9)          VALUE  "-042.3240"
F PICTURE X(7)          VALUE " 23.000"
G PICTURE X(8)          VALUE "-92924.3"****
H PICTURE X(6)          VALUE "93.21+"
I PICTURE X(9)          VALUE " 92.92  -"
TEMP PICTURE S9(5)V9(5)
```

e. Statements structure

At least one of the following COBOL statements must be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
```

Identifier-2 must never be used as a function invocation.
Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function.
Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Relative error = 0 or .00002

Specific features to be tested	Arguments (arg1)	Expected Answer
a) Z = constant string "N"	("9")	9
b) Z = constant string "NN...N"	("4738")	4738
c) Z = constant string ".NN...N"	(".935")	>= 0.934981 <= 0.935019
d) Z = constant string "NN...N.NN...N"	("385.93")	>= 385.922 <= 385.938

e) Z = constant string "SNN...N.NN...N "	("+394.2")	>= 394.192 <= 394.208
f) Z = constant string " NN...N.NN...N"	(" 939.83")	>= 939.811 <= 939.849
g) Z = constant string " S NN...N.NN...N "	(" - 4929.0323")	>=-4929.1309 <=-4928.9337
h) Z = constant string "NN...N.NN...NS"	("82.9312+")	>= 82.9295 <= 82.9329
i) Z = constant string " NN...N.NN...N S "	(" 200.0002 - ")	>=-200.0042 <=-199.9962
j) Z = variable string "N"	(A)	4
k) Z = variable string "NN...N"	(B)	203
l) Z = variable string ".NN...N"	(C)	>= 0.428991 <= 0.429009
m) Z = variable string "NN...N.NN...N"	(D)	>= 928.325 <= 928.363
n) Z = variable string "SNN...N.NN...N "	(E)	>=-42.3248 <=-42.3232
o) Z = variable string " NN...N.NN...N"	(F)	>= 22.9995 <= 23.0005
p) Z = variable string " S NN...N.NN...N "	(G)	>=-92926.16 <=-92922.44
q) Z = variable string "NN...N.NN...NS"	(H)	>= 93.2081 <= 93.2119
r) Z = variable string " NN...N.NN...N S "	(I)	>=-92.9219 <=-92.9181
s) NUMVAL used as part of an expression	TEMP=NUMVAL ("90")+10	TEMP = 100

t) NUMVAL used twice in an expression	TEMP = NUMVAL("2")+
	NUMVAL("8")
		TEMP = 10

5.26 IF126A

a. Features Tested

This function tests the Intrinsic Function NUMVAL-C, which returns the numeric value represented by the character string specified by argument-1. Any optional currency sign specified by argument-2 and any optional commas preceding the decimal point are ignored. The type of this function is numeric. The returned value is the numeric value represented by argument-1. If the DECIMAL-POINT IS COMMA clause is specified in the SPECIAL-NAMES paragraph, the functions of the comma and decimal point in argument-1 are reversed.

FUNCTION NUMVAL-C (arg1 [arg2])

Valid formats of argument 1:

[space] [+/-] [space] [cs] [space] digit [, digit] ... [. [digit]] [space]

[space] [+/-] [space] [cs] [space] . digit [space]

or

[space] [cs] [space] digit [, digit]...[. [digit]] [space] [x] [space]

[space] [cs] [space] . digit [space] [x] [space]

where x is one of +, -, CR, or DB

The total number of digits in argument-1 must not exceed 18. Argument-2, if specified must be a nonnumeric literal or alphanumeric data item. If argument-2 is not specified, the character used for cs is the currency symbol specified for the program.

b. Reference

Page A-59
Section 2.30

c. Number of tests

30

d. Variables

```
A PICTURE X(1)          VALUE "4"
B PICTURE X(5)          VALUE "203"
C PICTURE X(4)          VALUE ".429"
D PICTURE X(7)          VALUE "928.344"
E PICTURE X(9)          VALUE "-042.3240"
F PICTURE X(7)          VALUE " 23.000"
G PICTURE X(8)          VALUE "-92924.3"
H PICTURE X(6)          VALUE "93.21+"
I PICTURE X(9)          VALUE " 92.92  -"
J PICTURE X(9)          VALUE "8,848.934"
K PICTURE X(12)         VALUE "4,825,293.92"
L PICTURE X(12)         VALUE " - 5,555.55 "
M PICTURE X(9)          VALUE "5,555.55-"
N PICTURE X(13)         VALUE " 77,777.77 + "
O PICTURE X(3)          VALUE "$33"
P PICTURE X(5)          VALUE "$0.11"
Q PICTURE X(9)          VALUE "$4,000.00"
R PICTURE X(14)         VALUE "$1,000,000.50"
S PICTURE X(14)         VALUE " $  3,900.21 "
T PICTURE X(14)         VALUE " + $ 9,000.99"
U PICTURE X(15)         VALUE " $  3,890.20 + "
TEMP PICTURE S9(5)V9(5)
```

e. Statements structure

At least one of the following COBOL statements must be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
```

Identifier-2 must never be used as a function invocation.
Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function.
Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Relative error = 0 or .00002

Specific features to be tested	Arguments (arg1 ...)	Expected Answer
no argument 2 for tests a) - o)		
a) arg1 = constant string "N"	("9")	9
b) arg1 = variable string "NN...N"	(B)	203
c) arg1 = constant string "NN...N,NN...N"	("92,483")	92483
d) arg1 = variable string	(C)	>= 0.428991
".NN...N"		<= 0.429009
e) arg1 = constant string	("385.93")	>= 385.922
"NN...N.NN...N"		<= 385.938
f) arg1 = variable string	(J)	>= 8848.76
"NN...N,NN...N.NN...N"		<= 8849.11
g) arg1 = constant string	("+394.2 ")	>= 394.192
"SNN...N.NN...N "		<= 394.208
h) arg1 = constant string	(" 939.83")	>= 939.811
" NN...N.NN...N"		<= 939.849
i) arg1 = constant string	(" - 4929.0323")	>=-4929.1309
" S NN...N.NN...N "		<=-4928.9337
j) arg1 = variable string	(K)	>= 4825197.41
SNN...N,NN...N,NN...N.NN...N"		<= 4825390.43
k) arg1 = variable string	(L)	>=-5555.66
" S NN...N,NN...N.NN...N "		<=-5555.44

l) argl = constant string "NN...N.NN...NS"	("82.9312+")	>= 82.9295 <= 82.9329
m) argl = variable string "NN...N,NN...N.NN...NS"	(M)	>=-5555.66 <=-5555.44
n) argl = constant string " NN...N.NN...N S "	(" 200.0002 - ")	>=-200.0042 <=-199.9962
o) argl = variable string " NN...N,NN...N.NN...N S "	(N)	>= 77776.21 <= 77779.33
argument 2 = currency sign for test p) - cc)		
p) argl = constant string "2N"	("$5", "$")	5
q) argl = variable string "2NN...N"	(O, "$")	33
r) argl = constant string "2NN...N,NN...N"	("$93,021", "$")	93021
s) argl = constant string "2NN...N.NN...N"	("$924.93", "$")	>=924.912 <=924.948
t) argl = variable string "2NN...N,NN...N.NN...N"	(Q, "$")	4000
u) argl = constant string "S2NN...N.NN...N"	("-$34.03", "$")	>=-34.0307 <=-34.0293
v) argl = variable string "S2NN...N,NN...N,NN...N.NN...N"	(R, "$")	>= 999980.5 <=1000020.5
w) argl = constant string " 2 NN...N.NN...N"	(" $ 89.01", "$")	>= 89.0082 <= 89.0118
x) argl = variable string " 2 NN...N,NN...N.NN...N "	(S, "$")	>= 3900.13 <= 3900.29
y) argl = constant string "S 2 NN...N.NN...N"	("- $ 890.21", "$")	>=-890.228 <=-890.192
z) argl = variable string "S 2 NN...N,NN...N.NN...N"	(T, "$")	>= 9000.81 <= 9001.17
aa) argl = constant string " 2 NN...N.NN...N S "	(" $ 90.54 - ", "$")	>=-90.5418 <=-90.5382
bb) argl = variable string " 2 NN...N,NN...N.NN...N S "	(U, "$")	>= 3890.12 <= 3890.28
cc) NUMVAL-C used as part of an expression	TEMP=NUMVAL-C("90")+10	TEMP = 100
dd) NUMVAL-C used twice in an expression	TEMP=NUMVAL-C("2")+ NUMVAL-C("8")	TEMP = 10

5.27 IF127A

a. Features Tested

This program tests the Intrinsic Function ORD, which returns an integer value that is the ordinal position of argument-1 in the collating sequence for the program. The lowest ordinal position is 1. The type of this function is integer. Argument-1 must be one character in length and must be class alphabetic or alphanumeric. The returned value is the ordinal position of argument-1 in the collating sequence for the program.

FUNCTION ORD (arg1)

b. Reference

Page A-60
Section 2.31

c. Number of tests

9

d. Variables

```
A PICTURE X             VALUE "F"
B PICTURE X             VALUE "d"
C PICTURE X             VALUE "3"
ARG1 PICTURE X          VALUE "A"
TEMP PICTURE S9(10)
```

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL FUNCTION ORD(arg1) = 67

      procedure-name-1
            ...
            arg1 = "B"
```

Identifier-2 must never be used as a function invocation. Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function. Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Specific features to be tested	Arguments (argl ...)	Expected Answer
a) Z = constant upper case alphabetic character	("A")	66
b) Z = constant lower case alphabetic character	("m")	110
c) Z = constant numeral character	("5")	54
d) Z = variable upper case alphabetic character	(A)	71
e) Z = variable lower case alphabetic character	(B)	101
f) Z = variable numeral character	(C)	52
g) ORD used in an expression	TEMP=ORD("g")+1	TEMP = 105
h) ORD used twice in an expression	TEMP=ORD("A")+ORD(A)	TEMP = 137

5.28 IF128A

a. Features Tested

This program tests the Intrinsic Function <u>ORD-MAX</u>, which returns a value that is the ordinal number of the argument-1 that contains the maximum value. The type of this function is integer. If more than one argument-1 is specified, all arguments must be of the same class. The returned value is the ordinal number that corresponds to the position of the argument-1 having the greatest value in the argument-1 series. If more than one argument-1 has the same greatest value, the number returned corresponds to the position of the leftmost argument-1 having that value.

FUNCTION ORD-MAX (argl ...)

b. Reference

Page A-61
Section 2.32

c. Number of tests

16

d. Variables

```
A PICTURE S9(10)        VALUE  5
B PICTURE S9(10)        VALUE  7
C PICTURE S9(10)        VALUE  4
D PICTURE S9(10)        VALUE 10
I PICTURE X(4)          VALUE "R"
J PICTURE X(4)          VALUE "U"
P PICTURE S9(10)        VALUE  1
Q PICTURE S9(10)        VALUE  3
R PICTURE S9(10)        VALUE  5
ARG1 PICTURE S9(10)     VALUE  1
ARR                     VALUE  "40537"
      IND OCCURS 5 TIMES PICTURE 9
TEMP PICTURE S9(10)
```

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL FUNCTION ORD-MAX(5,arg1) =
   2

      procedure-name-1
            ...
            arg1 = arg1 + 1
```

Identifier-2 must never be used as a function invocation. Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function. Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Specific features to be tested	Arguments (argl ...)	Expected Answer
a) Z = multiple integer constants	(5, 3, 2, 8, 3, 1)	4
b) Z = multiple integer consts	(3, 2, 7, 1, 5)	3
c) Z = multiple integer variables	(A, B, D)	3
d) Z = multiple integer variables	(A, B, C)	2
e) Z = multiple variables and and constants	(A, 4, B, 7, C, 9)	6
f) Z = multiple variables and constants	(4, 9, A, 3)	2
g) Z = multiple alphanumeric characters	("A", I, "P")	2
h) Z = multiple alphabetic characters	("S", "D", J)	3
i) Z = multiple vars and consts (all the same value)	(A, 5, 5, A)	1
j) Z = a series of subscripted variables	(IND(1), IND(2), IND(3))	3
k) Z = a series of subscripted constants	(IND(R), IND(P), IND(Q))	1
l) Z = ALL used as a subscript to reference a table	(IND(ALL))	5
m) ORD-MAX function that invokes itself	TEMP=ORD-MAX(ORD-MAX (1,4),3, 1)	2
n) ORD-MAX function used as part of an expression	TEMP=ORD-MAX(2,3,C)+A	TEMP = 8
o) ORD-MAX function used twice within an expression	TEMP=ORD-MAX(2,3,A)+ ORD-MAX(1,1)	TEMP = 4

5.29 IF129A

a. Features Tested

This program tests the Intrinsic Function ORD-MIN, which returns a value that is the ordinal number of the argument that contains the minimum value. The type of this function is integer. If more than one argument-1 is specified, all arguments must be of the same class. The returned value is the ordinal number that corresponds to the position of the argument-1 having the least value in the argument-1 series. If more than one argument-1 has the same least value, the number returned corresponds to the position of the leftmost argument-1 having that value.

FUNCTION ORD-MIN (arg1 ...)

b. Reference

Page A-62
Section 2.33

c. Number of tests

17

d. Variables

```
A PICTURE S9(10)         VALUE  5
B PICTURE S9(10)         VALUE  7
C PICTURE S9(10)         VALUE  4
D PICTURE S9(10)         VALUE 10
I PICTURE X(4)           VALUE "R"
J PICTURE X(4)           VALUE "U"
P PICTURE S9(10)         VALUE  1
Q PICTURE S9(10)         VALUE  3
R PICTURE S9(10)         VALUE  5
ARG1 PICTURE S9(10)      VALUE  10
ARR                      VALUE  "40537"
      IND OCCURS 5 TIMES PICTURE 9
TEMP PICTURE S9(10)
```

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL FUNCTION ORD-MIN(2,arg1) >
   1

      procedure-name-1
            ...
            arg1 = arg1 - 1
```

Identifier-2 must never be used as a function invocation.
Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function.
Condition-1 refers to a conditional expression containing

arithmetic expressions for which one of its operands is an Intrinsic Function.

Specific features to be tested	Arguments (argl ...)	Expected Answer
a) Z = multiple integer constants	(5, 3, 2, 8, 3, 1)	6
b) Z = multiple integer consts	(3, 2, 7, 1, 5)	4
c) Z = multiple integer consts	(5, 4, 3, 6, 2, 8)	5
d) Z = multiple integer variables	(A, B, C)	3
e) Z = multiple integer variables	(A, B, D)	1
f) Z = multiple variables and and constants	(A, 4, B, 7, 1, 9)	5
g) Z = multiple variables and constants	(4, 1, A, 3)	2
h) Z = multiple alphanumeric characters	("A", I, "P")	1
i) Z = multiple alphabetic characters	("S", "D", J)	2
j) Z = multiple vars and consts (all the same value)	(A, 5, 5, A)	1
k) Z = a series of constant subscripted variables	(IND(1), IND(2), IND(3))	2
l) Z = a series of variable subscripted variables	(IND(P), IND(Q), IND(R))	1
m) ORD-MIN function that invokes itself	TEMP=ORD-MIN(ORD-MIN (1,4),3, 7))	1
n) Z = ALL used as a subscript to reference a table	(IND(ALL))	2
o) ORD-MIN function used as part of an expression	TEMP=ORD-MIN(2,3,C)+A	TEMP = 6
p) ORD-MIN function used twice within an expression	TEMP=ORD-MIN(9,3,A)+ ORD-MIN(1,1)	TEMP = 3

5.30 IF130A

a. Features Tested

This program tests the Intrinsic Function PRESENT-VALUE, which returns a value that approximates the present value of a series of future period-end amounts specified by argument-2 at a discount rate specified by argument-1. The type of this function is numeric. The returned value is an approximation of the summation of a series of calculations with each term in the following form:

arg2 / (1 + argl) ** n

There is one term for each occurrence of argument-2. The exponent, n, is incremented from one by one for each term in the series.

Arg1 = discount rate and must be greater than -1.
Arg2 = the series of integer and non-integer end-period amounts.
n = end-period position in arg2 series.

FUNCTION PRESENT-VALUE (arg1 arg2 ...)

b. Reference

Page A-63
Section 2.34

c. Number of tests

21

d. Variables

```
A PICTURE S9(10)        VALUE  5
B PICTURE S9(10)        VALUE  7
C PICTURE S9(10)        VALUE -4
D PICTURE S9(10)        VALUE 10
E PICTURE S9(5)V9(5)    VALUE 34.26
F PICTURE S9(5)V9(5)    VALUE -8.32
G PICTURE S9(5)V9(5)    VALUE  4.08
H PICTURE S9(5)V9(5)    VALUE  5.3
I PICTURE S9(5)V9(5)    VALUE  0.0009
J PICTURE S9(5)V9(5)    VALUE  0.0008
K PICTURE S9(10)        VALUE  23000
L PICTURE S9(10)        VALUE -23000
P PICTURE S9(10)        VALUE  1
Q PICTURE S9(10)        VALUE  3
R PICTURE S9(10)        VALUE  5
ARG1 PICTURE S9(10)     VALUE  0
ARR                     VALUE  "40537"
      IND OCCURS 5 TIMES PICTURE 9
TEMP PICTURE S9(10)V9(5)
```

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2

```
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL FUNCTION
   PRESENT-VALUE(arg1,2) < .5

      procedure-name-1
            ...
            arg1 = arg1 + 1
```

Identifier-2 must never be used as a function invocation. Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function. Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Simple Tests (relative error = .00002)

Specific features to be tested	Arguments (arg1 arg2 ...)	Expected Answer
a) arg1 = 0 for tests b) - t) arg1 = any integer or non-integer > -1	(0, 23, 12, 9)	>=43.9991 <=44.0009
b) arg2 = multiple integer constants	(1, 10, 20, 10, 5)	>= 11.5623 <= 11.5627
c) arg2 = multiple non-integer constants	(.5, 8.3, 2.4, 9.9)	>= 9.53314 <= 9.53352
d) arg2 = multiple integer and non-integer constants	(.1, 5, 4, 2.8, 3.1, 17)	>= 22.6274 <= 22.6283
e) arg2 = multiple integer variables	(.04, A, B, D)	>= 20.1691 <= 20.1699
f) arg2 = multiple non-integer variables	(.08, E, G, H, F)	>= 33.3113 <= 33.3127
g) arg2 = multiple integer constants and variables	(.2, C, A, 5, 4, 2)	>= 5.76505 <= 5.76528
h) arg2 = multiple integer and non-integer constants and variables	(.3, A, H, .07, -19)	>= 0.361674 <= 0.361689
i) arg2 = multiple low magnitude constants	(.09, -.0009, -.0008)	>=-0.001500 <=-0.001498
j) arg2 = multiple large magnitude constants	(.4, 30000, 40000, 100000, -80000)	>= 57454.07 <= 57456.37
k) arg2 = multiple large magnitude variables	(.07, L, K)	>=-1406.26 <=-1406.21
l) arg2 = multiple subscripted constants	(.15, IND(1), IND(2), IND(3))	>= 6.76570 <= 6.76597

m) arg2 = multiple subscripted variables	(.13, IND(P), IND(Q), IND(R))	>= 12.3066 <= 12.3071
n) arg2 = multiple integer constants of all same value	(.1, 10, 10, 10, 10, 10)	>= 37.9070 <= 37.9085

Complex Tests (relative error = .00004)

Specific features to be tested	Arguments (arg1 arg2 ...)	Expected Answer
a) arg2 = multiple integer expressions	(-.5, 2+3, 6/3, 9-3)	>= 65.9974 <= 66.0026
b) arg2 = multiple non-integer expressions	(-.2, 5/4, 3.3*4, 9.4+2)	>= 44.4513 <= 44.4549
c) arg2 = multiple variables, constants and expressions	(.5, A+2, 4.5/C, 8, B)	>= 7.91943 <= 7.92007
d) use PRESENT-VALUE in an expression	TEMP=PRESENT-VALUE (.08, 2, 3) + 18	TEMP: >= 22.4229 <= 22.4247
e) use PRESENT-VALUE in twice in an expression	TEMP=PRESENT-VALUE (.03, -6, -4)+PRESENT-VALUE(0.2, 9)	TEMP: >=-2.09570 <=-2.09554
f) use PRESENT-VALUE in an expression that invokes itself	TEMP=PRESENT-VALUE (PRESENT-VALUE(1, 2), 3)	TEMP: >= 1.49994 <= 1.50006

5.31 IF131A

a. Features Tested

This program tests the Intrinsic Function RANDOM, which returns a random number based on its argument (if any). The function is applied to non-negative numbers. If an argument is given then it is used as the seed value. All returned values should be in the range >= 0 and < 1. For a given seed value on a given implementation, the sequence of pseudo-random numbers will always be the same. The implementor will specify the subset of the domain of argument-1 values that will yield distinct sequence of pseudo-random numbers. The subset must include the values from 0 through at least 32767.

FUNCTION RANDOM [(arg1)]

b. Reference

Page A-64
Section 2.35

c. Number of tests

12

d. Variables

```
A PICTURE S9(10)        VALUE  4
P PICTURE S9(10)        VALUE  1
Q PICTURE S9(10)        VALUE  3
R PICTURE S9(10)        VALUE  5
ARR                     VALUE  "40537"
      IND OCCURS 5 TIMES PICTURE 9
TEMP PICTURE S9(8)V9(8)
```

e. Statements structure

At least one of the following COBOL statements must be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
```

Identifier-2 must never be used as a function invocation.
Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function.
Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Specific features to be tested	Arguments (arg1)	Expected Answer
a) no argument		=>0, <1
b) Z = integer constant	(3)	=>0, <1
c) Z = integer variable	(Q)	=>0, <1
d) Z = subscripted int constant	(IND(4))	=>0, <1
e) Z = subscripted int variable	(IND(A))	=>0, <1
f) RANDOM used in an expression	TEMP=RANDOM(2)+1	1<=TEMP<2

g) RANDOM used twice in an expression	TEMP=RANDOM(1)+ RANDOM(2)	0<=TEMP<2
h) RANDOM used recursively in an expression	TEMP= RANDOM(INTEGER (100 * RANDOM(1)))	0<=TEMP<1
i) use 0 as the seed	(0)	=>0, <=1
j) use 32767 as the seed	(32767)	=>0, <=1

k) check that when same seed is used, the same result is obtained	TEMP1 = RANDOM(1)
	TEMP2 = RANDOM(1)
		TEMP1=TEMP2 (both values (should be equal)

l) Check that returned values conform to a rectangular distribution	Specifics are yet to be discussed with NCC	

5.32 IF132A

a. Features Tested

This program tests the Intrinsic Function RANGE, which returns a value that is equal to the value of the maximum argument minus the value of the minimum argument. The returned value is equal to the greatest value of argument-1 minus the least value of argument-1. The type of this function depends upon the argument types as follows:

Argument Type	Function Type
All arguments integer	Integer
Numeric (some args. may be integer)	Numeric

Argument-1 must be class numeric.

FUNCTION RANGE (argl ...)

b. Reference

Page A-65
Section 2.36

c. Number of tests

16

d. Variables

A PICTURE S9(10)	VALUE 5
B PICTURE S9(10)	VALUE 7
C PICTURE S9(10)	VALUE -4
D PICTURE S9(10)	VALUE 10
E PICTURE S9(5)V9(5)	VALUE 34.26

```
F PICTURE S9(5)V9(5)    VALUE -8.32
G PICTURE S9(5)V9(5)    VALUE  4.08
H PICTURE S9(5)V9(5)    VALUE -5.3
M PICTURE S9(10)        VALUE 320000
N PICTURE S9(10)        VALUE 650000
O PICTURE S9(10)        VALUE -430000
P PICTURE S9(10)        VALUE  1
Q PICTURE S9(10)        VALUE  3
R PICTURE S9(10)        VALUE  5
ARG1 PICTURE S9(10)     VALUE  2
ARR                     VALUE  "40537"
      IND OCCURS 5 TIMES PICTURE 9
TEMP PICTURE S9(10)
```

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, must be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL FUNCTION RANGE(arg1,1) >
   10

      procedure-name-1
            ...
            arg1 = arg1 + 1
```

Identifier-2 must never be used as a function invocation.
Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function.
Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Simple Tests (relative error = 0 or .00002)

Specific features to be tested	Arguments (arg1 ...)	Expected Answer
a) Z = series of integer constants	(5, -2, -14, 0)	19
b) Z = series of non-integer constants	(3.9, -0.3, 8.7, 100.2)	>=100.498 <=100.502

c) Z = series of integer variables	(A, B, C, D)	14
d) Z = series of non-integer variables	(E, F, G)	>= 42.5791 <= 42.5809
e) Z = series of integer and non-integer constants	(10.2 -0.2, 5.6, -15.6)	>= 25.7992 <= 25.8005
f) Z = series of integer and non-integer variables	(A, B, C, D, E, F, G)	>= 42.5791 <= 42.5809
g) Z = a series of subscripted constants	(IND(1), IND(2), IND(3))	5
h) Z = a series of subscripted variables	IND(P), IND(Q), IND(R))	3
i) Z = ALL used as a subscript to reference a table	(IND(ALL))	7
j) Z = a series of medium to low magnitude constants	(0.032, 0.019, 0.014, -0.065)	>= 0.096998 <= 0.097002
k) Z = a series of large magnitude variables	(M, N, O)	1080000
l) Z = series of same values	(A, 5, A)	0

Complex Tests (relative error = .00002)

Specific features to be tested	Arguments (arg1 ...)	Expected Answer
a) RANGE used as part of an expression	TEMP=RANGE(A,B)+78	TEMP: >= 79.9984 <= 80.0160
b) RANGE used twice within an expression	TEMP=RANGE(A,B)+ RANGE(-2.6,-4.4,1)	TEMP: >= 7.39985 <= 7.40015
c) RANGE function that invokes itself	TEMP=RANGE(RANGE(6.8, -6.8),4)	TEMP: >= 9.59981 <= 9.60019

5.33 IF133A

a. Features Tested

This program tests the Intrinsic Function REM which returns a numeric value that is the remainder of argument-1 divided by argument-2. The type of this function is numeric. The returned value is specified by the expression:

REM(arg1,arg2) = arg1 - (arg2 * FUNCTION INTEGER-PART (arg1 / arg2))

argument-1 and argument-2 must be class numeric.
argument-2 must not be zero.

FUNCTION REM (argl arg2)

b. Reference

Page A-66
Section 2.37

c. Number of tests

17

d. Variables

```
A PICTURE S9(10)          VALUE  5
B PICTURE S9(5)V9(5)      VALUE  7.36
C PICTURE S9(10)          VALUE -4
D PICTURE S9(10)          VALUE  7
ARG2 PICTURE S9(10)       VALUE  1
TEMP PICTURE S9(10)
```

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, must be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL FUNCTION REM(5,arg2) >= 2

      procedure-name-1
            ...
            arg2 = arg2 + 1
```

Identifier-2 must never be used as a function invocation.
Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function.
Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Simple Tests (relative error = 0 or .00002)

Specific features to be tested	Arguments (arg1 arg2)	Expected Answer
a) arg1 = 0	(0, 20)	0
b) arg1=arg2 (both constants)	(10.674, 10.674)	>=-0.000020 <= 0.000020
c) arg1 = const, arg2 = var	(2.5, A)	>= 2.49995 <= 2.50005
d) arg1 = var, arg2 = const	(A, 2)	1
e) arg1 = var, arg2 = var	(B, A)	>= 2.35995 <= 2.36005
f) arg1 < 0, arg2 < 0	(-11, -5)	-1
g) arg1 > 0, arg2 < 0	(11, -5)	1
h) arg1 < 0, arg2 > 0	(-11, 5)	-1

Complex Tests (relative error = .00002)

Specific features to be tested	Arguments (arg1 arg2)	Expected Answer
a) arg1 = const, arg2 = expr	(0.89, B + 1)	>= 0.889982 <= 0.890018
b) arg1 = var, arg2 = expr	(B, C + 2.2)	>= 0.159997 <= 0.160003
c) arg1 = expr, arg2 = const	(3 / 2, .75)	>=-0.000020 <= 0.000020
d) arg1 = expr, arg2 = var	(8 + 6, B)	>= 6.63987 <= 6.64013
e) arg1 = expr, arg2 = expr	(C + 1, 2)	>=-1.00002 <=-0.999980
f) arg1 = REM Fxn, arg2 = var	(REM (D, A), A)	>= 1.99996 <= 2.00004
g) arg1 = var, arg2 = REM Fxn	(C,REM (C, D))	>=-0.000020 <= 0.000020
h) arg1 = REM Fxn, arg2 REM Fxn	(REM(9, 5),REM(D, 4))	>= 0.999980 <= 1.00002

5.34 IF134A

a. Features Tested

This program tests the Intrinsic Function <u>REVERSE</u>, which returns a character string of exactly the same length as argument-1 and whose characters are exactly the same as those of argument-1, except that they are in reverse order. The type of this function is alphanumeric. Argument-1 must be class alphabetic or alphanumeric and must be at least one character in length. If argument-1 is a character string of length n, the returned value is a character string of length n, such that for 1<=J<=n, the character in position j of the returned value is the character from position n-j+1 of argument-1.

FUNCTION REVERSE (arg1)

b. Reference

Page A-67
Section 2.38

c. Number of tests

13

d. Variables

```
A PICTURE A(10)          VALUE "tumble"
B PICTURE A(10)          VALUE "WEED"
C PICTURE X(10)          VALUE "Was"
D PICTURE X(10)          VALUE "4"
E PICTURE X(10)          VALUE "And4"
TEMP1 PICTURE X(7)       VALUE "giZZard"
TEMP PICTURE S9(10)
```

e. Statements structure

At least one of the following COBOL statements must be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the MOVE statement where possible.

```
1) MOVE identifier-1 TO identifier-2
2) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
```

Identifier-1 refers to a function invocation.
Identifier-2 must never be used as function invocation.
Condition-1 refers to a conditional expression for which one of its operands is an Intrinsic Function.

Specific features to be tested	Arguments (arg1)	Expected Answer
a) Z = constant string of only lower case alphabetic characters	("figure")	"erugif"
b) Z = constant string of only upper case alphabetic characters	("CAPS")	"SPAC"
c) Z = constant string of mixed case alphabetic chars	("highnLOW")	"WOLnhgih"
d) Z = constant string of non-alphabetic characters	("95")	"59"
e) Z = constant string of alphabetic and non-alphabetic characters	("8isaNUMBER")	"REBMUNasi8"
f) Z = variable string of all lower case alphabetic characters	(A)	" elbmut"
g) Z = variable string of all upper case alphabetic characters	(B)	" DEEW"
h) Z= variable string of mixed case alphabetic chars	(C)	" saW"
i) Z = variable string of non-alphabetic characters	(D)	" 4"
j) Z = variable string of alphabetic and non-alphabetic characters	(E)	" 4dnA"
k) REVERSE used as part of an expression	TEMP=LENGTH(REVERSE("Homer"))	TEMP = 5
l) REVERSE used to invoke itself	TEMP1=REVERSE(REVERSE ("giZZard"))	TEMP1= "giZZard"
m) REVERSE used twice in an expression	TEMP=LENGTH(REVERSE("HOMER"))+LENGTH(REVERSE ("Gizzard"))	TEMP = 12

5.35 IF135A

a. Features Tested

This program tests the Intrinsic Function SIN, which returns a numeric value that approximates the sine of angle or arc, expressed in radians, that is specified by argument-1. The type of this function is numeric. Argument-1 must be class numeric. The returned value is the approximation of the sine of argument-1 and is greater than or equal to -1 and less than or equal to 1.

FUNCTION SIN (argl)

b. Reference

Page A-68
Section 2.39

c. Number of tests

32

d. Variables

```
A PICTURE S9(5)V9(5)    VALUE  -0.00004
B PICTURE S9(5)V9(5)    VALUE  14000.105
C PICTURE S9(10)        VALUE  100000
D PICTURE S9(10)        VALUE  1000
PI PICTURE S9V9(17)     VALUE  3.141592654
P PICTURE S9(10)        VALUE  1
ARG1 PICTURE S9(10)     VALUE  3
ARR                     VALUE  "40537"
      IND OCCURS 5 TIMES PICTURE 9
TEMP PICTURE S9(5)V9(5)
```

e. Statements structure

At least one the following COBOL statements, excluding PERFORM, be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL FUNCTION SIN(argl) < 0

      procedure-name-1
            ...
            argl = argl - 1
```

Identifier-2 must never be used as a function invocation.
Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function.
Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Simple Tests (relative error = .00002)

Specific features to be tested	Arguments (argl)	Expected Answer
a) Z = 0	(0)	>=-0.000020 <= 0.000020
b) Z = PI	(PI)	>=-0.000020 <= 0.000020
c) Z = -PI	(-PI)	>=-0.000020 <= 0.000020
d) Z = Values close to 0	(0.001)	>= 0.000998 <= 0.001000
e) Z = a low magnitude non-integer constant (const < .0001)	(.00009)	>= 0.000089 <= 0.000090
f) Z = a low magnitude non-integer variable (var < .0001)	(A)	>=-0.000040 <=-0.000039
g) Z = a variable subscripted variable	(IND(P))	>=-0.756817 <=-0.756787
h) Z = a constant subscripted variable	(IND(4))	>= 0.141117 <= 0.141123

Complex Tests (relative error = .00004)

Specific features to be tested	Arguments (argl)	Expected Answer
a) Z = PI/3	(PI / 3)	>= 0.865990 <= 0.866060
b) Z = PI/2	(PI / 2)	>= 0.999960 <= 1.00000
c) Z = 3PI/2	((3 * PI)/ 2)	>=-1.00000 <=-0.999960
d) Z = -PI/3	(-PI / 3)	>=-0.866060 <=-0.865990
e) Z = -PI/2	(-PI / 2)	>=-1.00000 <=-0.999960
f) Z = -3PI/2	((3 * -PI)/ 2)	>= 0.999960 <= 1.00000
g) Z = Values close to PI/2	((PI/2) - 0.001)	>= 0.999960 <= 1.00000
h) Z = Values close to PI/3	((PI/3) + 0.001)	>= 0.866489 <= 0.866559
i) Z = Values close to PI	(PI - 0.001)	>= 0.000874 <= 0.001126
j) Z = Values close to 3PI/2	(((3 * PI) / 2) + 0.001)	>=-1.00000 <=-0.999960

k) Z = Expr. with value close to or equal to 0	(PI * (4 - 2) / 180)	>= 0.034898 <= 0.034900
l) Z = Expr. with value close to or equal to PI/2	((PI / 2) - (PI / 180))	>= 0.999807 <= 0.999887
m) Z = Expr. with value close to or equal to PI/3	((PI / 3) - (PI / 180))	>= 0.857132 <= 0.857201
n) Z = Expr. with value close to or equal to PI	(PI + (PI / 180))	>=-0.017453 <=-0.017451
o) Z = Expr. with value close to or equal to 3PI/2	((PI * 272) / 180)	>=-0.999430 <=-0.999350
p) Z = an integer expression using constants only	(4 / 2)	>= 0.909261 <= 0.909333
q) Z = a non-integer expression using constants only	(3 / 2)	>= 0.997454 <= 0.997534
r) Z = a non-integer expression using variables only	(PI - A)	>=-0.000040 <=-0.000039
s) Z = an integer expression using variables and constants	(D / 100)	>=-0.544043 <=-0.543999
t) Z = a non-integer expression using variables and constants	(PI / 180)	>= 0.017451 <= 0.017453
u) SIN used as part of an expression	TEMP=SIN(PI)+1	TEMP: >= 0.999960 <= 1.00000
v) SIN function that invokes itself (i.e SIN(SIN(x))), where x may be a variable or an expression	TEMP=SIN(SIN(2))	TEMP: >= 0.789040 <= 0.789104
w) The SIN function applied twice in an expression	TEMP=SIN(PI/3)+ SIN(-PI/3)	TEMP: >=-0.000040 <= 0.000040

5.36 IF136A

a. Features Tested

This program tests the Intrinsic Function SQRT, which returns a numeric value that approximates the square root of argument-1. The type of this function is numeric. Argument-1 must be class numeric and must be zero or positive. The returned value is the absolute value of the approximation of the square root of argument-1.

FUNCTION SQRT (arg1)

b. Reference

Page A-69
Section 2.40

c. Number of tests

26

d. Variables

```
A PICTURE S9(5)V9(5)     VALUE  0.00004
B PICTURE S9(5)V9(5)     VALUE  14000.105
C PICTURE S9(10)         VALUE  100000
D PICTURE S9(10)         VALUE  1000
E PICTURE S9(10)         VALUE  7
F PICTURE S9(10)         VALUE  6
P PICTURE S9(10)         VALUE  1
ARG1 PICTURE S9(10)      VALUE  10
ARR                      VALUE  "40537"
      IND OCCURS 5 TIMES PICTURE 9
TEMP PICTURE S9(5)V9(5)
```

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL FUNCTION SQRT(arg1) < 2.0

      procedure-name-1
            ...
            arg1 = arg1 - 1
```

Identifier-2 must never be used as a function invocation.
Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function.
Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Simple Tests (relative error = .00002)

Specific features to be tested	Arguments (argl)	Expected Answer
a) Z = 0	(0)	>= 0.000000 <= 0.000020
b) Z = 1	(1)	>= 0.999980 <= 1.00002
c) Z = 4	(4)	>= 1.99996 <= 2.00004
d) Z = values close to 0	(.001)	>= 0.031621 <= 0.031623
e) Z = values close to 1	(.999)	>= 0.999479 <= 0.999519
f) Z = values close 4	(4.01)	>= 2.00246 <= 2.00254
g) Z = a large magnitude non-integer constant	(31409.84)	>= 177.224 <= 177.231
h) Z = a large magnitude integer constant	(860000)	>= 927.342 <= 927.379
i) Z = a low magnitude non-integer constant	(.00009)	>= 0.0094866 <= 0.0094870
j) Z = a large magnitude non-integer variable	(B)	>= 118.320 <= 118.324
k) Z = a large magnitude integer variable	(C)	>= 316.222 <= 316.234
l) Z = a low magnitude non-integer variable	(A)	>= 0.0063244 <= 0.0063246
m) Z = a variable subscripted variable	(IND(P))	>= 1.99996 <= 2.00004
n) Z = a constant subscripted variable	(IND(3))	>= 2.23601 <= 2.23610

Complex Tests (relative error = .00004)

Specific features to be tested	Arguments (argl)	Expected Answer
a) Z = expr. with value close to or equal to 0	(9 - 8.9)	>= 0.316214 <= 0.316240
b) Z = expr. with value close to or equal to 4	(8 / 2.1)	>= 1.95172 <= 1.95188
c) Z = an integer expression using constants only	(35 * 9)	>= 17.7475 <= 17.7489
d) Z = a non-integer expression using constants only	(9 / 7)	>= 1.13384 <= 1.13393
e) Z = an integer expression using variables only	(E + F)	>= 3.60541 <= 3.60569

f) Z = a non-integer expression using variables only	(D / E)	>= 11.9517 <= 11.9527
g) Z = an integer expression using constants and and variables	(F - 3)	>= 1.73198 <= 1.73212
h) Z = a non-integer expression using constants and variables	(E * 2.3)	>= 4.01232 <= 4.01264
i) Z = SQRT function that invokes itself using both variables & expressions	TEMP=SQRT(SQRT(F))	TEMP: >= 1.56502 <= 1.56514
j) The SQRT function used twice within an expression	TEMP=SQRT(6.5)+ SQRT(5.4)	TEMP: >= 4.87309 <= 4.87348
k) check that x=(sqrt(x**2))	TEMP = SQRT(10)**2	TEMP: >= 9.99960 <= 10.0004

5.37 IF137A

a. Features Tested

This program tests the Intrinsic Function STANDARD-DEVIATION, which returns a numeric value that approximates the standard deviation of its arguments. The type of this function is numeric. Argument-1 must be class numeric. The returned value is the approximation of the standard deviation of the argument-1 series. The returned value is calculated as follows:

a) The difference between each argument-1 value and the arithmetic mean of the argument-1 series is calculated and squared.
b) The values obtained are then added together. This quantity is divided by the number of values in the argument-1 series.
c) The square root of the quotient obtained is then calculated.

The returned value is the absolute value of this square root.

If the argument-1 series consists of only one value or if the argument-1 series consists of all variable occurrences data items and the total number of occurrences for all of them is one, the returned value is zero.

FUNCTION STANDARD-DEVIATION (arg1 ...)

b. Reference

Page A-70
Section 2.41

c. Number of tests

17

d. Variables

```
A PICTURE S9(10)        VALUE  5
B PICTURE S9(10)        VALUE  7
C PICTURE S9(10)        VALUE -4
D PICTURE S9(10)        VALUE 10
E PICTURE S9(5)V9(5)    VALUE 34.26
F PICTURE S9(5)V9(5)    VALUE -8.32
G PICTURE S9(5)V9(5)    VALUE  4.08
H PICTURE S9(5)V9(5)    VALUE -5.3
P PICTURE S9(10)        VALUE  4
Q PICTURE S9(10)        VALUE  3
R PICTURE S9(10)        VALUE  5
ARR                     VALUE  "40537"
ARG3 PICTURE S9(10)     VALUE  2
      IND OCCURS 5 TIMES PICTURE 9
TEMP PICTURE S9(10)
```

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL FUNCTION
   STANDARD-DEVIATION (1,1,arg3) > 1

      procedure-name-1
            ...
            arg3 = arg3 + 1
```

Identifier-2 must never be used as a function invocation. Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function. Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Simple Tests (relative error = .00002)

Specific features to be tested	Arguments (arg1 ...)	Expected Answer
a) Z = series of integer constants	(5, -2, -14, 0)	>= 6.97750 <= 6.97778
b) Z = series of non-integer constants	(3.9, -0.3, 8.7, 100.2)	>= 41.7333 <= 41.7350
c) Z = series of integer variables	(A, B, C, D)	>= 5.22005 <= 5.22025
d) Z = series of non integer variables	(E, F, G, H)	>= 16.8440 <= 16.8447
e) Z = series of integer and non-integer constants	(10.2, -0.2, 5.6, -15.6)	>= 9.73119 <= 9.73158
f) Z = series of integer and non-integer variables	(A, B, C, D, E, F, G, H)	>= 12.4976 <= 12.4981
g) Z = a series of constant subscripted variables	(IND(1), IND(2), IND(3))	>= 2.16020 <= 2.16028
h) Z = a series of variable subscripted variables	(IND(P), IND(Q), IND(R))	>= 1.63296 <= 1.63302
i) Z = ALL used as subscript to reference a table	(IND(ALL))	>= 2.31511 <= 2.31521
j) Z = a series of low magnitude constants	(0.00032, 0.00019, 0.00014, -0.06574)	>= 0.028559 <= 0.028561
k) Z = series of same values	(A, 5, A)	>=-0.000020 <= 0.000020

Complex Tests (relative error = .00004)

Specific features to be tested	Arguments (arg1 ...)	Expected Answer
a) Z = mixture of expressions and constants	(2.6 + 30, 4.5 * 2)	>= 11.7995 <= 11.8005
b) Z = mixture of expressions variables and consts	(E, 9 * A, 0, B / 2)	>= 19.3556 <= 19.3572
c) STANDARD-DEVIATION used as part of an expression	TEMP=STANDARD-DEVIATION (A)+78	TEMP: >= 77.9969 <= 78.0031

d) STANDARD-DEVIATION used twice within an expression	TEMP=STANDARD-DEVIATION (A,B)+STANDARD-DEVIATION (1,1)	TEMP: >= 0.99996 <= 1.00004
e) STANDARD-DEVIATION used recursively	TEMP=STANDARD-DEVIATION (STANDARD-DEVIATION(0, 0))	TEMP: >=-0.000040 <= 0.000040

5.38 IF138A

a. Features Tested

This program tests the Intrinsic Function SUM, which returns a value that is the sum of the arguments. The type of this function depends upon the argument types as follows:

Argument Type	Function Type
All arguments integer	Integer
Numeric (some args. may be integer)	Numeric

Argument-1 must be class numeric.
The returned value is the sum of the arguments.

FUNCTION SUM (arg1 ...)

b. Reference

Page A-71
Section 2.42

c. Number of tests

17

d. Variables

A PICTURE S9(10)	VALUE 5
B PICTURE S9(10)	VALUE 7
C PICTURE S9(10)	VALUE -4
D PICTURE S9(10)	VALUE 10
E PICTURE S9(5)V9(5)	VALUE 34.26
F PICTURE S9(5)V9(5)	VALUE -8.32
G PICTURE S9(5)V9(5)	VALUE 4.08
H PICTURE S9(5)V9(5)	VALUE -5.3
M PICTURE S9(10)	VALUE 320000
N PICTURE S9(10)	VALUE 650000
O PICTURE S9(10)	VALUE -430000
P PICTURE S9(10)	VALUE 1
Q PICTURE S9(10)	VALUE 3
R PICTURE S9(10)	VALUE 5

```
ARG1 PICTURE S9(10)      VALUE  1
ARR                      VALUE  "40537"
      IND OCCURS 5 TIMES PICTURE 9
TEMP PICTURE S9(10)
```

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL FUNCTION SUM(arg1,1) > 10

      procedure-name-1
            ...
            arg1 = arg1 + 1
```

Identifier-2 must never be used as a function invocation.
Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function.
Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Simple Tests (relative error = 0 or .00002)

Specific features to be tested	Arguments (arg1 ...)	Expected Answer
a) Z = series of integer constants	(5, -2, -14, 0)	-11
b) Z = series of non-integer constants	(3.9, -0.3, 8.7, 100.2)	>= 112.498 <= 112.502
c) Z = series of integer variables	(A, B, C, D)	18
d) Z = series of non integer variables	(E, F, G, H)	>= 24.7195 <= 24.7205
e) Z = series of integer and non-integer constants	(10.2, -0.2, 5.6, -15.6)	>=-0.000020 <= 0.000020
f) Z = series of integer and non-integer variables	(A, B, C, D, E, F, G, H)	>= 42.7191 <= 42.7209
g) Z = a series of constant subscripted variables	(IND(1), IND(2), IND(3))	9

h) Z = a series of variable subscripted variables	(IND(P), IND(Q), IND(R))	16
i) Z = ALL used as subscript to reference a table	(IND(ALL))	19
j) Z = a series of medium to low magnitude constants	(0.032, 0.019, 0.014, -0.065)	>=-0.000020 <= 0.000020
k) Z = a series of large magnitude variables	(M, N, O)	540000

Complex Tests (relative error = .00002)

Specific features to be tested	Arguments (argl ...)	Expected Answer
a) Z = mixture of expressions and constants	(2.6 + 30, 4.5 * 2)	>= 41.5992 <= 41.6008
b) Z = mixture of expressions variables and consts	(E, 9 * A, B / 2)	>= 82.7583 <= 82.7616
c) SUM used as part of an expression	TEMP=SUM(A,B)+78	TEMP: >= 89.9982 <= 90.0018
d) SUM used twice within an expression	TEMP=SUM(A,B)+SUM(-2.6, -4.4)	TEMP: >= 4.99990 <= 5.00010
e) SUM function used recursively	TEMP=SUM(SUM(6.8,-6.8), 4)	TEMP: >= 3.99992 <= 4.00008

5.39 IF139A

a. Features Tested

This program tests the Intrinsic Function TAN, which returns a numeric value that approximates the tangent of an angle or arc, expressed in radians, that is specified by argument-1. The type of this function is numeric. Argument-1 must be class numeric. The returned value is the approximation of the tangent of argument-1.

FUNCTION TAN (argl)

b. Reference

Page A-72
Section 2.43

c. Number of tests

32

d. Variables

```
A PICTURE S9(5)V9(5)    VALUE  -0.00004
B PICTURE S9(5)V9(5)    VALUE  14000.105
C PICTURE S9(10)        VALUE  100000
D PICTURE S9(10)        VALUE  1000
PI PICTURE S9V9(17)     VALUE  3.141592654
P PICTURE S9(10)        VALUE  1
ARG1 PICTURE S9(10)     VALUE  1
ARR                     VALUE  "40537"
      IND OCCURS 5 TIMES PICTURE 9
TEMP PICTURE S9(5)V9(5)
```

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL FUNCTION TAN(arg1) < 0

      procedure-name-1
             ...
            arg1 = arg1 - .25
```

Identifier-2 must never be used as a function invocation.
Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function.
Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Simple Tests (relative error = .00002)

Specific features to be tested	Arguments (arg1)	Expected Answer
a) Z = 0	(0)	>=-0.000020 <= 0.000020
b) Z = PI	(PI)	>=-0.000020 <= 0.000020

c) Z = -PI	(-PI)	>=-0.000020 <= 0.000020
d) Z = Values close to 0	(.001)	>= 0.000999 <= 0.001000
e) Z = a low magnitude non-integer constant (const < .0001)	(.00009)	>= 0.000089 <= 0.000090
f) Z = a low magnitude non-integer variable (var < .0001)	(A)	>=-0.000040 <=-0.000039
g) Z = a variable subscripted variable	(IND(P))	>= 1.15780 <= 1.15784
h) Z = a constant subscripted variable	(IND(5))	>= 0.871430 <= 0.871464

Complex Tests (relative error = .00004)

Specific features to be tested	Arguments (arg1)	Expected Answer
a) Z = PI/4	(PI / 4)	>= 0.999960 <= 1.00004
b) Z = 3PI/4	((3 * PI) / 4)	>=-1.00004 <=-0.999960
c) Z = 5PI/4	((5 * PI) / 4)	>= 0.999960 <= 1.00004
d) Z = 7PI/4	((7 * PI) / 4)	>=-1.00004 <=-0.999960
e) Z = -PI/4	(-PI / 4)	>=-1.00004 <=-0.999960
f) Z = -3PI/4	((3 * -PI)/ 4)	>= 0.999960 <= 1.00004
g) Z = -5PI/4	((5 * -PI) / 4)	>=-1.00004 <=-0.999960
h) Z = -7PI/4	((7 * -PI) / 4)	>= 0.999960 <= 1.00004
i) Z = Values close to PI/4	((PI / 4) - .001)	>= 0.997961 <= 0.998041
j) Z = Values close to PI	(PI + .001)	>= 0.000874 <= 0.001126
k) Z = Expr. with value close to or equal to 0	(1 / 180)	>= 0.0055554 <= 0.0055558
l) Z = Expr. with value close to or equal to PI/4	((PI / 4) - (PI / 180))	>= 0.965649 <= 0.965727
m) Z = Expr. with value close to or equal to PI	(PI + ((2 * PI) / 180))	>= 0.034919 <= 0.034921
n) Z = Expr. with value close to or equal to 3PI/4	(((PI * 3)/4)+(1/180))	>=-0.988990 <=-0.988910
o) Z = Expr. with value close to or equal to 5PI/4	(((PI * 5)/4) - (2/180))	>= 0.977982 <= 0.978060

p) Z = an integer expression using constants only	(4 / 2)	>=-2.18512 <=-2.18494
q) Z = a non-integer expression using constants only	(3 / 2)	>= 14.1008 <= 14.1020
r) Z = a non-integer expression using variables only	(PI - A)	>=-0.000086 <= 0.000166
s) Z = an integer expression using variables and constants	(D / 100)	>= 0.648334 <= 0.648386
t) Z = a non-integer expression using variables & constants	(PI / 180)	>= 0.017454 <= 0.017456
u) TAN used as part of an expression	TEMP=TAN(PI)+1	TEMP: >= 0.999960 <= 1.00004
v) TAN function used recursively (i.e (TAN(TAN(x))), where x may be a variable or an expression	TEMP=TAN(TAN(2))	>= 1.41786 <= 1.41798

w) The TAN function used twice in an expression	TEMP=TAN(PI/3)+
	TAN(-PI/3)
		TEMP:
		>=-0.000040
		<= 0.000040

5.40 IF140A

a. Features Tested

This program tests the Intrinsic Function UPPER-CASE, which returns a character string that is the same length as argument-1 with each lowercase letter replaced by the corresponding uppercase letter. The type of this function is alphanumeric. Argument-1 must be class alphabetic or alphanumeric and must be at least one character in length. The character string returned has the same length as argument-1.

FUNCTION UPPER-CASE (argl)

b. Reference

Page A-73
Section 2.44

c. Number of tests

13

d. Variables

```
A PICTURE A(10)          VALUE "tumble"
B PICTURE A(10)          VALUE "WEED"
C PICTURE X(10)          VALUE "Was"
D PICTURE X(10)          VALUE "4"
E PICTURE X(10)          VALUE "And4"
TEMP PICTURE S9(10)
```

e. Statements structure

At least one of the following COBOL statements must be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the MOVE statement where possible.

```
1) MOVE identifier-1 TO identifier-2
1) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
```

Identifier-1 refers to a function invocation.
Identifier-2 must never be used as function invocation.
Condition-1 refers to a conditional expression for which one of its operands is an Intrinsic Function.

Specific features to be tested	Arguments (arg1)	Expected Answer
a) Z = constant string of only lower case alphabetic characters	("figure")	FIGURE
b) Z = constant string of only upper case alphabetic characters	("CAPS")	CAPS
c) Z = constant string of mixed case alphabetic chars	("highnLOW")	HIGHNLOW
d) Z = constant string of non-alphabetic characters	("95")	95
e) Z = constant string of alphabetic and non-alphabetic characters	("8isaNUMBER")	8ISANUMBER
f) Z = variable string of all lower case alphabetic characters	(A)	TUMBLE
g) Z = variable string of all upper case alphabetic characters	(B)	WEED

h) Z = variable string of mixed case alphabetic chars	(C)	WAS
i) Z = variable string of non-alphabetic characters	(D)	4
j) Z = variable string of alphabetic and non-alphabetic characters	(E)	AND4
k) UPPER-CASE used as part of an expression	TEMP=LENGTH(UPPER-CASE ("Homer"))	TEMP = 5
l) UPPER-CASE used recursively	TEMP=UPPER-CASE(UPPER-CASE("giZZard"))	GIZZARD
m) UPPER-CASE used twice in an expression	TEMP=LENGTH(UPPER-CASE ("HOMER"))+LENGTH(UPPER-CASE("gizzard"))	TEMP = 12

5.41 IF141A

a. Features Tested

This program tests the Intrinsic Function VARIANCE, which returns a numeric value that approximates the variance of its arguments. The type of this function is numeric. Argument-1 must be class numeric. The returned value is the approximation of the variance of the argument-1 series and it is defined as the square of the STANDARD DEVIATION of the argument-1 series. If the argument-1 series consists of only one value, or if the argument-1 series consists of all variable occurrence data items and the total number of occurrences for all of them is one, the returned value is zero.

VARIANCE (arg1 ...) = FUNCTION STANDARD-DEVIATION(arg1 ...))**2

b. Reference

Page A-74
Section 2.45

c. Number of tests

17

d. Variables

A PICTURE S9(10)	VALUE 5
B PICTURE S9(10)	VALUE 7
C PICTURE S9(10)	VALUE -4
D PICTURE S9(10)	VALUE 10
E PICTURE S9(5)V9(5)	VALUE 34.26
F PICTURE S9(5)V9(5)	VALUE -8.32
G PICTURE S9(5)V9(5)	VALUE 4.08
H PICTURE S9(5)V9(5)	VALUE -5.3

```
P PICTURE S9(10)        VALUE  4
Q PICTURE S9(10)        VALUE  3
R PICTURE S9(10)        VALUE  5
ARG3 PICTURE S9(10)     VALUE  2
ARR                     VALUE  "40537"
      IND OCCURS 5 TIMES PICTURE 9
TEMP PICTURE S9(10)
```

e. Statements structure

At least one of the following COBOL statements, excluding PERFORM, be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the COMPUTE statement where possible.

```
1) COMPUTE identifier-2 = arithmetic-expression-1
2) EVALUATE expression-1 ALSO expression-2
3) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
4) PERFORM procedure-name-1 UNTIL FUNCTION
   VARIANCE(1,1,arg3) > 3

      procedure-name-1
            ...
            arg3 = arg3 + 1
```

Identifier-2 must never be used as a function invocation.
Expression-1 and expression-2 refers to expressions in which one of its operands is an Intrinsic Function.
Condition-1 refers to a conditional expression containing arithmetic expressions for which one of its operands is an Intrinsic Function.

Simple Tests (relative error = .00002)

Specific features to be tested	Arguments (argl ...)	Expected Answer
a) Z = series of integer constants	(5, -2, -14, 0)	>= 48.6865 <= 48.6885
b) Z = series of non-integer constants	(3.9, -0.3, 8.7, 100.2)	>= 1741.70 <= 1741.77
c) Z = series of integer variables	(A, B, C, D)	>= 27.2494 <= 27.2505
d) Z = series of non integer variables	(E, F, G, H)	>= 283.728 <= 283.740
e) Z = series of integer and non-integer constants	(10.2, -0.2, 5.6, -15.6)	>= 94.6981 <= 94.7019

f) Z = series of integer and non-integer variables	(A, B, C, D, E, F, G, H)	>= 156.194 <= 156.200
g) Z = a series of subscripted constants	(IND(1), IND(2), IND(3))	>= 4.66657 <= 4.66675
h) Z = a series of subscripted variables	(IND(P), IND(Q), IND(R))	>= 2.66661 <= 2.66671
i) Z = ALL used as a subscript to reference a table	(IND(ALL))	>= 5.35989 <= 5.36011
j) Z = a single value	(0.032)	>=-0.000020 <= 0.000020
k) Z = series of same values	(A, 5, A)	>=-0.000020 <= 0.000020

Complex Tests (relative error = .00004)

Specific features to be tested	Arguments (argl ...)	Expected Answer
a) VARIANCE used as part of an expression	TEMP=VARIANCE(A,B)+78	TEMP: >= 78.9968 <= 79.0031
b) Z = mixture of expressions and constants	(2.6 + 30, 4.5 * 2)	>= 139.234 <= 139.245
c) Z = mixture of expressions variables and consts	(E, 9 * A, 0, B / 2)	>= 374.658 <= 374.688
d) VARIANCE used twice within an expression	TEMP=VARIANCE(A,B)+ VARIANCE(1,1)	TEMP: >= 0.999960 <= 1.00004
e) VARIANCE used recursively	TEMP=VARIANCE(VARIANCE (0),0)	TEMP: >=-0.000040 <= 0.000040

5.42 IF142A

a. Features Tested

This program tests the Intrinsic Function WHEN-COMPILED, which returns the date and time the program was compiled as provided by the system on which the program was compiled. The type of this function is alphanumeric. For more information related to the returned value see Pages A-75 and A-76 of X3.23A-1989, "INTRINSIC FUNCTION MODULE ADDENDUM TO AMERICAN NATIONAL STANDARD COBOL, X3.23-1985."

FUNCTION WHEN-COMPILED

b. Reference

Page A-75
Section 2.46

c. Number of tests

2

d. Variables

```
TEMP1 PICTURE X(21)
TEMP2 PICTURE X(21)
```

e. Statements structure

At least one of the following COBOL statements must be used to test the features given below. Each type of statement is to be allocated the first corresponding argument(s) in the table and then all remaining arguments tested with the MOVE statement where possible.

```
1) MOVE identifier-1 TO identifier-2
2) IF condition-1 THEN
      statement-1
   ELSE
      statement-2
```

Identifier-1 refers to a function invocation.
Identifier-2 must never be used as function invocation .
Condition-1 refers to a conditional expression for which one of its operands is an Intrinsic Function.

Specific features to be tested	Arguments	Expected Answer
a) check that the range is valid	TEMP1=WHEN-COMPILED	

b) check again to make sure the same time is returned as first function call	TEMP2=WHEN-COMPILED	TEMP1=time
		TEMP2=>TEMP1

6 ACKNOWLEDGMENTS

The authors wishes to acknowledge Mr. John Cugini and Ms. Kathryn Miles from NIST for their technical suggestions and review of this paper.

Acknowledgments are also in order for Mr. Dave Bamber and Mr. Gerald Bermingham from the National Computer Centre in England for their work in translating the specifications into COBOL code.

Finally we would like to acknowledge the work of Ms. Mabel Vickers for all her technical suggestions and review of an earlier version of these specifications.

7 REFERENCES

[1] Federal Information Processing Standards Publication (FIPS PUB) 21-3, COBOL, National Institute of Standards and Technology, Computer System Laboratory, January 1990

[2] American National Standard Programming Language COBOL, ANSI X3.23-1985, ISO 1989-1985, American National Standards Institute, New York, NY, September 1985

[3] Intrinsic Function Module Addendum to American National Standard COBOL X3.23A-1989, American National Standards Institute, New York, NY, October 1988

[4] Programming Procedures Manual for the 1978 Fortran Compiler Validation System, Version 2.0, Report FCTC-81-46, Federal Compiler Testing Center, General Services Administration, March 1982

[5] Compiler Validation Procedures, National Institute of Standards and Technology, Gaithersburg, MD, February 1990

[6] Cugini, John V. "Specifications and Test Methods for Numeric Accuracy in Programming Language Standard," NBS Special Publication 500-77, National Bureau of Standards (U.S.), Gaithersburg, MD, June 1981

[7] Ambrose, William G. College Algebra and Trigonometry, Macmillan Publishing Co., Inc., New York, NY, 1977, 201-399

[8] Grady, Michael D., Beckenbach Edwin F., and Wooton William Precalculus, Wadsworth Publishing Company, Bellmont, CA, 1980, 166-255

NIST-114A
(REV. 3-90)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NUMBER
NIST/SP-500/203

2. PERFORMING ORGANIZATION REPORT NUMBER

3. PUBLICATION DATE
July 1992

4. TITLE AND SUBTITLE

Conformance Test Specifications for COBOL Intrinsic Function Module

5. AUTHOR(S)

Carmelo Montanez-Rivera and L. Arnold Johnson

6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
GAITHERSBURG, MD 20899

7. CONTRACT/GRANT NUMBER

8. TYPE OF REPORT AND PERIOD COVERED
Final

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)

Same as item #6

10. SUPPLEMENTARY NOTES

11. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.)

This document contains test specifications for the COBOL Intrinsic Functions Module of the Federal Information Processing Standard (FIPS) Programming Language COBOL, FIPS PUB 21-3. It serves as a reference manual and as a user's guide for the COBOL Intrinsic Function Module Tests in the 1985 COBOL Compiler Validation System. The tests are used by the National Institute of Standards and Technology (NIST) to test COBOL implementations for conformance to COBOL.

Tests are divided into two major categories, Simple Tests and Complex Tests. A Simple test uses a single entity as the argument, i.e., a constant or a literal by itself. Complex tests take as an argument entities such as expressions or other Intrinsic Functions. The nature of arguments for the Complex tests will most likely have an effect on the accuracy of the expected value. This effect may be compensated by allowing a greater error margin for such tests.

The testing of language processors to determine the degree to which they conform to FIPS may be required by the Government departments and agencies in accordance with the FIPS, the Federal Information Resources Management Regulation 201.13 and 201.39, and the associated Federal ADP and Telecommunications Standards Index.

12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)

ANSI; argument; COBOL; complex tests; Intrinsic Function; simple tests

13. AVAILABILITY

☒ UNLIMITED
☐ FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).
☒ ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.
☒ ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.

14. NUMBER OF PRINTED PAGES
111

15. PRICE

ELECTRONIC FORM

* U.S. G.P.O.:1992-311-891:60489

ANNOUNCEMENT OF NEW PUBLICATIONS ON COMPUTER SYSTEMS TECHNOLOGY

Superintendent of Documents
Government Printing Office
Washington, DC 20402

Dear Sir:

Please add my name to the announcement list of new publications to be issued in the series: National Institute of Standards and Technology Special Publication 500-.

Name ____

Company ____

Address ____

City ________________ State ____________ Zip Code __________

(Notification key N-503)

NIST Technical Publications

Periodical

Journal of Research of the National Institute of Standards and Technology—Reports NIST research and development in those disciplines of the physical and engineering sciences in which the Institute is active. These include physics, chemistry, engineering, mathematics, and computer sciences.

Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Institute's technical and scientific programs. Issued six times a year.

Nonperiodicals

Monographs—Major contributions to the technical literature on various subjects related to the Institute's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NIST, NIST annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NIST under the authority of the National Standard Data Act (Public Law 90-396). NOTE: The Journal of Physical and Chemical Reference Data (JPCRD) is published bimonthly for NIST by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements are available from ACS, 1155 Sixteenth St., NW., Washington, DC 20056.

Building Science Series—Disseminates technical information developed at the Institute on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NIST under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NIST administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series—Practical information, based on NIST research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

Order the **above** *NIST publications from: Superintendent of Documents, Government Printing Office, Washington, DC 20402.*

Order the **following** *NIST publications—FIPS and NISTIRs—from the National Technical Information Service, Springfield, VA 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NIST pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NIST Interagency Reports (NISTIR)—A special series of interim or final reports on work performed by NIST for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Service, Springfield, VA 22161, in paper copy or microfiche form.

U.S. Department of Commerce
National Institute of Standards and Technology
Gaithersburg, MD 20899

Official Business
Penalty for Private Use $300