RELI Manual (FORM, SORM, AFORM procedures)

Annotations for reading the text:

- ~~Inside brackets () are only annotations.~~
- An italic text designates what to type in command line.
- Bold text indicates the names of programs.

Instructions for a remote connection

1. **Windows** > start **putty** (**putty** is a small freeware program for remote connection).
2. Enter host name, user and password:
    - Host name: *titan.fsb.hr* (IP address: *161.53.116.9*)
    - User login: on request (bblag@fesb.hr)
    - Password: on request (bblag@fesb.hr)
3. Now, you are in the **UNIX** environment with all the commands available at the prompt. To list all directories and files in the present directory type: *ls* (various switches are available, *ls –l*; *ls –a*; etc., see **UNIX** or **Linux** help).
4. To change a working directory just type: *cd dirname* (to get out to previous directory, type: *cd..*).
5. To change to reliability directory: *cd rex* (rex is a directory with reliability programs).

Running programs

The full path of the directory with reliability software is: */usr1/kalman/rex*. Also, this directory contains various examples of reliability analysis (FORM, SORM, Monte Carlo simulation).

Files with extensions **.c** are UNIX shell scripts used to start various types of calculations. For example: **re.c** is a shell script to start reliability calculation (FORM) and **ms.c** is a shell script to start Monte Carlo simulation, etc.

IMPORTANT:

Input files have an extension **.s**. They are a plain text files that but have to be edited and saved in editor that is capable of saving files in UNIX format, e.g. **GWDeditor**, **Crimson Editor** (**Windows**) or vi-editor (**Linux**, **UNIX**). Initially, editors in Windows save files in MSDOS based format, so be sure to check a proper option(s) in editor's menu(s) for saving files in UNIX format. Otherwise, the input files will not be compiled!

To start a reliability calculation just type: *sh re.c filename*, where *filename* is a name of an input file made by user (this file will be explained in details later in the text). Input files must have an extension **.s** (*filename.s*) but when 'calling' them with *sh re.c* command the extension must be omitted.

After calculation is finished, program automatically opens output file that contains calculation results: *filename.outr***.** It is also a plain text file. At this moment it is opened inside the **vi-editor** in UNIX which is really user unfriendly. To get out of this type: *:q* (you must type a colon too!). (Napomena: izbaciti iz skripti opciju da se rezultati automatski otvaraju u vi editoru!!)

Copying files to your PC

Use some FTP program (**WinFTP**, **Total Commander**,…) to copy files from UNIX system to your PC. Every file than can be edited in **Windows** environment using **notepad.exe** (or any other text editor).

Also, use the FTP program to copy input files (*filename.s*) from your PC to UNIX system. Keep in mind to copy input file (*filename.s*) to the *rex* directory on UNIX system, otherwise you will not get proper output file (*filename.outr*). Unfortunately, if you copy input file to the wrong directory, you will not see any massage about that. The reliability calculation will be performed and you will probably get an empty *.outr* file. The full path of the reliability directory is: */usr1/kalman/rex*. (Napomena: žiha kaže da koji put treba utipkati cijeli path!?)

Action in brief:

Prepare input file *filename.s* in **editor** (for example **Crimson Editor** in **Windows**). Copy input file *filename.s* with FTP program to */usr1/kalman/rex* on *titan.fsb.hr* (IP address:

161.53.116.9). Start **putty** > connect to *titan.fsb.hr* > enter username: *kziha* > enter password: *kziha48* > change to reliability directory: *cd rex* > run program *sh re.c filename*. Exit vi editor: :q. > copy output file *filename.outr* with FTP program to your PC > edit *filename.outr* (**notepad** will suffice) to view the results.

Shell scripts (programs)

re.c = shell script for running III-level FORM (AFOSM)

input file: *filename.s* (SH)

calling: *sh re.c filename*

mandatory data: *fort.31*

output file: *filename.outr*

rs.c = shell script for running II-level FORM (FOSM)

input file: *filename.s* (SH)

calling: *sh rs.c filename*

mandatory data: *fort.31*

output file: *filename.outs*

ms.c = shell script for calculating reliability with Monte Carlo simulation.

input file: *filename.s* (SH)

calling: *sh ms.c filename*

mandatory data: *fort.26, fort.37, fort.30* (for descriptive sampling)

output file: *filename.outm* (PRT)

se.c = shell script for preparation and running simulation tests (Monte Carlo simulation).

input file: *filename.s* (SH)

calling: *sh ms.c filename*

mandatory data: *fort.26*, *fort.37*

output file: *filename.x*

st.c = shell script for analyzing simulation tests results (Monte-Carlo simulation).

input file: *filename.s* (SH)

calling: *sh ms.c filename*

mandatory data: *fort.26*, *fort.37*

output file: *filename.oute* (PRT)

Auxiliary Data (Program's Data and Parameters)

fort.25 = safety indices and coordinates of the. indeksi sigurnosti i koordinate tocke najvece izglednosti (d)

fort.26 = parameters for running the simulation procedure. (d)

fort.30 = descriptive sampling data (FORTRAN output)

fort.31 = parameters for running AFOSM program. (d)

fort.32 = data for sensitivity analysis (delta ro?). (d)

fort.37 = parameters for running simulation experiments. (d)

Examples (input files)

ext1.s = simple reliability analysis example. (EX)

ext2.s = simple example, component reliability with independent random variables. (EX)

ex1.s = reliability analysis of tanker midship section (global beam). (EX)

ex2.s = reliability analysis of a hatch cover. (EX)

ex4.s = reliability analysis of a plane frame structure, Madsen. (EX)

ex4m.s = reliability analysis of a plane frame structure, Melchers. (EX)

ex5.s = reliability analysis of a cylindrical beam under axial loading. (EX)

ex6.s = reliability analysis of a console beam. (EX)

exc1.s = CALREL, Manual Example 1. (EX)

exc2.s = CALREL, Manual Example 2. (EX)

exc4.s = CALREL, Manual Example 4. (EX)

exc7.s = CALREL, Manual Example 7. (EX)

exc8.s = CALREL, Manual Example 8. (EX)

File extensions

.f = fortran code

.o = object module

.run = executable files

.d = data

Legend

(PP) = subprogram (subroutine)

(PPi) = subprograms (subroutines)

(BIBL) = library of subprograms

(DOC) = document

(DAT) = data

(INC) = include procedure

(SH) = shell procedure

(EX) = examples

(PRT) = print files

(d) = data

Input file description

Annotation: font styles (bold, italic) in the examples here are only for the clarification purposes. The actual input file has no styles at all (plain text).

General remarks

Input files are plain text files with an extension **.s**, e.g. *ext2.s*, which have to be edited (or written from scratch) in editor that is capable of saving files in UNIX format. It is recommended to use some programming editor which recognizes FORTRAN code (**GWDeditor**, **Crimson Editor** in **Windows**), but it is not necessary.

Although input files are plain text files it is important to follow some rules when writing them because each input file, e.g. *ext2.s*, actually consists of two parts: FORTRAN code and data. The code part begins with a keyword `HEADING` and the data part begins with a keyword `DATA`. In the compiling process the input file (*ext2.s*) will be separated into two files according to those keywords, i.e. compiling process will generate files *ext2.f* (FORTRAN code file) and *ext2.d* (data file). User does not need to take care about separation – the process is completely automatic.

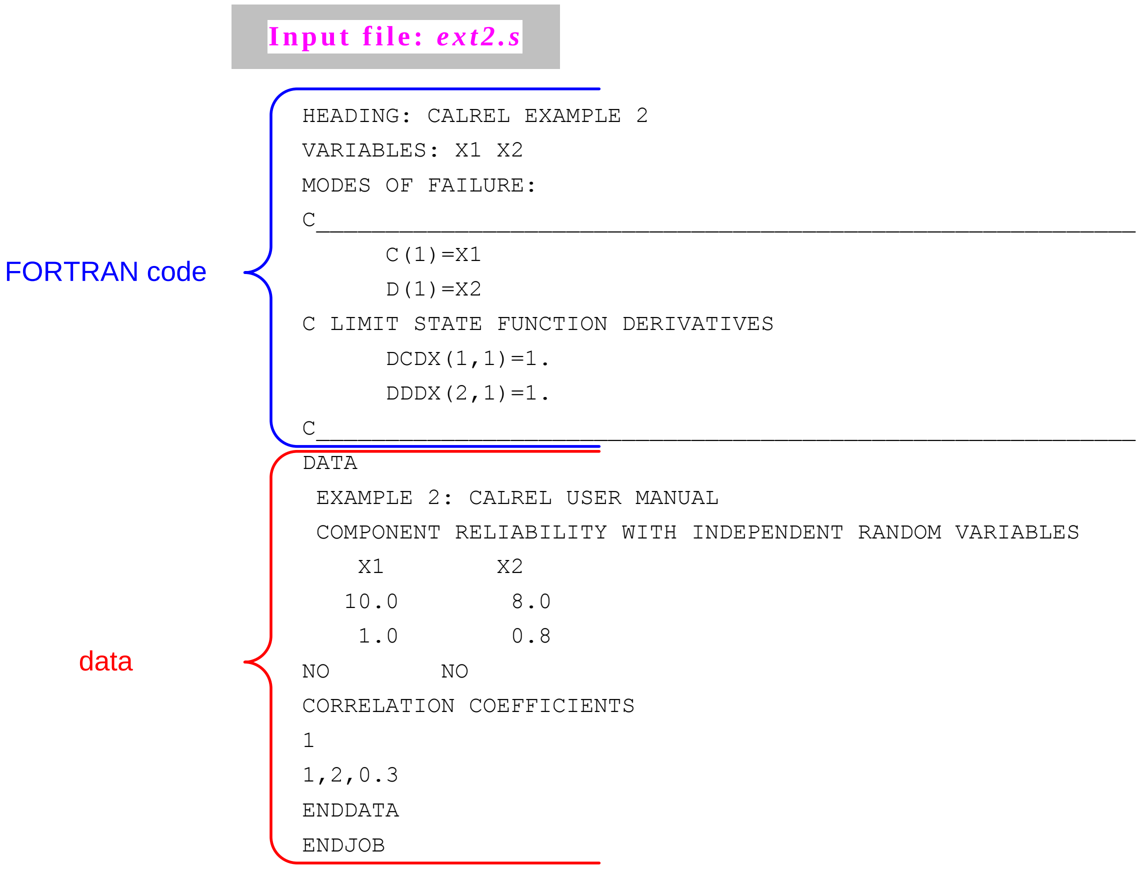
This separation into two files is important, from the user's point of a view, only regarding positioning text and data in the input file. Since all the reliability programs, that will use the input file, were written in FORTRAN user should take care about aligning text and data at appropriate distance from the left margin of the document (according to FORTRAN rules for entering data in columns!).

For example, in DATA part of the input file *ext2.s* (see below), there is a limit of 10 characters (a field) for each variable, starting with column 0 (left margin). Thus, if user enters even a single character outside that field, it will produce false results, or even will not start the compiling process at all!

IMPORTANT troubleshooting: if an outcome is unexpected, the most probable reason for that is a WRONG ALIGNING of the characters (text) in the input file!! The recommended way to make a new input file is to edit and overwrite some existing input file!

Example of input file: *ext2.s*

```
HEADING: CALREL EXAMPLE 2
VARIABLES: X1 X2
MODES OF FAILURE:
C___________________________________________________________
      C(1)=X1
      D(1)=X2
C LIMIT STATE FUNCTION DERIVATIVES
      DCDX(1,1)=1.
      DDDX(2,1)=1.
C___________________________________________________________
DATA
 EXAMPLE 2: CALREL USER MANUAL
 COMPONENT RELIABILITY WITH INDEPENDENT RANDOM VARIABLES
    X1        X2
   10.0        8.0
    1.0        0.8
NO        NO
CORRELATION COEFFICIENTS
1
1,2,0.3
ENDDATA
ENDJOB
```

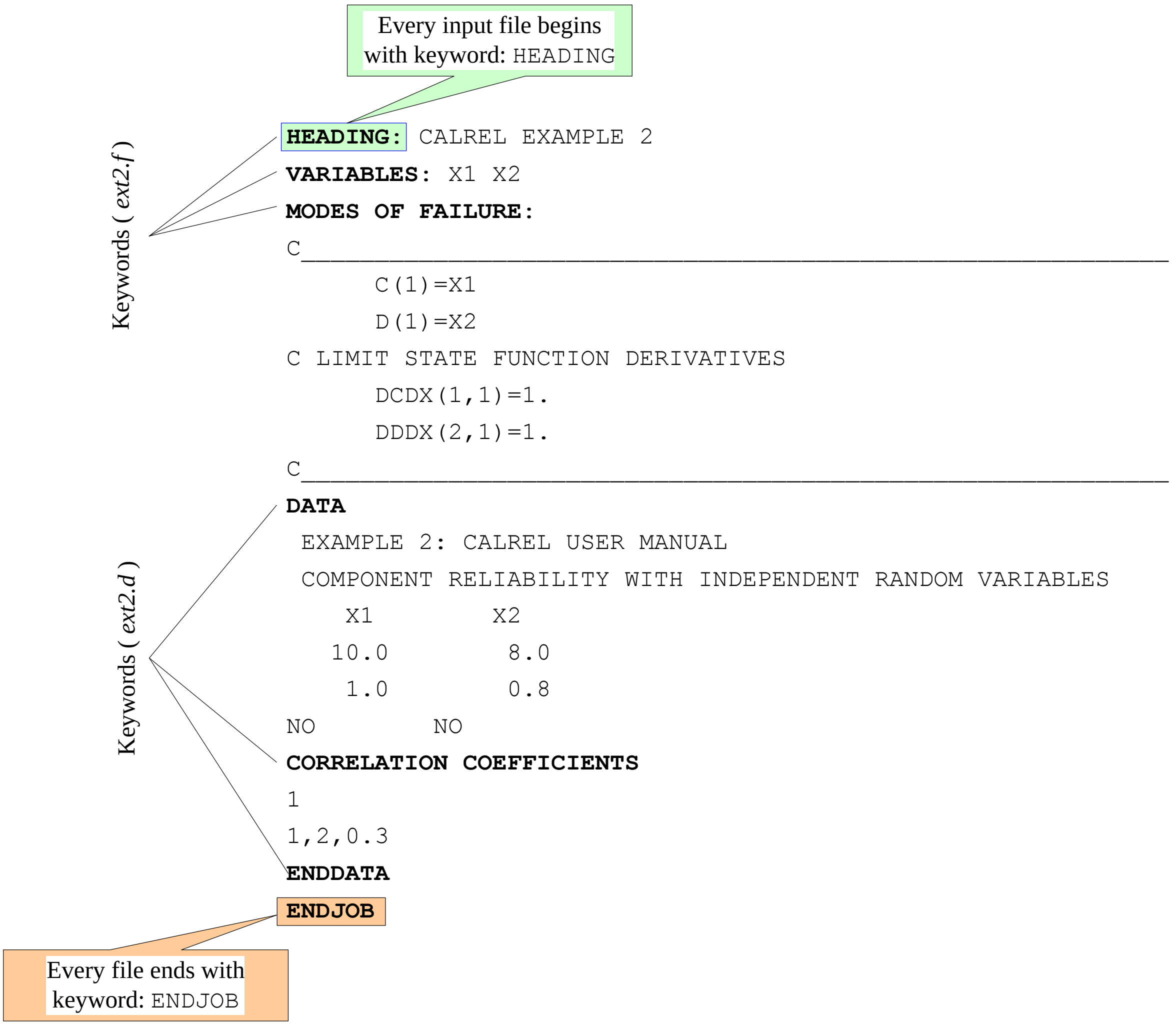


Input file: *ext2.s*

```
HEADING: CALREL EXAMPLE 2
VARIABLES: X1 X2
MODES OF FAILURE:
C___________________________________________________________
      C(1)=X1
      D(1)=X2
C LIMIT STATE FUNCTION DERIVATIVES
      DCDX(1,1)=1.
      DDDX(2,1)=1.
C___________________________________________________________
DATA
 EXAMPLE 2: CALREL USER MANUAL
 COMPONENT RELIABILITY WITH INDEPENDENT RANDOM VARIABLES
    X1        X2
   10.0        8.0
    1.0        0.8
NO        NO
CORRELATION COEFFICIENTS
1
1,2,0.3
ENDDATA
ENDJOB
```

Figure: input file parts

Input file: *ext2.s*



Every input file begins with keyword: `HEADING`

Keywords (*ext2.f*)

```
HEADING: CALREL EXAMPLE 2
VARIABLES: X1 X2
MODES OF FAILURE:
C___________________________________________________________
      C(1)=X1
      D(1)=X2
C LIMIT STATE FUNCTION DERIVATIVES
      DCDX(1,1)=1.
      DDDX(2,1)=1.
C___________________________________________________________
DATA
 EXAMPLE 2: CALREL USER MANUAL
 COMPONENT RELIABILITY WITH INDEPENDENT RANDOM VARIABLES
    X1         X2
   10.0         8.0
    1.0         0.8
NO         NO
CORRELATION COEFFICIENTS
1
1,2,0.3
ENDDATA
ENDJOB
```

Keywords (*ext2.d*)

Every file ends with keyword: `ENDJOB`

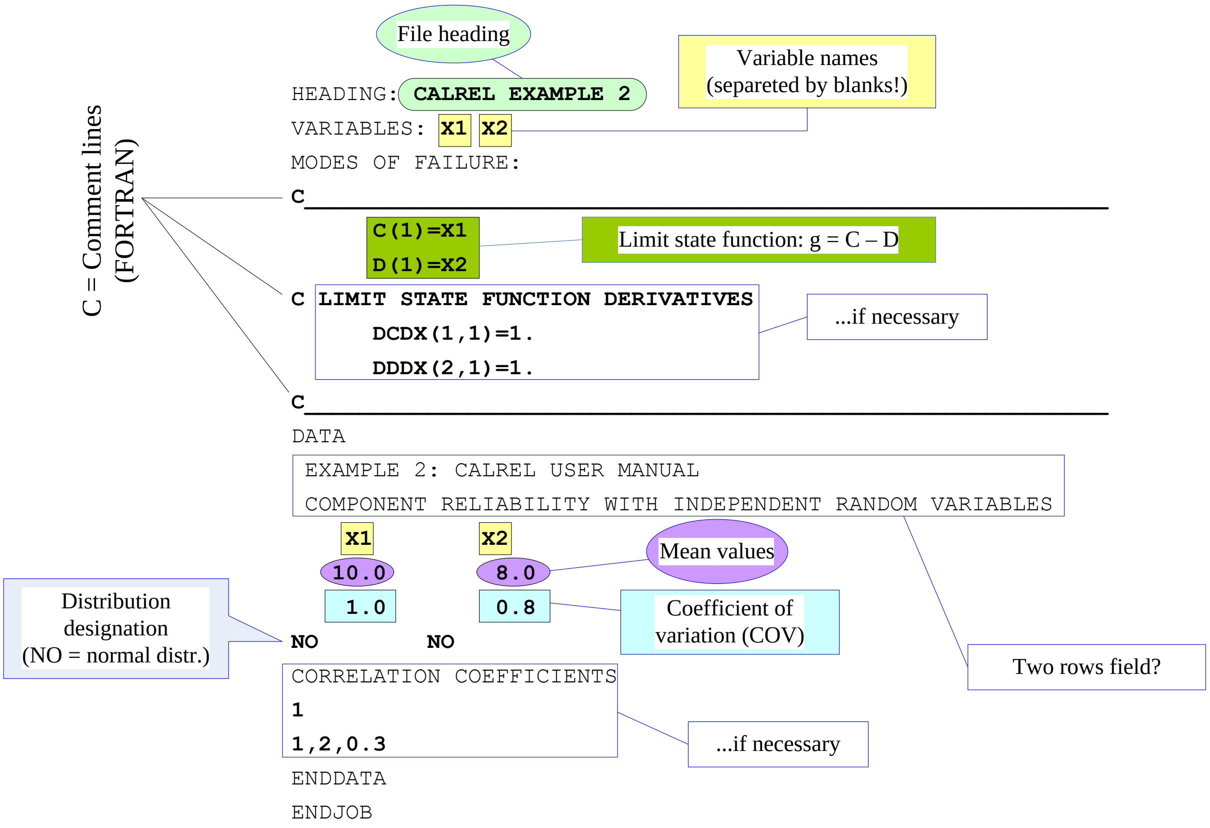
Figure: keywords

List of keywords (objasniti):

```
HEADING
DATA
ENDDATA
ENDJOB
VARIABLES
MODES OF FAILURE
LOADS
PARAMETERS
END
CORRELATION COEFFICIENTS
TYPE
```

Keywords cannot be used as variable names!

Input file: *ext2.s*



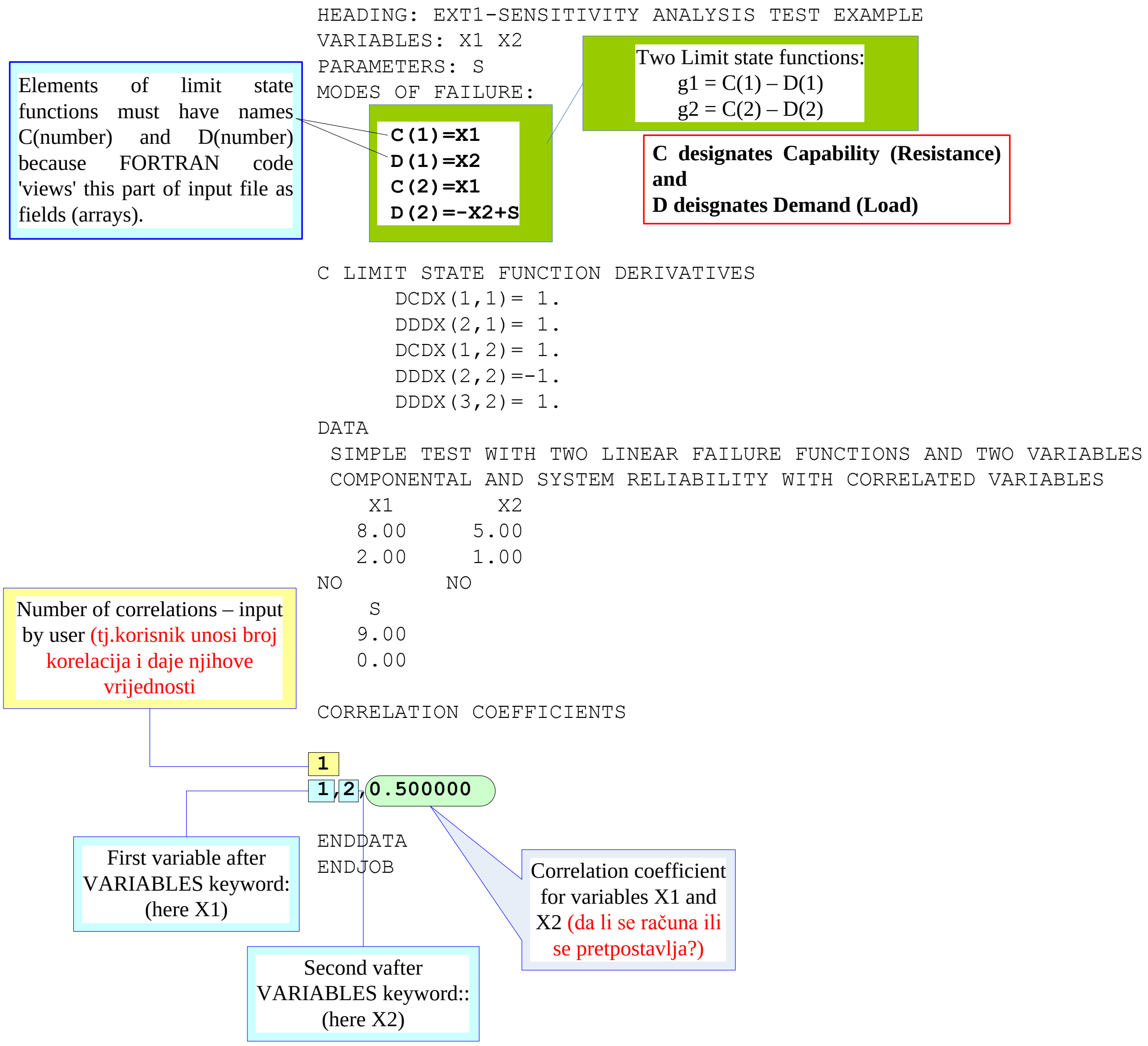
```
HEADING:CALREL EXAMPLE 2
VARIABLES: X1 X2
MODES OF FAILURE:
C
      C(1)=X1
      D(1)=X2
C LIMIT STATE FUNCTION DERIVATIVES
      DCDX(1,1)=1.
      DDDX(2,1)=1.
C
DATA
 EXAMPLE 2: CALREL USER MANUAL
 COMPONENT RELIABILITY WITH INDEPENDENT RANDOM VARIABLES
   10.0        8.0
    1.0        0.8
NO         NO
CORRELATION COEFFICIENTS
1
1,2,0.3
ENDDATA
ENDJOB
```

Figure: input file dissection

Variable names

Ne mogu biti:

Input file: *ext1.s*



```
HEADING: EXT1-SENSITIVITY ANALYSIS TEST EXAMPLE
VARIABLES: X1 X2
PARAMETERS: S
MODES OF FAILURE:
      C(1)=X1
      D(1)=X2
      C(2)=X1
      D(2)=-X2+S

C LIMIT STATE FUNCTION DERIVATIVES
      DCDX(1,1)= 1.
      DDDX(2,1)= 1.
      DCDX(1,2)= 1.
      DDDX(2,2)=-1.
      DDDX(3,2)= 1.
DATA
 SIMPLE TEST WITH TWO LINEAR FAILURE FUNCTIONS AND TWO VARIABLES
 COMPONENTAL AND SYSTEM RELIABILITY WITH CORRELATED VARIABLES
    X1         X2
   8.00       5.00
   2.00       1.00
NO         NO
    S
   9.00
   0.00

CORRELATION COEFFICIENTS
1
1,2,0.500000

ENDDATA
ENDJOB
```

Elements of limit state functions must have names C(number) and D(number) because FORTRAN code 'views' this part of input file as fields (arrays).

Two Limit state functions:
g1 = C(1) – D(1)
g2 = C(2) – D(2)

C designates Capability (Resistance) and
D deisgnates Demand (Load)

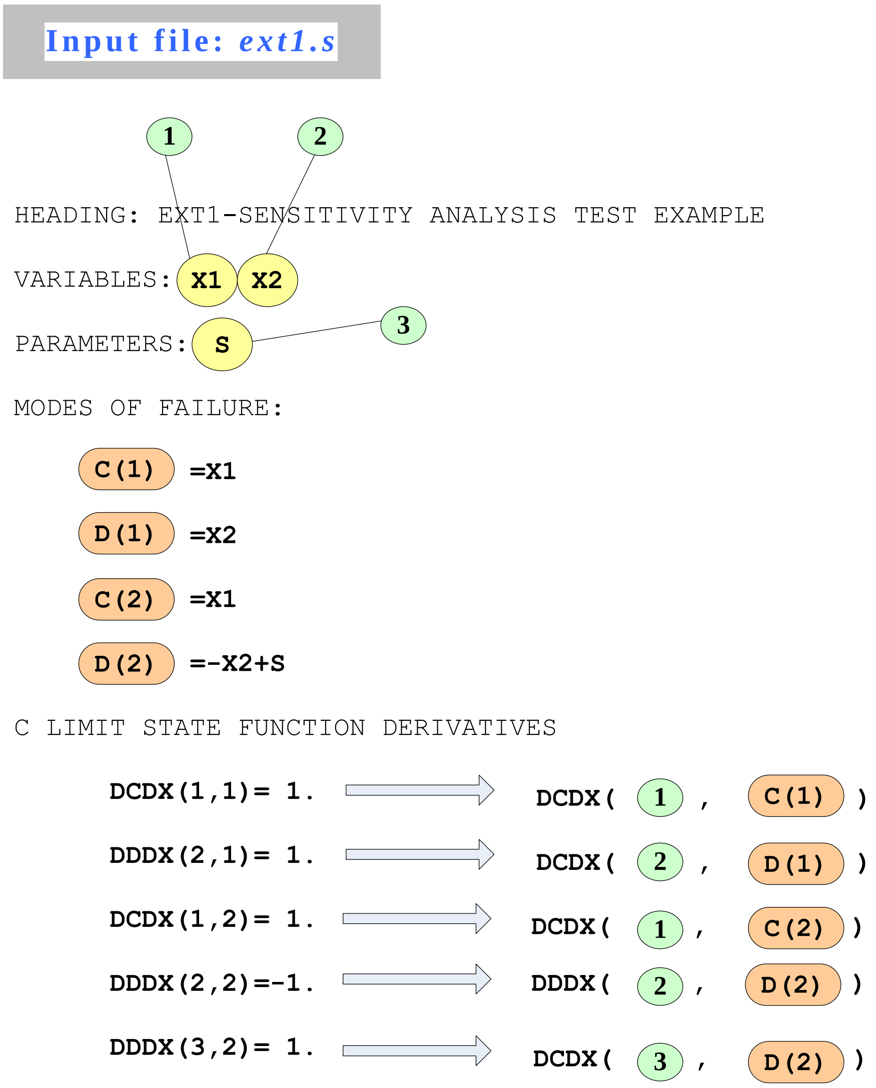
Number of correlations – input by user (tj.korisnik unosi broj korelacija i daje njihove vrijednosti

First variable after VARIABLES keyword: (here X1)

Second vafter VARIABLES keyword:: (here X2)

Correlation coefficient for variables X1 and X2 (da li se računa ili se pretpostavlja?)

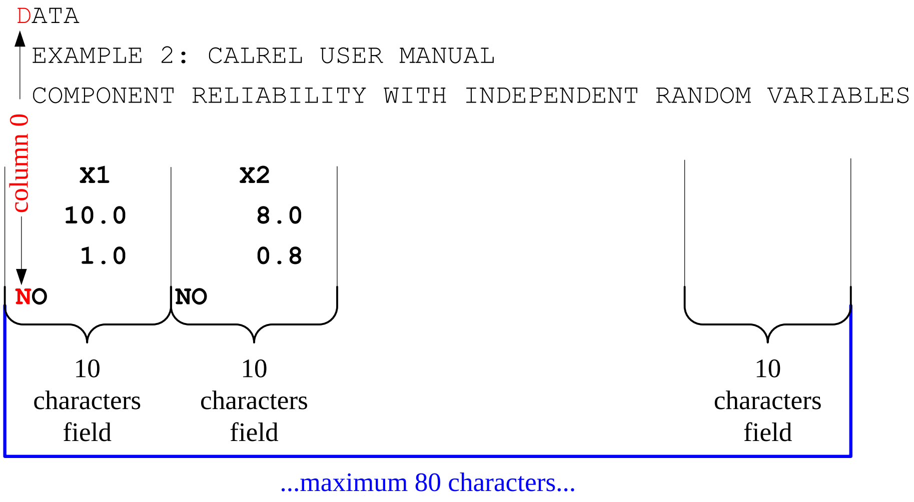
Figure: designation explanation for limit state functions and correlation coefficients



Input file: *ext1.s*

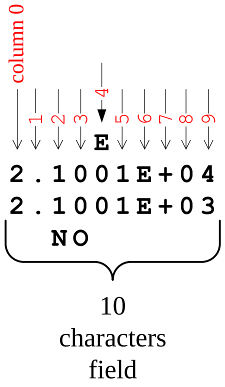
```
HEADING: EXT1-SENSITIVITY ANALYSIS TEST EXAMPLE
VARIABLES: X1 X2
PARAMETERS: S
MODES OF FAILURE:
     C(1) =X1
     D(1) =X2
     C(2) =X1
     D(2) =-X2+S
C LIMIT STATE FUNCTION DERIVATIVES
      DCDX(1,1)= 1.
      DDDX(2,1)= 1.
      DCDX(1,2)= 1.
      DDDX(2,2)=-1.
      DDDX(3,2)= 1.
```

Figure: limit state derivatives explanation



```
DATA
EXAMPLE 2: CALREL USER MANUAL
COMPONENT RELIABILITY WITH INDEPENDENT RANDOM VARIABLES
      X1        X2
    10.0       8.0
     1.0       0.8
NO        NO
```

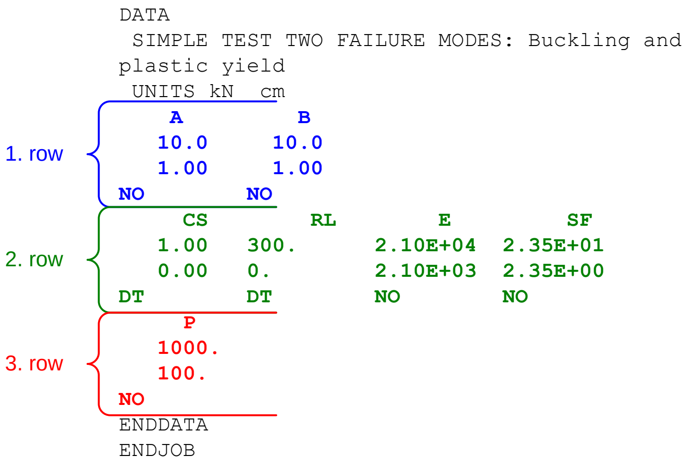
Figure: watch out!



```
    E
2.1001E+04
2.1001E+03
   NO
```

Figure: All number must be entered within 10 characters field!

You can always enter data in several rows:



```
DATA
 SIMPLE TEST TWO FAILURE MODES: Buckling and
plastic yield
 UNITS  kN   cm
    A         B
   10.0      10.0
   1.00      1.00
NO        NO
     CS        RL        E         SF
   1.00   300.      2.10E+04  2.35E+01
   0.00   0.        2.10E+03  2.35E+00
DT        DT        NO        NO
     P
   1000.
   100.
NO
ENDDATA
ENDJOB
```

Figure: data input

Distribution designations

NO = normal
LN = log – normal
DT =
GU =
N =
UN = uniform
EX =
WB = Weibull
L2 =