сообщения
объединенного
института
ядерных
исследований
дубна

E11-89-537

A.G.Grozin [1], H.Perlt [2]

THE ALGEBRAIC MANIPULATION PROGRAM DIRAC ON IBM PERSONAL COMPUTERS

[1] Institute of Nuclear Physics, SD of Academy of Sciences, Novosibirsk, USSR
[2] Permanent address: Karl Marx Universität Leipzig, Sektion Physik, Karl-Marx-Platz, Leipzig, DDR-7010

1989

1. Introduction

Computer algebra languages are firm part of mathematical, physical or technical researches. Especially, since the appearance of personal computers the community of users has been enlarged drastically. There are some comprehensive systems that are in common use, the best known among them is surely REDUCE /1/. Nevertheless, it seems to be justified to look for special smaller systems to solve selected problems if computer time and storage can be spared considerably. One of these is DIRAC /2,3/ - a system for algebraic manipulation with polynomial and tensorial expressions. It is based on PASCAL and was initially designed for main frame computers. Its advantages are the high speed of calculation and effective use of computer storage. These were the reasons to transcribe DIRAC for usage on a personal computer.

2. Implementation and usage

The version DIRAC (2.2) is based on TURBO PASCAL (versions 4.0 and 5.0) /4/. Therefore, it is restricted to IBM compatible personal computers. Two typical TURBO PASCAL features used should be mentioned explicitely: Integer coefficients are implemented as longint types (32 bits) - arbitrary large integers are not supported. Powers of scalars in polynomials occupy 1 byte and, therefore, cannot exceed 255. DIRAC (2.2) is available on one 360 kByte discette, containing the following files:

DIRAC.EXE
README.EXE
DIRAC.TXT

After typing 'DIRAC' the entrance menue appears on the screen. The user can select four possible modes by pressing the 'DOWN-ARROW' and/or 'ENTER':

<u>Information file</u> : it contains all presently implemented commands, flags and functions

<u>Help file</u> : it describes the usage of the implemented commands, flags by short examples

<u>Execution</u> : it switches to DIRAC

<u>Exit</u> : it switches to MS-DOS

After typing 'README' a short write-up of DIRAC (2.2) is displayed on the screen.

3. User's guide to DIRAC (2.2)

The following program example contains a comprehensive description of DIRAC (2.2). After reading this part the user is supposed to work with DIRAC (2.2) without difficulties. Additionally, there are given six useful examples. They illustrate the possibilities of DIRAC (2.2) to deal with typical problems often encountered in physical or mathematical calculations. A systematic error handling description is appended.

```
_ COM 'DIRAC IS A COMPUTER ALGEBRA SYSTEM FOR OPERATIONS WITH
_ POLYNOMIALS AND TENSORS. THIS IS AN EXAMPLE OF THE USER'S DIALOG
_ WITH DIRAC VERSION 2.2 ON IBM PC. LINES WITH USER'S INPUT ARE
_ PRECEEDED BY DIRAC'S PROMPT "_", LINES WITH DIRAC'S OUTPUT ARE
_ NOT. THE TEXT YOU ARE READING NOW IS PART OF A COMMENT. COMMENTS
_ MAY FILL AN ARBITRARY NUMBER OF LINES AND ARE TERMINATED BY THE
_ SAME CHARACTER IT STARTED WITH. ANY NON-BLANK CHARACTER MAY BE USED
_ AS A DELIMITER. IN THE PRESENTED CASE IT IS THE CHARACTER '
_ COM /THIS IS ANOTHER COMMENT. NOW THE DELIMITER IS :/
_ COM
_ END-OF-LINE ALSO SERVE AS A ONE-LINE COMMENT DELIMITER
_ TEXT 'THE COMMAND TEXT WRITES THE TEXT ENCLOSED BY THE DELIMITERS'
THE COMMAND TEXT WRITES THE TEXT ENCLOSED BY THE DELIMITERS

_
_ COM '--------------POLYNOMIALS----------------------------'

_
_ COM 'FIRST OF ALL, WE SHALL CONSIDER SOME EXAMPLES WITH POLYNOMIALS.
_ POLYNOMIALS HAVE RATIONAL (IN PARTICULAR INTEGER) COEFFICIENTS AND
_ MAY DEPEND ON SEVERAL SCALARS. ORDERS OF SMALLNESS CAN BE ASSIGNED
_ TO SCALARS, AND DIRAC WILL DROP OUT TERMS WHICH HAVE THE TOTAL
_ ORDER OF SMALLNESS GREATER THAN A GIVEN MAXIMUM. POLYNOMIALS CAN
_ BE ADDED, SUBTRACTED, MULTIPLIED, RAISED TO AN INTEGER POWER,
_ DIFFERENTIATED AND INTEGRATED. A POLYNOMIAL MAY BE SUBSTITUTED
_ FOR A PRODUCT OF POWERS OF SCALARS (IN PARTICULAR, FOR A SCALAR).'
_ SCALAR X,Y,Z,S:1,T:2;POLY A,B,C,D;ORDER 4
_ COM 'SCALAR S HAS ORDER 1 AND T HAS ORDER 2. TERMS UP TO THE
_ ORDER 4 WILL BE RETAINED. THUS, S^4, T^2, S^2*T WILL SURVIVE,
_ BUT S^5, S*T^2, S^3*T WILL NOT. INITIALLY, MAXIMUM ORDER IS 0,
_ AND ALL SMALL SCALARS ARE DROPPED OUT. THE USER SHOULD DEFINE
```



```
_ MAXIMUM  ORDER BEFORE ANY CALCULATION INVOLVING SMALL SCALARS!'
_ A=(1+2/3*X*Y^2-3/2*X^2*Y) WRITE #A
1+2/3*X*Y^2-3/2*X^2*Y
_ COM 'VALUE OF A VARIABLE IS RETAINED IF IT IS PRECEEDED BY #
_ AND OTHERWISE DELETED'
_ WRITE A
1+2/3*X*Y^2-3/2*X^2*Y
_ WRITE A
        ^
ERROR AT LINE  33 IN UNDEFINED ARGUMENT 1 : A
_ COM 'NEW VALUE CANNOT BE ASSIGNED TO A VARIABLE UNTIL OLD
_ ONE IS DELETED.'
_ A=0 A=1
        ^
ERROR AT LINE  36 IN REDEFINED RESULT : A
_ DELETE A A=(1+X+Y) B=#A^5 WRITE #B
1+5*Y+10*Y^2+10*Y^3+5*Y^4+Y^5+5*X+20*X*Y+30*X*Y^2+20*X*Y^3+5*X*Y^4+
10*X^2+30*X^2*Y+30*X^2*Y^2+10*X^2*Y^3+10*X^3+20*X^3*Y+10*X^3*Y^2+5*
X^4+5*X^4*Y+X^5
_ COM 'THE OUTPUT WAS REORGANISED FOR PRINTING; DIRAC USES 80
_ CHARACTERS PER LINE.'
_ C=#A^3 D=A^2 D=-D C=C*D B=B+C
_ COM 'RESULT SHOULD BE ZERO' WRITE B
0
_ A=(X+S+T) B=A^5 WRITE B
5*X*S^4+30*X^2*S^2*T+10*X^2*S^3+10*X^3*T^2+20*X^3*S*T+10*X^3*S^2+5*
X^4*T+5*X^4*S+X^5
_ COM 'TERMS WITH ORDER OF SMALLNES >4 ARE DROPPED OUT'
_ A=(1+X-X^2*Y-Y^2)
_ COM 'DERIVATIVE IN X'
_ B=DIF X:#A WRITE #B
1-2*X*Y
_ COM 'SECOND DERIVATIVE IN X AND FIRST IN Y'
_ C=DIF X^2*Y:#A WRITE C
-2
_ COM 'INTEGRAL IN X'
_ B=DIF X^-1:B WRITE #B
X-X^2*Y
_ COM 'B CONTAINS ALL TERMS OF A EXCEPT X INDEPENDENT ONES.'
_ B=-B B=B+A WRITE B
1-Y^2
_ A=(2+X-Y+3*X*Y-X^2*Y+X^2*Y^2) B=(Z)
_ COM 'SUBSTITUTE THE VALUE OF B FOR X*Y'
```

```
_ A=SUB X*Y=B:A WRITE A
2+3*Z+Z^2-Y+X-X*Z
_ COM 'A MAXIMUM POSSIBLE POWER OF X*Y IS SUBSTITUTED.'
_
_ COM '--------------FLAGS----------------------------------'
_
_ COM 'THE USER CAN INFLUENCE THE DIRAC'S BEHAVIOUR BY USING
_ FLAGS. IF THE FLAG <WRITE> IS TURNED ON, THEN THE RESULT OF
_ EVERY ASSIGNEMENT WILL BE PRINTED AUTOMATICALLY:'
_ +WRITE A=(X+Y) A=A^2 DELETE A -WRITE
A=(Y+X)
A=(Y^2+2*X*Y+X^2)
_ COM 'THE FLAG <LINE> REQUIRES TO START EVERY COMMAND FROM A NEW LINE.
_ ALL THE TEXT FROM THE END OF A COMMAND TO THE END OF LINE IS
_ TREATED AS A COMMENT. THIS OPTION ALLOWS TO EXPLAIN EVERY COMMAND
_ ON THE SAME LINE AND IS USEFUL IN PROGRAMS WRITTEN TO A FILE
_ FOR REPEATED USE.'
_ +LINE
_ A=0          THIS TEXT IS TREATED
_ DELETE A     AS A COMMENT
_ -LINE
_ COM 'THE FLAG <ECHO> CAUSES COPYING OF INPUT LINES TO OUTPUT.
_ THEY ARE NUMBERED. THE NUMBER OF A LINE CAN BE CHANGED BY
_ THE COMMAND CONTAINING SIMPLY THE NEW NUMBER.'
_ +ECHO COM 'IT WILL BE ACTIVE FROM THE NEXT LINE.'
_ A=123 WRITE #A DELETE A COM 'THIS IS THE REST OF THE LINE'
 93  A=123 WRITE #A
123
DELETE A COM 'THIS IS THE REST OF THE LINE.'
_ TEXT 'FIRST LINE
 94 TEXT 'FIRST LINE
FIRST LINE
_ SECOND LINE'
 95 SECOND LINE'
SECOND LINE

_ 500 COM 'THE CHANGE OF THE LINE NUMBER WILL BE ACTIVE'
 96 500 COM 'THE CHANGE OF THE LINE NUMBER WILL BE ACTIVE'
_ COM 'FROM THIS LINE.'
501 COM 'FROM THIS LINE.'
_-ECHO COM 'THIS WILL BE ACTIVE FROM THE NEXT LINE TOO.'
502 -ECHO COM 'THIS WILL BE ACTIVE FROM THE NEXT LINE TOO.'
_ COM 'THIS LINE IS NOT ECHOED.'
```



```
_ COM 'THE FLAG <DIALOG> IS INITIALLY ON. IT CAUSES DIRAC TO ISSUE
_ PROMPTS AND INFLUENCES THE REACTION ON ERRORS. IT IS TURNED OFF,
_ AUTOMATICALLY, WHEN INPUT OR OUTPUT ARE CONNECTED WITH A FILE.'
_ COM 'ALGEBRAIC COMPUTATIONS CAN TAKE A LOT OF TIME. THE USER MAY
_ WANT TO KNOW WHEN DIRAC IS READY FOR A NEW COMMAND. THE FLAG
_ <SOUND> CAUSES DIRAC TO ISSUE A SOUND SIGNAL EVERY TIME IT WRITES
_ A PROMPT.'
_ +SOUND
_ COM 'UNFORTUNATELY, YOU CANNOT HEAR THE SIGNAL WHEN READING THIS
_ LISTING'
_ COM ' SOMETIMES IT IS INTERESTING TO KNOW THE TIME USED BY DIRAC
_ FOR CALCULATION. FLAG <TIME> CAUSES DIRAC TO WRITE EXECUTION TIME
_ FOR EVERY COMMAND. HERE IS A RATHER LENGTHY EXAMPLE:'
_ +TIME
_ A=(X+Y+Z) A=A^10 B=+#A A=A*B
TIME 0 MSEC
TIME 220 MSEC
TIME 0 MSEC
TIME 18670 MSEC
_ DELETE A
TIME 60 MSEC
_ -TIME -SOUND
_
_ COM '--------------TENSORS--------------------------------'
_
_ COM 'NOW WE TURN TO SOME EXAMPLES WITH TENSORS. TENSORS CONSISTS
_ OF TENSOR STRUCTURES WITH POLYNOMIAL COEFFICIENTS. TENSOR STRUCTURES
_ ARE MADE OF VECTORS WITH INDICES, METRIC TENSOR AND UNIT
_ ANTISYMMETRIC TENSOR. SPACE DIMENSION MAY BE ARBITRARY, IN
_ PARTICULAR IT MAY BE GIVEN BY AN ANALYTIC EXPRESSION. ALL SQUARES
_ AND SCALAR PRODUCTS OF VECTORS MUST BE EXPRESSED THROUGH SCALARS.
_ TENSORS MAY BE ADDED, SUBTRACTED, MULTIPLIED WITH  CONTRACTION
_ OVER REPEATED INDICES, DIFFERENTIATED WITH RESPECT TO A VECTOR.
_ A TENSOR MAY BE SUBSTITUTED FOR A TENSOR STRUCTURE MULTIPLIED BY
_ A PRODUCT OF POWERS OF SCALARS. SOME INDICES MAY BE REGARDED AS
_ FORMAL: SUBSTITUTION WILL BE APPLIED AT ANY ACTUAL VALUES OF THESE
_ INDICES. TENSORS MAY BE ALSO DIFFERENTIATED AND INTEGRATED WITH
_ RESPECT TO SCALARS; POLYNOMIAL SUBSTITUTIONS CAN BE APPLIED
_ TO TENSORS AS WELL.'
_ +WRITE +LINE
_ VECTOR U,V,W,P;         DECLARATION OF VECTORS
_ TENSOR E,F,G;           DECLARATION OF TENSOR VARIABLES
_ DIM(4)                  SPACE DIMENSION
```

```
_ COM 'INITIAL VALUE OF SPACE DIMENSION IS 0. THIS LEADS TO SOME
_ STRANGE RESULTS. THE USER SHOULD DEFINE SPACE DIMENSION BEFORE
_ ANY OPERATIONS WITH TENSORS.'
_ EPS 4                    NUMBER OF INDICES OF ANTISYMMETRIC TENSOR
_ COM 'INITIALLY, EPS IS 0. IN THIS CASE CALCULATION WITH THE
_ UNIT ANTISYMMETRIC TENSOR IS IMPOSSIBLE.'
_ INDEX L,M,N,R;
_ (U.U=X)
_ (V.V=Y)                  SCALAR PRODUCTS
_ (U.V=Z)
_ COM 'INITIALLY, ALL SCALAR PRODUCTS ARE 0. THE USER SHOULD DEFINE
_ ALL NON-ZERO SCALAR PRODUCTS BEFORE ANY TENSOR CALCULATIONS.'
_ E=(U.M*U.N+M.N)     U.M - VECTOR WITH INDEX,M.N - METRIC TENSOR
E=(1*N.M+1*U.M*U.N)
_ A=(1+X)
A=(1+X)
_ E=E*A             TENSOR MULTIPLIED BY POLYNOMIAL (DEF. ORDER!)
E=((1+X)*N.M+(1+X)*U.M*U.N)
_ DELETE E
_ E=((1+Z)*U.M*L.N*U.R+U.M*U.N*U.L*U.R)
E=((1+Z)*N.L*U.M*U.R+1*U.L*U.M*U.N*U.R)
_ F=(L.M*N.R)
F=(1*M.L*R.N)
_ E=E*F             TENSOR WITHOUT INDICES
E=((X+X*Z+X^2))
_ A=+E              IT MAY BE COPIED INTO POLYNOMIAL VARIABLES
A=(X+X*Z+X^2)
_ E=-A              AND VICE VERSA
E=((-X-X*Z-X^2))
_ DELETE E
_ E=([U,V,M,N])     I* UNIT ANTISYMMETRIC TENSOR
E=(1*[M,N,U,V])
_ -LINE F=+#E E=E*F DELETE E
F=(1*[M,N,U,V])
E=((-2*Z^2+2*X*Y))
_ E=(U.M*U.N*V.L)
E=(1*V.L*U.M*U.N)
_ COM 'DERIVATIVE IN U.R' F=DIF U.R:#E DELETE F
F=(1*V.L*R.M*U.N+1*V.L*U.M*R.N)
_ COM 'DERIVATIVE IN U.L' F=DIF U.L:#E DELETE F
F=(1*U.M*V.N+1*V.M*U.N)
_ COM 'DERIVATIVE IN U.M' F=DIF U.M:E DELETE F
F=(5*V.L*U.N)
```



```
_49 E=((1+X)*U.M*U.N)
E=((1+X)*U.M*U.N)
_F=(V.M-W.M)
F=(1*V.M-1*W.M)
_ COM 'SUBSTITUTE THE VALUE OF F FOR X*U.M IN E'
_ G=SUB:X*U.M=F:E DELETE G
G=(1*U.M*U.N+1*V.M*U.N-1*W.M*U.N)
_ E=(V.L*U.M*U.N+V.M*U.N*U.L+V.N*U.M*U.L) F=(M.N/4)
E=(1*U.L*U.M*V.N+1*U.L*V.M*U.N+1*V.L*U.M*U.N)
F=(1/4*N.M)
_ COM 'FOR ALL M,N SUBSTITUTE THE VALUE OF F FOR U.M*U.N IN E'
_ G=SUB M,N:U.M*U.N=F:E DELETE G
G=(1/4*M.L*V.N+1/4*N.L*V.M+1/4*V.L*N.M)
_ E=(U.L*[P,W,U,M]) F=(U.M*V.N-U.N*V.M)
E=(1*U.L*[M,U,W,P])
F=(1*U.M*V.N-1*V.M*U.N)
_ COM 'IN [...] A VECTOR CAN APPEAR ON THE PLACE OF A FORMAL INDEX'
_ G=SUB M,N:[P,W,M,N]=F:E DELETE G
G=(-Z*U.L*U.M+X*U.L*V.M)
_ COM 'A TENSOR SUBSTITUTION IS APPLIED TO EVERY TERM AT MOST
_ ONCE. THEREFORE, IT DIFFERS FROM A POLYNOMIAL SUBSTITUTION
_ EVEN IF ITS LEFT HAND SIDE CONTAINS ONLY SCALARS:'
_ E=((1+X*Y+X^2*Y^2)*U.M) A=(Z) F=+#A
E=((1+X*Y+X^2*Y^2)*U.M)
A=(Z)
F=(Z)
_ COM 'POLYNOMIAL SUBSTITUTION' G=SUB X*Y=A:#E DELETE G
G=((1+Z+Z^2)*U.M)
_ COM 'TENSOR SUBSTITUTION'   G=SUB:X*Y=F:#E DELETE G
G=((1+Z+X*Y*Z)*U.M)
_ COM 'FIRST DERIVATIVE IN X AND SECOND IN Y'
_ G=DIF X*Y^2:#E DELETE G
G=(4*X*U.M)
_ COM 'INTEGRAL IN X AND Y' G=DIF X^-1*Y^-1:E DELETE G
G=((X*Y+1/4*X^2*Y^2+1/9*X^3*Y^3)*U.M)
_
_ COM '--------------DECLARATIONS---------------------------'
_
_ COM 'NOW WE SHALL DISCUSS DECLARATIONS IN MORE DETAIL. FOR EACH
_ CLASS OF OBJECTS (SCALARS, INDICES, VECTORS, POLYNOMIALS AND
_ TENSORS) THERE EXISTS A GLOBAL NAME TABLE. DECLARATIONS FILL AND
_ CHANGE THESE TABLES. A DECLARATION MAY END WITH "?" INSTEAD OF
_ ";". IN THIS CASE THE NEW STATE OF THE TABLE IS PRINTED.'
```

```
_ POLY?
POLY A,B,C,D,,,,,,,,,,,,,,,,,,,,,,,,,,,;
_ POLY A1,,C1?
POLY A1,B,C1,D,,,,,,,,,,,,,,,,,,,,,,,,,,,;
_ COM 'USUALLY, DECLARATION ELEMENTS ACTS ON SUCCESSIVE POSITIONS
_ IN THE TABLE. BUT IF AN ELEMENT IS PRECEEDED BY /OLDNAME/, THEN
_ ALL POSITIONS IN THE TABLE UP TO "OLDNAME" WILL BE SKIPPED.
_ FOR EXAMPLE, IF WE WANT TO RENAME A1 TO A AND C1 TO C, WE CAN
_ DO:'
_ POLY /A1/A,/C1/C?
POLY A,B,C,D,,,,,,,,,,,,,,,,,,,,,,,,,,,;
_ COM 'AN ESPECIALLY USEFUL CASE OF THIS IS SKIPPING TO THE FIRST
_ EMPTY PLACE IN THE TABLE. IF WE WANT TO ADD A POLYNOMIAL
_ VARIABLE H TO ALL EXISTING ONES, WE CAN DO:'
_ POLY //H?
POLY A,B,C,D,H,,,,,,,,,,,,,,,,,,,,,,,,,,;
_ COM 'SCALARS HAVE AN ADDITIONAL ATTRIBUTE : ORDER OF SMALLNESS:'
_ SCALAR?
SCALAR X:0,Y:0,Z:0,S:1,T:2,,,;
_ COM 'TO CHANGE THE NAME X TO X1 AND THE ORDER OF SMALLNESS OF Z
_ TO 3, WE CAN DO:'
_ SCALAR X1,,:3?
SCALAR X1:0,Y:0,Z:3,S:1,T:2,,,;
_ COM 'TO CHANGE THE ORDER OF SMALLNESS OF Z BACK TO 0, WE CAN DO:'
_ SCALAR /Z/:0?
SCALAR X1:0,Y:0,Z:0,S:1,T:2,,,;
_ COM 'DUE TO THE INTERNAL REPRESENTATION OF TENSORS, FIRST EPS
_ POSITIONS OF THE INDEX TABLE ARE NOT USED. THEREFORE, WHEN EPS
_ IS CHANGED, THE NUMBER OF AVAILABLE INDEX POSITIONS IS CHANGED
_ ALSO.'
_ EPS?
EPS 4
_ INDEX?
INDEX L,M,N,R;
_ EPS 0 INDEX?
INDEX ,,,,L,M,N,R;
_ INDEX L1,M1,N1,R1;
INDEX L1,M1,N1,R1,L,M,N,R;
_ EPS 3 INDEX?
INDEX R1,L,M,N,R;
_ COM 'THEREFORE, IT IS RECOMMENDED TO SET EPS BEFORE DECLARING
_ INDICES!'
_
```



```
_ COM '--------------FILES-----------------------------------'
_
_ COM 'OFTEN IT IS USEFUL TO READ FROM A FILE OR TO WRITE TO A FILE.
_ A FREQUENTLY USED SEQUENCE OF COMMANDS MAY BE GROUPED INTO A
_ FILE, AND IT CAN BE READ AS OFTEN AS REQUIRED. SUCH A FILE ACTS
_ LIKE A SUBROUTINE WITH ALL INFORMATION TRANSFER IS REALISED
_ ONLY THROUGH GLOBAL VARIABLES. AT THE BEGINNING OF AN INTERACTIVE
_ SESSION IT MAY BE USEFUL TO READ A FILE WITH SOME DECLARATIONS
_ AND DEFINING COMMANDS. THE COMMAND <FILE REDIRECTS INPUT FROM
_ KEYBORD TO THE SPECIFIED FILE. ANY MS-DOS FILE SPECIFICATION
_ CAN BE USED; THE DEFAULT EXTENSION IS .DIR. THE COMMAND <FILE
_ AUTOMATICALLY TURNS OFF THE FLAG <DIALOG>. THE LAST COMMAND IN SUCH
_ A FILE MUST BE HALT. IT RETURNS INPUT TO THE KEYBORD (AND OUTPUT
_ TO THE SCREEN), AND TURNS ON THE FLAG <DIALOG>. SUPPOSE THAT THE
_ FILE INFILE.DIR CONTAINS THE FOLLOWING LINES :
_               TEXT /LINE 1/
_               TEXT /LINE 2/
_               HALT
_ THEN READING THE FILE RESULTS IN:'
_ <INFILE
LINE 1
LINE 2
_ COM 'IN ORDER TO PREPARE RESULTS OF SOME CALCULATION FOR LATER USE
_ AS INPUT OR FOR PRINTING, THEY SHOULD BE WRITTEN TO A FILE. THE
_ COMMAND >FILE REDIRECTS OUTPUT FROM THE SCREEN TO THE SPECIFIED
_ FILE (AND TURNS OFF THE FLAG <DIALOG>).THE COMMANDS WRITE AND TEXT
_ CAN BE USED TO GENERATE THE WANTED OUTPUT.'
_ -LINE
_ A=(X+Y) B=#A^2
_ >OUTFILE
TEXT '+ECHO'
TEXT 'A=(' WRITE A TEXT ')'
TEXT 'B=(' WRITE B TEXT ')'
TEXT 'HALT'
HALT
_ <OUTFILE
  2 A=(
  3 Y+X
  4 )
  5 B=(
  6 Y^2+2*X*Y+X^2
  7 )
  8 HALT
```

```
_ -ECHO
  1 -ECHO
_ WRITE #A WRITE #B
Y+X
Y^2+2*X*Y+X^2
_ COM 'IT IS OFTEN USEFUL TO EXECUTE A PROGRAM IN BATCH MODE. IT IS
_ MUCH EASIER TO CORRECT AN ERROR INSIDE A FILE AND TO REEXECUTE
_ IT THAN TO REPEAT AN INTERACTIVE SESSION. SUPPOSE THE PROGRAM
_ IS IN THE FILE "PROGARM.DIR". THEN IT IS ENOUGH TO INPUT FROM
_ KEYBORD ONLY ONE LINE:
_      >LISTING <PROGRAM
_ THE RESULT WILL BE STORED IN THE FILE "LISTING.DIR". IT IS USEFUL
_ TO BEGIN A PROGRAM WITH +ECHO. A PROGRAM SHOULD BE ENDED WITH
_ <END> RATHER THAN WITH <HALT>. THE COMMAND <END> TERMINATES
_ EXECUTION MODE OF DIRAC.'
_ COM 'THERE ARE MANY DIRAC COMMANDS THAT GENERATES A LEGAL INPUT
_ FOR DIRAC. THEREFORE, FILES FOR INPUT MAY  CONVENIENTLY BE
_ GENERATED FROM A DIRAC SESSION. THE CURRENT ENVIROMENT CAN  BE
_ SAVED WITH THE HELP OF THE FOLLOWING COMMANDS:'
_ SCALAR?
SCALAR X:0,Y:0,Z:0,S:1,T:2,,,;
_ POLY?
POLY A,B,C,D,H,,,,,,,,,,,,,,,,,,,,,,,,,,;
_ ORDER?
ORDER 4
_ VECTOR?
VECTOR U,V,W,P,,,,;
_ DIM?
DIM(4)
_ EPS?
EPS 3
_ INDEX?
INDEX R1,L,M,N,R;
_ TENSOR?
TENSOR E,F,G,,,,,,,,,,,,,,,,,,,,,,,,,,,,;
_ COM 'SAVING FLAGS'
_ ?WRITE
+WRITE
_ ?ECHO
+ECHO
_ ?LINE
+LINE
_ COM 'AND SO ON'
```



```
_ COM 'SAVING SCALAR PRODUCTS'
_ (U.U)
(U.U=X)
_ (U.V)
(U.V=Z)
_ (V.V)
(V.V=Y)
_ COM 'AND SO ON'
_ COM 'SAVING VARIABLES A AND B'
_ +WRITE
_ A=+A
A=(Y+X)
_ B=+B
B=(Y^2+2*X*Y+X^2)
_ COM 'AND SO ON'
_ -WRITE
_ COM 'THESE INFORMATION COMMANDS MAY BE USEFUL ALSO DURING
_ AN INTERACTIVE SESSION'
_ DELETE A
_ DELETE B
_
_ COM '--------------EXAMPLE 1-------------------------------'
_ COM 'DIRAC CANNOT WORK WITH NEGATIVE POWERS OF SCALARS. IF IT
_ NECESSARY TO USE BOTH NEGATIVE AND POSITIVE POWERS OF A SCALAR X,
_ WE CAN INTRODUCE ANOTHER SCALAR X1 TO DENOTE X^(-1). POLYNOMIALS
_ CONTAINING BOTH X AND X1 SHOULD BE SIMPLIFIED BY USING THE
_ IDENTITY X*X1=1.
_ FOR EXAMPLE:'
_ SCALAR //X1; POLY //UNIT; UNIT=1
_ A=(X+X1) A=A^2 A=SUB X*X1=#UNIT:A WRITE A
2+X1^2+X^2
_
_ COM '--------------EXAMPLE 2-------------------------------'
_ COM 'SUPPOSE WE WANT TO DIVIDE A POLYNOMIAL A BY (X-1).
_ IT IS POSSIBLE IF A BECOMES ZERO AFTER SUBSTITUTING 1
_ FOR X. IN THIS CASE WE CAN MAKE THE SHIFT X -> X+1, DIVIDE
_ BY X AND MAKE THE SHIFT BACK. THIS METHOD IS REALISED BY
_ THE FOLLOWING PROGRAM:'
_ A=(X^2-1)
_ B=1 B=SUB X=B:#A
_ IF0 B
_     B=(X+1) A=SUB X=B:A
_     B=(X1) A=A*B A=SUB X*X1=#UNIT:A
```

```
_      B=(X-1) A=SUB X=B:A
_ ELSE
_      TEXT 'FAILED TO DIVIDE'
_ ENDIF
_ WRITE A
1+X
_ COM 'HERE, THE CONDITIONAL OPERATOR OF DIRAC HAS BEEN USED.
_ IT CHECKS IF B IS 0. IF IT IS, THEN THE FIRST SEQUENCE OF
_ COMMANDS IS USED; IF IT IS NOT THE SECOND ONE (AFTER ELSE). THE
_ VALUE OF B IS DELETED AFTER CHECK; IN ORDER TO RETAIN IT
_ ONE SHOULD USE #B. THE ELSE-PART OF THE CONDITIONAL OPERATOR
_ MAY BE OMITTED. CONDITIONAL OPERATORS MAY BE NESTED.'
_
_ COM '--------------EXAMPLE 3------------------------------'
_ COM 'VERY OFTEN MORE GENERAL EXPRESSIONS THAN POLYNOMIALS
_ ARE NEEDED. IN SOME CASES IT IS POSSIBLE TO DENOTE COMPLICATED
_ FUNCTIONS OR 1/(...) BY ADDITIONAL SCALARS. AFTER THAT, SUCH
_ EXPRESSIONS CAN BE REPRESENTED AS POLYNOMIALS IN THIS EXTENDED
_ SET OF SCALARS. FOR EXAMPLE, LET LNX DENOTE LOG(X). WE CAN
_ FORM EXPRESSIONS LIKE THIS ONE:'
_ SCALAR //LNX;A=(1+LNX-Z*LNX^2+2*Z*LNX^3) C=+#A
_ COM 'SUPPOSE WE WANT TO INTEGRATE THIS EXPRESSION OVER X.
_ WE SHOULD FIRST CONSTRUCT THE TABLE OF ALL NECESSARY
_ INTEGRALS. RECURRENT RELATIONS CAN BE USED TO EXPRESS THESE
_ INTEGRALS THROUGH SIMPLER ONES. LET US DENOTE THE INTEGRAL
_ SYMBOL BY THE SCALAR INT. AN INTEGRATION PROGRAM FOR THIS EXAMPLE
_ MAY LOOK LIKE:'
_ SCALAR //INT;B=(INT) A=A*B
_ B=(X*LNX^3-3*INT*LNX^2) A=SUB INT*LNX^3=B:A
_ B=(X*LNX^2-2*INT*LNX)   A=SUB INT*LNX^2=B:A
_ B=(X*LNX-INT)           A=SUB INT*LNX=B:A
_ B=(X)                   A=SUB INT=B:A
_ WRITE #A
X*LNX-14*X*Z+14*X*Z*LNX-7*X*Z*LNX^2+2*X*Z*LNX^3
_ COM 'IT IS IMPORTANT THAT SUCH A TABLE MUST BE USED FROM HIGHER
_ POWERS TO LOWER ONES. IN THE OPPOSITE CASE THE SUBSTITUTION FOR
_ E.G. INT*LNX WILL BE APPLIED TO THE TERM INT*LNX^2, AND THE
_ RESULT WILL BE INCORRECT.'
_ COM 'AS A CHECK WE CAN PERFORM THE DIFFERENTIATION OF THIS
_ EXPRESSION:'
_ POLY //D1;
_ D=DIF X:#A
_ B=DIF LNX:A D1=(X1) B=B*D1 D=D+B
```



```
_ D=SUB X*X1=#UNIT:D
_ C=-C C=C+D WRITE C
0
_
_ COM '----------- EXAMPLE 4-------------------------------'
_ COM 'DIRAC CAN CALCULATE PARTIAL DERIVATIVES OF TENSORS WITH
_ RESPECT TO VECTORS EXPLICITLY OCCURING IN TENSOR STRUCTURES.
_ BUT OFTEN SOME SCALARS DEPEND IMPLICITLY ON VECTORS. ONE SHOULD
_ TAKE IT INTO ACCOUNT WHEN CALCULATING TOTAL DERIVATIVES. FOR
_ EXAMPLE, SCALARS X AND Z DEPEND ON VECTOR U. LET S ALSO DENOTE
_ 1/(U.U-1). THEN CALCULATION OF THE TOTAL DERIVATIVE WITH RESPECT
_ TO U.M CAN BE DONE IN SUCH A WAY:'
_ TENSOR //Q;
_ E=(S*U.L*U.N+X*U.L*V.N-Z*V.L*V.N)
_ H=DIF U.M:#E
_ F=DIF X:#E G=(2*U.M)       F=F*G Q=Q+F
_ F=DIF Z:#E G=(V.M)         F=F*G Q=Q+F
_ F=DIF S:E  G=(-2*S^2*U.M)  F=F*G Q=Q+F
_ TEXT 'TOTAL DERIVATIVE IS'  WRITE Q
TOTAL DERIVATIVE IS
S*M.L*U.N+X*M.L*V.N+S*U.L*N.M-2*S^2*U.L*U.M*U.N+2*U.L*U.M*V.N-
1*V.L*V.M*V.N
_
_ COM '--------------EXAMPLE 5------------------------------'
_ COM 'SUPPOSE WE WANT TO AVERAGE A TENSOR OVER DIRECTIONS OF
_ THE VECTOR U. THERE ARE EXPLICIT FORMULAS FOR AVERAGES OF
_ THE PRODUCTS OF 2,4,6,... VECTORS U, BUT THEY ARE RATHER
_ LONG. IT IS SIMPLER TO USE RECURRENT RELATIONS TO EXPRESS
_ THE AVERAGE OF THE PRODUCT OF N VECTORS U THROUGH AVERAGES
_ OF THE PRODUCTS OF N-2 VECTORS U. FOR EXAMPLE, IN 3 DIMENSIONAL
_ SPACE:'
_ EPS 0 DIM(3)
_ E=(U.L*U.M*U.N*U.R*U.M1*U.N1-X*U.M*U.N*U.M1*U.N1*L.R)
_ F=(X/7*L.M*U.N*U.R*U.M1*U.N1
_   +X/7*L.N*U.M*U.R*U.M1*U.N1
_   +X/7*L.R*U.M*U.N*U.M1*U.N1
_   +X/7*L.M1*U.M*U.N*U.R*U.N1
_   +X/7*L.N1*U.M*U.N*U.R*U.M1)
_ E=SUB L,M,N,R,M1,N1:U.L*U.M*U.N*U.R*U.M1*U.N1=F:E
_ F=(X/5*L.M*U.N*U.R
_   +X/5*L.N*U.M*U.R
_   +X/5*L.R*U.M*U.N)
_ E=SUB L,M,N,R:U.L*U.M*U.N*U.R=F:E
```

```
_ F=(X/3*L.M)
_ E=SUB L,M:U.L*U.M=F:E
_ WRITE E
1/105*X^3*N1.M1*M.L*R.N+1/105*X^3*N1.M1*N.L*R.M-2/35*X^3*N1.
M1*R.L*N.M+1/105*X^3*L.M1*M.N1*R.N+1/105*X^3*L.M1*N.N1*R.M+
1/105*X^3*L.M1*R.N1*N.M+1/105*X^3*M.M1*L.N1*R.N-2/35*X^3*M.M1*
N.N1*R.L+1/105*X^3*M.M1*R.N1*N.L+1/105*X^3*N.M1*L.N1*R.M-2/
35*X^3*N.M1*M.N1*N.L+1/105*X^3*N.M1*R.N1*M.L+1/105*X^3*R.M1*
L.N1*N.M+1/105*X^3*R.M1*M.N1*N.L+1/105*X^3*R.M1*N.N1*M.L
_ COM 'THESE SUBSTITUTIONS SHOULD BE EXECUTED FROM HIGHER NUMBERS
_ OF VECTORS TO LOWER ONES.'
_
_ COM '------------EXAMPLE 6-------------------------------'
_ COM 'LET S BE 1/(U.U+1), AND SUPPOSE WE WANT TO CALCULATE
_ THE INTEGRAL OVER U OF A TENSOR CONTAINING U AND POWERS OF S.
_ LET US CONSIDER ONLY INTEGRALS WITH NO MORE THEN 4 VECTORS U,
_ WHICH ARE CONVERGENT AND CONTAIN NOT TOO LARGE POWERS OF S.
_ IT IS POSSIBLE TO USE THE TABLE OF SUCH INTEGRALS.IN THE 4
_ DIMENSIONAL SPACE THE PROGRAM MAY LOOK LIKE:
_ SCALAR  //PI;
_ E=((S+1)*S^7*U.M*U.N*U.L*U.R-(S+1)S^5*U.M*U.N*L.R
_  +(S+1)*S^3*M.N*L.R)
_ A=(INT) E=E*A
_ F=(PI^2/3360*M.N*L.R+PI^2/3360*M.L*N.R+PI^2/3360*M.R*N.L)
_ E=SUB L,M,N,R:INT*S^8*U.M*U.N*U.L*U.R=F:E
_ F=(PI^2/1440*M.N*L.R+PI^2/1440*M.L*N.R+PI^2/1440*M.R*N.L)
_ E=SUB L,M,N,R:INT*S^7*U.M*U.N*U.L*U.R=F:E
_ COM /INTEGRALS WITH 4 VECTORS U HAVE BEEN CONSIDERED/
_ F=(PI^2/120*M.N) E=SUB M,N:INT*S^6*U.M*U.N=F:E
_ F=(PI^2/24*M.N)  E=SUB M,N:INT*S^5*U.M*U.N=F:E
_ COM /INTEGRALS WITH 2 VECTORS U HAVE BEEN CONSIDERED/
_ A=(PI^2/6) E=SUB INT*S^4=A:E
_ A=(PI^2/2) E=SUB INT*S^3=A:E
_ COM /ALL INTEGRALS HAVE BEEN CONSIDERED/
_ COM SUCH TABLES OF INTEGRALS MAY BE LARGE; IT IS CONVENIENT TO
_ PLACE THEM INTO FILES.'
_
_ COM '----------- APPENDIX: ERRORS ------------------------'
_ COM 'THE FOLLOWING LISTING IS A PROTOCOLE OF DIALOGE WITH
_ DIRAC OF AN UNLUCKY USER MAKING ALMOST EVERY POSSIBLE ERROR.
_ AFTER DISCOVERING AN ERROR DIRAC SKIPS THE REST OF LINE
_ AND READS NEW COMMAND FROM THE NEXT LINE. BUT SOME COMMANDS
_ MAY BE RATHER LENGTHY. IT WOULD BE TIRESOME TO RETYPE THE
```



```
_ ENTIRE COMMAND. THEREFORE, IN SOME CASES (WHEN <DIALOG> IS ON)
_ DIRAC ASKS TO REPEAT ONLY THE PART OF THE COMMAND THAT CONTAINS
_ THE ERROR.'
_ COM 'SET LINE NUMBER: ' 1
_ /WRITE
        ^
_
ERROR AT LINE   2 IN COMMAND :!
_ ?WRITE
-WRITE
_  +WWRITE
          ^
_
ERROR AT LINE  4 IN FLAG : WWRITE
_ ++WRITE
         ^
ERROR AT LINE  4 IN FLAG : +
_ ELSE TEXT 'ELSE' ENDIF
                        ^
ERROR AT LINE  5 IN ELSE
_ ENDIF
        ^
ERROR AT LINE  6 IN ENDIF
_ ORDER!
        ^
ERROR AT LINE  7 IN ORDER: !
_ ORDER?
ORDER 4
_ A=3 DIM A
            ^
ERROR AT LINE  9 IN DIM : A
_ DIM(A)
         ^
ERROR AT LINE  10 IN FACTOR : A
REPEAT N*X^N...+...)
_ 0)
_ DIM 3
        ^
ERROR AT LINE  12 IN DIM : 3
_ DIM(3)
_ DIM?
_ EPS A
        ^
ERROR AT LINE  15 IN EPS : A
_ EPS(3)
```

```
          ^
ERROR AT LINE  16 IN EPS : (
_ EPS 33
          ^
ERROR AT LINE  17 IN EPS : 33
_ EPS 3
_ ORDER A
           ^
ERROR AT LINE  19 IN ORDER : A
_ ORDER 10
_ (UU.V=2*X)
             ^
ERROR AT LINE  21 IN FIRST VECTOR : UU
_ (U*V=2*X)
            ^
ERROR AT LINE  22 IN SCALAR PRODUCT : *
_ (U.M=2*X)
            ^
ERROR AT LINE 23 IN SECOND VECTOR : M
_ (U.V:2*X)
            ^
ERROR AT LINE 24 IN EQUALITY : :
_ (U.V:2*X)
_ (U.V?)
          ^
ERROR AT LINE 26 IN EQUALITY : ?
_ (U.V)
2*X
_ DELET A
           ^
ERROR AT LINE 28 IN COMMAND : DELET
_ DELETE A A=-2
               ^
ERROR AT LINE 29 IN ARGUMENT : 2
_ A=(-2) B=A*5
              ^
ERROR AT LINE 30 IN ARGUMENT : 5
_ B=5*A
        ^
ERROR AT LINE 31 IN COMMAND : *
_ COM 'DIRAC INTERPRETS THIS AS B=5 FOLLOWED BY THE WRONG
_ COMMAND *A'
_ C:=A*B
```



```
      ^
ERROR AT LINE 34 IN ASSIGNMENT : :
_ C=A-B
      ^
ERROR AT LINE 35 IN OPERATION : -
_ C=A/B
      ^
ERROR AT LINE 36 IN OPERATION : /
_ C=A^B
      ^
ERROR AT LINE 37 IN POWER : B
_ C=A*##B
       ^
ERROR AT LINE 38 IN ARGUMENT : #
_ C=A*#B
_ D=B+A
      ^
ERROR AT LINE 40 IN UNDEFINED ARGUMENT 2 : A
_ D=X
    ^
ERROR AT LINE 41 IN ARGUMENT : X
_ D=(X)
_ D=C+#D
       ^
ERROR AT LINE 43 IN REDEFINED RESULT : D
_ DELETE B DELETE C
_ COM 'AFTER AN ERROR IN A MONOMIAL, DIRAC ASKS TO REPEAT THIS
_ MONOMIAL (AND ALL WHAT FOLLOWS IT). IF THIS MESSAGE BEGINS
_ WITH "N*" THEN IT MEANS THAT NUMBERS MAY BE USED IN THE
_ MONOMIAL (ALSO IN DENOMINATORS). THE NEXT PART OF THE MESSAGE
_ IS USUALLY "S^N"; IT MEANS THAT SCALARS RAISED TO
_ INTEGER POWERS MAY BE USED. IF THIS PART OF THE MESSAGE
_ IS "S^-N" INSTEAD, THEN NEGATIVE POWERS ARE ALLOWED (THIS
_ IS THE CASE OF THE DIF FUNCTION). IN THE CASE OF TENSORS
_ THE NEXT PART OF THE MESSAGE IS "*V.I"; IT MEANS THAT VECTORS
_ WITH INDICES ARE ALLOWED. IF EPS>0 THEN [...] IS ALLOWED
_ TOO. THE NEXT PART OF THE REPEAT MESSAGE IS "...";
_ IT REMINDS THAT SUCH FACTORS MAY BE REPEATED AS MANY TIMES
_ AS NEEDED. THE MESSAGE IS USUALLY ENDED WITH "+...)"; IT
_ MEANS THAT FURTHER MONOMIALS MAY BE USED. THEY ARE SEPARATED
_ BY "+" OR "-", AND ENDED BY ")". IF THE MESSAGE IS
_ ENDED WITH ":" OR "=" INSTEAD, THEN IT MEANS THAT THIS
_ MONOMIAL IS USED IN DIF OR SUB FUNCTION AND SHOULD BE
```

```
_ FOLLOWED BY ":" OR "="  CORRESPONDINGLY."
_ A=(1++X
         ^
ERROR AT LINE 63 IN FACTOR : +
REPEAT N*X^N...+...)
_ X Y
     ^
ERROR AT LINE 64 IN MONOM : Y
REPEAT N*X^N...+...)
_ X:Y
     ^
ERROR AT LINE 65 IN MONOM : :
REPEAT N*X^N...+...)
_ X*A
     ^
ERROR AT LINE 66 IN FACTOR : A
REPEAT N*X^N...+...)
_ X^Y
     ^
ERROR AT LINE 67 IN POWER : Y
REPEAT N*X^N...+...)
_ 1/2*X/Y
         ^
ERROR AT LINE 68 IN DENOMINATOR : Y
REPEAT N*X^N...+...)
_ 1/0*X
       ^
ERROR AT LINE 69 IN DENOMINATOR : 0
REPEAT N*X^N...+...)
_ X^-3*Y^-4*Z^-5*X^+2*Y^3*Z^6+
                              ^
ERROR AT LINE 70 IN NEGATIVE POWER : X,Y
REPEAT N*X^N...+...)
_ 5/6*X*Y^2*7*X^-2*Z^3/4/5*X^+4)
A=(1+7/24*X^3*Y^2*Z^3)
_ COM !CORRECT!
_ A=DIF X*2:A
           ^
ERROR AT LINE  73 IN FACTOR : 2
REPEAT X^-N...:
_ X/2:A
       ^
ERROR AT LINE 74 IN MONOM : /
```



```
REPEAT X^-N...:
_ X^-2=A
        ^
ERROR AT LINE 75 IN DIF : =
_ X^-2:A
A=(1/2*X^2+7/480*X^5*Y^2*Z^3)
_ COM !CORRECT!
_ B=10 A=SUB X*2=B:A
                  ^
ERROR AT LINE 78 IN FACTOR : 2
REPEAT X^N...=
_ X/2=B:A
         ^
ERROR AT LINE 79 IN FACTOR : 2
REPEAT X^N...=
_ X^-2=B:A
          ^
ERROR AT LINE 80 IN NEGATIVE POWER : X
REPEAT X^N...=
_ X^2:A
      ^
ERROR AT LINE 81 IN MONOM : :
REPEAT X^N...=
_ X^2=B:A
A=(5+35/24*X*Y^2*Z^3)
_ COM !CORRECT!
_
_ E=4 COM 'IT IS LEGAL; E IS NOW A TENSOR WITHOUT INDICES.'
e=(4)
_ E=A*E
      ^
ERROR AT LINE 86 IN ARGUMENT :A
_ E=E*A COM 'THIS ORDERING IS LEGAL'
e=((20+35/6*X*Y^2*Z^3))
_ E=E^2
      ^
ERROR AT LINE 88 IN OPERATION : ^
_ E=F*E
      ^
ERROR AT LINE 89 IN UNDEFINED TENS ARGUMENT 1 : F
_ E=0
    ^
ERROR AT LINE 90 IN REDEFINED TENS RESULT : E
```

```
_ DELETE E
_ E=[U,V,M]
           ^
ERROR AT LINE 92 IN ARGUMENT : [
_ E=([U,V,M]) F=(U.N*V.L)
E=(1*[M,U,V])
F=(1*V.L*U.N)
_ F=F+E
       ^
ERROR AT LINE 94 IN INDEX MISUSE IN SUM : L,M,N
_ DELETE E DELETE F
_ E=(U.L*U.M*U.N+2*(1+X)*V.L*V.M*V.N
                                    ^
ERROR AT LINE 96 IN FACTOR : (
REPEAT N*S^N*V.I...+...)
_ (1+X)*2*V.L*V.M*V.L
                     ^
ERROR AT LINE 97 IN REPEATED INDEX : L
REPEAT N*S^N*V.I...+...)
_ V.L*V.R*V.N
             ^
ERROR AT LINE 98 IN PRESENT INDEX : R
REPEAT N*S^N*V.I...+...)
_ V.L*X+
        ^
ERROR AT LINE 99 IN ABSENT INDEX : M,N
REPEAT N*S^N*V.I...+...)
_ L.V*V.M*V*N
             ^
ERROR AT LINE 100 IN SCALAR PRODUCT : *
REPEAT N*S^N*V.I...+...)
_ V.L*V.M*V..N
              ^
ERROR AT LINE 101 IN SECOND VECTOR : .
REPEAT N*S^N*V.I...+...)
_ V.L*V.M*V.X
             ^
ERROR AT LINE 102 IN SECOND VECTOR : X
REPEAT N*S^N*V.I...+...)
_ V.L8V.M*V.U
             ^
ERROR AT LINE 103 IN V.U
REPEAT N*S^N*V.I...+...)
```



```
_[U.L]*V.M*V.N
              ^
ERROR AT LINE 104 IN EPS LIST : ]
REPEAT N*S^N*V.I...+...)
_ [U,V,L,M]*V.N
               ^
ERROR AT LINE 105 IN EPS LIST : ,
REPEAT N*S^N*V.I...+...)
_ [U,V,L]*V.M*[U,V,N]
                     ^
ERROR AT LINE 106 IN FACTOR : [
REPEAT N*S^N*V.I...+...)
_ [U,V;L]*V.M*V.N
                 ^
ERROR AT LINE 107 IN EPS LIST : ;
REPEAT N*S^N*V.I...+...)
_ [U,X,L]*V.M*V.N
                 ^
ERROR AT LINE 108 IN EPS VECTOR : X
REPEAT N*S^N*V.I...+...)
_ (1-X^2)*[U,V,L]/3*Y^2*M.V*V.N)
E=(1*U.L*U.M*U.N+(1/3*Y^2-1/3*X^2*Y^2)*[L,U,V]*V.M*V.N)
_ COM !CORRECT!
_ E=DIF U R:E
             ^
ERROR AT LINE 111 IN SCALAR PRODUCT : R
_ E=DIF U..R:E
              ^
ERROR AT LINE 112 IN SECOND VECTOR : .
_ E=DIF U.V:E
             ^
ERROR AT LINE 113 IN INDEX : U
_ E=DIF M.R:E
             ^
ERROR AT LINE 114 IN VECTOR : R
_ E=DIF U.R=E
             ^
ERROR AT LINE 115 IN DIF VECTOR : =
_ E=DIF U.R:E
E=(1*R.L*U.M*U.N+1*U.L*R.M*U.N+1*U.L*U.M*R.N+(1/3*Y^2-1/3*X^
2*Y^2)*[L,R,V]*V.M*V.N)
_ COM !CORRECT!
_ F=M.N
```

```
       ^
ERROR AT LINE 118 IN ARGUMENT : M
_ F=(M.N)
_ E=SUB M N:U.M*U.N=F:E
                   ^
ERROR AT LINE 120 IN INDEX LIST : N
_ E=SUB M,U:U.M*U.N=F:E
                   ^
ERROR AT LINE 121 IN INDEX : U
_ E=SUB M,,N:U.M*U.N=F:E
                    ^
ERROR AT LINE 122 IN INDEX : ,
_ E=SUB M,X:U.M*U.N=F:E
                   ^
ERROR AT LINE 123 IN INDEX : X
_ E=SUB L,R:U.M*U.N=F:E
                   ^
ERROR AT LINE 124 IN FORMAL INDEX ABSENT : L,R
_ E=SUB:U.L*U.R=F:E
                 ^
ERROR AT LINE 125 IN PRESENT INDEX : M,N
_ E=SUB:U.L*U.M*U.N*U.R=F:E
                     ^
ERROR AT LINE 126 IN ABSENT INDEX : L,R
_ E=SUB:U.M*U.N*X/2=F:E
                   ^
ERROR AT LINE 127 IN MONOM : /
REPEAT S^N*V.I...=
_ U.M*U.N*2*X=F:E
              ^
ERROR AT LINE 128 IN FACTOR : 2
REPEAT S^N*V.I...=
_ U.M*U.N*X-F:E
             ^
ERROR AT LINE 129 IN MONOM : -
REPEAT S^N*V.I...=
_ U.M*U.N*X=F:E
E=(1*R.L*U.M*U.N+1*U.L*R.M*U.N+1*U.L*U.M*R.N+(1/3*Y^2-1/3*X^
2*Y^2)*[L,R,V]*V.M*V.N)
_ COM !CORRECT!
_
_ <ABRA
       ^
```



```
ERROR AT LINE 133 IN FILE : ABRA.DIR
_ COM 'THIS FILE PROBABLY DOES NOT EXIST'
_ >AAAAAAAAAAAAAAAAAAAA.AAA
                          ^
ERROR AT LINE 134 IN FILEE : AAAAAAAAAAAAAAAAAAAA.AAA

_
_ COM 'AFTER AN ERROR IN A DECLARATION,DIRAC ASKS TO REPEAT
_ THE LAST DECLARATION ELEMENT. REPEAT MESSAGE BEGINS WITH
_ "/OLD/NEW"; IT REMINDS THE STRUCTURE OF A DECLARATION ELEMENT.
_ IN THE CASE OF SCALAR DECLARATION, ":ORDER " FOLLOWS.
_ THE MESSAGE ENDS WITH ",...;"; IT REMINDS THAT OTHER
_ DECLARATION ELEMENTS MAY FOLLOW. THEEY ARE SEPARATED BY ",",
_ AND TERMINATED BY ";" OR "?".
_ POLY?
POLY A,B,C,D,H,UNIT,D1,,,,,,,,,,,,,,,,,,,,,,,,;
_ POLY /.H/H1,
            ^
ERROR AT LINE 145 IN OLD NAME : .
REPEAL /OLD/NEW,...;
_ /HH/H1,
      ^
ERROR AT LINE 146 IN OLD NAME : HH
REPEAL /OLD/NEW,...;
_ /H H1,
      ^
ERROR AT LINE 147 IN DECLARATION ELEMENT : H1
REPEAL /OLD/NEW,...;
_ /H/A,
     ^
ERROR AT LINE 148 IN REDEFINED NAME : A
REPEAL /OLD/NEW,...;
_ C,/H/H1,
       ^
ERROR AT LINE 149 IN REDEFINED NAME : C
REPEAL /OLD/NEW,...;
_ E,
  ^
ERROR AT LINE 150 IN ILLEGAL NAME : E
REPEAL /OLD/NEW,...;
_ DIM,
     ^
ERROR AT LINE 151 IN ILLEGAL NAME ; DIM
REPEAL /OLD/NEW,...;
```

```
_ ?
POLY A,B,C,D,H,UNIT,D1,,,,,,,,,,,,,,,,,,,,,,,,;
_ SCALAR ?
SCALAR X:0,Y:0,Z:0,S:1,T:2,X1:0,INT:0,PI:0;
 _SCALAR X:-1;
         ^
ERROR AT LINE 154 IN ORDER : -
REPEAL /OLD/NEW:ORDER,...;
_ /PI/PI1,PI2;
         ^
ERROR AT LINE 155 IN LONG DECLARATION : ,
_ ?
SCALAR X:0,Y:0,Z:0,S:1,T:2,X1:0,INT:0,PI:0;
_
_ COM !GOOD BYE!
_
_ END
```

Acknowledgements

We are grateful to V.P. Gerdt for his support and interest in this work and for providing us the opportunity to use an IBM compatible personal computer. One of use (H.P.) is grateful to him also for hospitality during stay in Dubna, where this work was done.

References

/1/ Hearn, A. C. "REDUCE 3.2 User's Manual", Rand Corporation, 1983
/2/ Grosin, A. G. Preprint 83-117, Institute of Nuclear Research, Novosibirsk, 1983
/3/ Grosin, A. G. Proceedings of the International Conference on Computer Algebra and its Application in Theoretical Physics, JINR D11-85-791, Dubna, 1985
/4/ "TURBO PASCAL Owner's Handbook", Borland International, Scotts Valley, 1987

Received by Publishing Department
on July 14, 1989.

Грозин А.Г., Перлт Х. E11-89-537

Программа алгебраических вычислений DIRAC на персональных компьютерах типа IBM PC

Описывается версия DIRAC (2.2) для IBM-совместимых персональных компьютеров. Она разработана для аналитических вычислений с полиномами и тензорами. После короткого введения, касающегося реализации и использования системы на персональных компьютерах, дан пример программы. Он содержит детальное руководство пользователя по DIRAC (2.2) с некоторыми полезными приложениями.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1989

Grozin A.G., Perlt H. E11-89-537

The Algebraic Manipulation Program DIRAC on IBM Personal Computers

The version DIRAC (2.2) for IBM compatible personal computers is described. It is designed to manipulate algebraically with polynomials and tensors. After a short introduction concerning implementation and usage on personal computers an example program is given. It contains a detailed user's quide to DIRAC (2.2) and, additionally, some useful applications.

The investigation has been performed at the Laboratory of Computing Techniques and Automation, JINR.

Communication of the Joint Institute for Nuclear Research. Dubna 1989