```
COM DOPLN NAZEV PROGRAMU;
COM
AUTOR:B. LACKO
TYP; TYPOVY TEXT
FUNKCE;PROGRAM VYBIRA Z FILE UFDB URCITE VETY
         PODLE ZADANEHO KRITERIA A VYBRANE VETY
         PREVADI NA JINY FILE FPS,
         PROGRAMATOR SI MUSI;
             =DOPSAT SVUJ VLASTNI FILE
             =ZKOMPILOVAT TAKTO UPRAVENY PROGRAM
             NAVOD K POUZITI VIZ SOFTIN 11
;
ENV FILE
FDB(TAPE(A)),FPS(TAPE(B)),
PASKA (TAPEREADER,ALGOLFLEX),
POMTISK (LINEPRINTER);
                                             COM POPISY FILE FDB A FPS;
IOC SAME RECORD AREA (FDB,FPS);
DATA FILE
FD FDB LAB LABDB,100 REC DB BLQ 1023;
COM PRVNI REKORD FILU MUSI BYT NAZVAN DB;

COPY RECDB;
FD FPS LAB LABPS,0 REC RPS BLQ 1023;
1 RPS IS DB;
                                             COM KONEC POPISU;
FD POMTISK REC TEXT,TREL,TINT,
HLA;
1 HLA;
2 DEN INT < 2SR Z2,0,>;
2 MES<Z2,0,>;
2 ROK <4,0>;
2 NAD STR SIZ 40;
1 TEXT;
2 TXT STR SIZ 70;
1 TREL;
2 PTREL INT < 2SR ZZ=3.0>;
2 STREL;
3 < 2SR  ZZ3,0>;
3 < 2SR ZZ3,0>;
3 < 2SR ZZ3,0>;
3 < 2SR ZZ=6,0>;
3 REAL < 2SR  ZZ=6,2 2SR >;
3 STR SIZ 30;
1 TINT;
2 HINT INT < 4SR Z=3,0>;
FD PASKA REC RECDP;
1 RECDP;
2 DPPR STR SIZ 500 DER </> NON;
WOS
1 POLE REL;
2 QPR OCC 50;
3 QT INT;
3 QI1;
3 QR;
3 HO INT;
3 HR REAL;
3 HA STR SIZ 30;
77 POMO STR SIZ 30;
77 IPPR STR SIZ 501 VAL <501SR >;
77 PSIP STR SIZ 4;
77 LABDB STR SIZ 21 VAL <DB|>;
77 LABPS STR SIZ 21 VAL<PS>;
77 A INT VAL 6;
77 B INT VAL 2;
```

```
77 SRR STR SIZ 30;
SQN
                                                CQN KONSTANTY PRO TEST INDEXU;
1 VAL 0;
2 TCI INT QCC 1;
77 POCETTCI INT VAL 1; CQN TEST TYPU C;
1 VAL 0;
2 TCR INT QCC 1;
77 POCETTCR INT VAL 1; CQN TEST TYPU R;
1 VAL 0;
2 TCA INT QCC 1;
77 POCET TCA INT VAL1; CQN TEST TYPU A;
                                                CQN KONEC POPISU KONSTANT;
1 VAL
<CH SR 1 2SR PR 2SR CHYBNY SR INDEX SR UDAJE  SR R>,
<CH SR 2 2SR PR 2SR
PODTRHNUTO SR NEPRIPUSTNE SR RELACNI SR ZNAMENKO>,
< CH SR 3 2SR PR 2SR PRESKRTNUTO SR NEPRIPUSTNE SR RELACNI SR ZNAMENKO>,
< CH SR 4 2SR PR 2SR NEPRIPUSTNE SR RELACNI SR ZNAMENKO>,
< CH SR 5 2SR POC. SR ZNAKU SR V SR PRAVIDLU >500, SR NEBO SR RELACI >50
<CH SR 6 2SR PR 2SR CHYBNY SR INDEX SR UDAJE SR C>,
<CH SR 7 2SR PR 2SR CHYBNY SR INDEX SR UDAJE SR A >,
< CH SR 8 2SR PR 2SR POCET SR ZNAKU SR V SR RETEZU SR RELACE SR A SR JE
< CH SR 9 2SR PR 2SR NEZNAME SR ZNAMENKO SR NEBO SR TYP SR RELACE >,
< CH 10 2SR PR 2SR SPATNY SR INDEX SR V SR OZNACENI SR UDAJE>,
< CH 11 2SR PR 2SR CHYBNE SR ZAVORKOVANI SR NEBO SR UMISTENI SR ZNAMENEK
< CH 12 2SR PR 2SR CHYBNE SR NAPSANE SR PRAVIDLO >,
<CH13 6SR PRVNI SR RETEZ SR NENI SR PR SR ANI SR BAND>,
< CH 14 2SR PR 2SR ZA SR PR SR NENI SR RETEZ - PRAVIDLO>,
< CH 15 6SR PRVNI SR IP SR NENI SR RETEZ>,
< CH 16 2SR BAND 2SR CHYBI SR IP SR BAND>,
< CH 17 2SR BAND 2SR CHYBNE SR ZADANI SR MECH,SR I>,
< CH [illegible] CISLO SR MECH,SR I >7 SR NEBO SR I<0>,
< CH 19 2SR BAND 2SR CHYBI SR NAVESTI SR PRO SR MECH.SR I >,
< CH 20 2SR BAND 2SR CHYBNE SR ZADANI SR MECH.SR J >,
< CH 21 2SR BAND 2SR CISLO SR MECH,SR J SR >7 SR NEBO SR I LO >,
< CH 22 2SR BAND 2SR CHYBNE SR NAVESTI SR PRO SR MECH.SR J >,
<CH23 2SR BAND 2SR MALO SR IP SR V SR TETO SR CASTI>,
< CH24 2SR PR 2SR CHYBNE SR ZADANE SR PRAVIDLO SR Z SR RL >,
<CH25 2SR PR 2SR CHYBNY SR POCET SR ZAVOREK>,
<CH26 2SR PR 2SR CHYBNY SR POCET SR OPERATORU>,
< CH27 2SR PR 2SR SPATNY SR TYP SR OPERATORU >,
<CH28 2SR PR 2SR CHYBNE SR UMISTENY SR OPERATOR SR 32 N 32>,
<CH29>,
<CH30>;
2 TCH STR SIZ 50 QCC 30;
SRQ
INT TR,PI IP,II,
ZAVOR,OPERATOR,OPERAND,JI;
BOOL KF,VSE,ABORT,
CLEAR;
INT ARR X,PQ[1:501];
INT PRO HII (GR,P);
VALP; INT P;
GRQ GR;
ING DREFI;
REAL PRO HIR(GR,P);
VAL P; INT P;
GRQ GR;
ING DREFR;
PRO HIS(GR,P,S);
VAL P; INT P;
GRQ GR;
STR S;
ING DREFS;
PRO VCH(A,B);
```

```
132  VAL A;
133  INT A;
134  LABEL B;
135  BEGIN
136  ABORT:=TRUE;
137  ADVANCELINE(1);
138  ADVANCELINE(-2);
139  DATINT(DEN,MES,ROK);
140  MOVESTRING(<
141  1$$R EXTRAKCE DBI 2$$R CHYBY>,
142  NAD,-1);
143  WRITELINE(HLA,-2);
144  COMBSTRING(<$$R >,TCH[A],TXT);
145  WRITELINE(TEXT,-2);
146  GOTO B;
147  END;
148  PROC CHYB(A,B);
149  INT A,B;
150  BEGIN
151  IF A=81 THEN VCH(10,ZK);
152  END;
153  PROC TRANSF;
154  BEGIN
155  INT I,Z,K,
156  R,PX;
157  EXPLODE(IPPR,PQ,1,501);
158  FOR I:= 1 STEP 1 UNT 501 DO
159  X[I]:=0;
160  I:=PX:=1;
161  R:=0;
162  X[PX]:=37;
163  ZAC;
164  IF PQ[I]=16 THEN GOTO KTRA;
165  IF PQ[I]=65 OR PQ[I]=82 OR PQ[I]=67
166  OR PQ[I]=97 OR PQ[I]=114 OR PQ[I]=99
167  THEN
168  BEGIN  COM  DESIFRACE RELACI;
169  R:=R+1;
170  PX:=PX+1;
171  IF R>50 OR (I+5)>501 THEN VCH(5,ZK);
172  X[PX]:=R*(-1);
173  IF NOT (PQ[I]=67 OR PQ[I]=99) THEN GOT DESR;
174  OT[R]:=67;
175  I:=I+1; Z:=I;
176  ZN:IF PQ[I]>47  AND PQ[I]<58 THEN BEG I:=I+1; GOTO ZN END;
177  IMPLODE(POMO,PQ,Z,I-Z);
178  K:=STRINGTOREAL(POMO);
179  IF K<1 THE VCH(1,ZK);
180  OD1[R]:=K;
181  IF PQ[I]=17 THEN
182  BEG
183  I:=I+1;
184  IF PQ[I]=41 THEN OR[R]:=1 ELSE
185  IF PQ[I]=43 THEN OR[R]:=2 ELSE VCH(2,ZK);
186  END
187  ELSE
188  IF PQ[I]=18 THEN
189  BEG
190  I:=I+1;
191  IF PQ[I]#42 THEN VCH(3,ZK);
192  OR[R]:=3;
193  END
194  ELSE
195  IF PQ[I]= 41 THEN OR[R]:=4
196  ELSE IF PQ[I]=42 THEN OR[R]:=5
197  ELSE IF PQ[I]=43 THEN OR[R]:=6
```

```
198  ELSE
199  VCH(4,ZK);
200  I:= I+1; Z:=I;
201  ZN2:
202  IF PQ[I]=26 OR PQ[I]=27
203  OR PQ[I]=39 OR PQ[I]=40
204  OR PQ[I]=38 OR PQ[I]=37
205  OR PQ [I] = 17
206  OR PQ[I]=0 OR PQ[I]=16
207  THEN
208  BEG
209  IMPLODE(POMQ,PQ,Z,I-Z);
210  HC[R]:=STRINGTOREAL(POMQ);
211  GOTO ZAC;
212  END
213  ELSE
214  BEG I:=I+1;
215  GOTO ZN2;
216  END;
217  DESR:
218  IF GOT (PQ[I]=82 OR PQ[I]=114) THE GOT DESA;
219  QT[R]:=82;
220  I:=I+1;Z:=I;
221  ZN3:
222  IF PQ[I]>47 AND PQ[I]<58 THEN
223  BEG I:=I+1;GOTO ZN3 END;
224  IMPLODE(POMQ,PQ,Z,I-Z);
225  K:=STRINGTOREAL (POMQ);
226  IF K<1 THE VCH(6,ZK);
227  C1[R]:=K;
228  IF PQ[I]=17 THEN
229  BEG
230  I:=I+1;
231  IF PQ[I]=41 THEN  QR[R]:=1 ELSE
232  IF PQ[I]=43 THEN QR[R]:=2 ELSE VCH(2,ZK);
233  END
234  ELSE
235  IF PQ[I]=18 THEN
236  BEG
237  I:=I+1;
238  IF PQ[I]#42 THEN VCH(3,ZK);
239  QR[R]:=3;
240  END
241  ELSE
242  IF PQ[I]=41 THEN QR[R]:=4
243  ELSE IF PQ[I]=42 THEN QR[R]:=5
244  ELSE IF PQ[I]=43 THEN QR[R]:=6
245  ELSE
246  VCH(4,ZK);
247  I:=I+1;Z:=I;
248  ZN4:
249  IF PQ[I]=26 OR PQ[I]=27
250  OR PQ[I]=38 OR PQ[I]=37
251  OR PQ[I]=39 OR PQ[I]= 40
252  OR PQ [I] = 17
253  OR PQ[I]=0 OR PQ[I]=16
254  THEN BEG
255  IMPLODE(POMQ,PQ,Z,I-Z);
256  HR [R]:= STRINGTOREAL (POMQ);
257  GOTO ZAC;
258  END
259  ELSE
260  BEG
261  I:=I+1;
262  GOTO ZN4;
263  END;
```

```
264  DESA;
265  OT[R]:=65;
266  I:=I+1;Z:=I;
267  ZN5:
268  IF PQ[I]>47 AND PQ[I]<58 THEN
269  BEG I:=I+1; GOTO ZN5 END;
270  IMPLODE(POMO,PQ,Z,I-Z);
271  K:=STRINGTOREAL(POMO);
272  IF K<1 THE VCH(7,ZK);
273  Q1[R]:=K;
274  IF PQ[I]=17 THEN
275  BEG
276  I:=I+1;
277  IF PQ[I]=41 THEN OR[R]:=1 ELSE
278  IF PQ[I]=43 THEN OR[R]:=2 ELSE VCH(2,ZK);
279  END
280  ELSE
281  IF PQ[I]=18 THEN
282  BEG
283  I:=I+1;
284  IF PQ[I]#42 THEN VCH(3,ZK);
285  OR[R]:=3;
286  END
287  ELSE
288  IF PQ[I]=41 THEN OR[R]:=4
289  ELSE IF PQ[I]=42 THEN OR[R]:=5
290  ELSE IF PQ[I]=43 THEN OR[R]:=6
291  ELSE
292  VCH(4,ZK);
293  I:=I+1; Z:=I;
294  ZN6:
295  IF PQ[I]=26 OR PQ[I]=27
296  OR PQ[I]=38 OR PQ[I]=37
297  OR PQ[I]=39 OR PQ[I]= 40
298  OR PQ [I] =17
299  OR PQ[I]=0 OR PQ[I]=16
300  THEN
301  BEG
302  IF (I-Z)>30 THEN VCH (8,ZK);
303  IMPLODE(POMO,PQ,Z,I-Z);
304  MOVESTRING (POMO,HA[R],-1);
305  GOTO ZAC;
306  END
307  ELSE
308  BEG
309  I:=I+1;
310  GOTO ZN6;
311  END;
312  END      KONEC DESIFRACE RELACI;
313  IF PO [I] = 17 THEN
314  I:=I+1; COM PRESKOCENI PODTRZENE OPERACE;
315  IF PQ[I]=26 OR PQ[I]=27
316  OR PQ[I]=38 OR PQ[I]=37
317  OR PQ[I]=39 OR PQ[I]=40
318  OR PQ[I]=78 OR PQ[I]= 69 OR PQ[I]=73
319  THEN
320  BEG
321  PX:=PX+1;
322  X[PX]:=PQ[I];
323  IF X[PX]=39 THEN X[PX]:=37; COM MODIFIKACE INA()
324  IF X[PX]=40 THEN X[PX]:=38; COM MODIFIKACE ] NA );
325  I:=I+1;
326  GOTO ZAC;
327  END
328  ELSE
329  VCH(9,ZK);
```

```
330  KTRA;
331  PX:=PX+1;
332  X[PX]:=38;
333  PX:=PX+1;
334  X[PX]:=16;
335  COM FORMALNI KONTROLY PRAVIDLA;
336  ZAVOR:=0;
337  J:=0;
338  FOR J:=J+1 WHILE X[J]#16 DO
339  IF X [J] = 37 THEN ZAVOR:=ZAVOR+1
340  ELSE
341  IF X[J]=38 THEN
342  BEG
343  ZAVOR:=ZAVOR-1;
344  IF ZAVOR<0 THE VCH(25,ZK);
345  END;
346  IF ZAVOR>0 THEN VCH (25,ZK);
347  OPERATOR:=OPERAND:=J:=0;
348  FOR J:=J+1 WHILE X[J]#16 DO
349  IF
350  X[J]=26 OR X[J]=27
351  OR X[J]=69 OR X[J]=73
352  THEN
353  OPERATOR:=OPERATOR+1
354  ELSE IF X[J]<0 THEN OPERAND:=OPERAND+1;
355  IF (OPERATOR+1)#OPERAND THE VCH(26,ZK);
356  J:=0;
357  FOR J:=J+1 WHILE X[J]#16 DO
358  IF X[J]=78 THEN
359  BEG
360  IF NOT (X[J+1]=78 OR X[J+1]=37 OR X[J+1]<0) THEN
361  VCH (28,ZK);
362  END;
363                                        COM KONTROLA INDEXU;
364  IF TCI [1]=0 AND TCR [1] = 0 AND TCA [1] = 0 THEN
365  GOTO NEKONTR;
366  I:=0;
367  FOR I:=I+1 WHILE OT [I] # 0 DO
368  BEG
369  IF OT [I] = 67 THEN
370  BEG
371  IF TCI [I] = 0 THEN GOTO KOTIN;
372  FOR J:=1 STEP 1 UNT POCET TCI DO
373  IF Q I 1 [I] = TCI [J] THEN GOTO KOTIN;
374  VCH (6,KOTIN);
375  END
376  ELSE
377  IF OT [I] = 65 THEN
378  BEG
379  IF TCA [1] = 0 THEN GOTO KOTIN;
380  FOR J:=1 STEP 1 UNT POCET TCA DO
381  IF Q I 1 [I] = TCA [J] THEN GOTO KOTIN;
382  VCH(7,KOTIN);
383  END
384  ELSE
385  BEG
386  IF TCR [1] = 0 THEN GOTO KOTIN;
387  FOR J:=1 STEP 1 UNT POCET TCR DO
388  IF Q I 1 [I] = TCR[J] THEN GOTO KOTIN;
389  VCH(1,KOTIN);
390  END;
391  KOTIN:
392  END;
393  IF ABORT THEN GOTO ZK;
394  NEKONTR:
395  END KONEC TRANSF;
```

```
396  $BOOL PRO PRAVIDLO;
397  $BEG
398  INT Z,V,Y,
399  I;
400  INT ARRAY SZ[1:50];
401  $BOOL ARRAY SV[1:50];
402  $BOOL PRO EQA(G);
403  VAL G;INT G;
404  $BEG
405  INT P;
406  $VI SW:=A,B,C,D,E,F;
407  HIS(DB,Q1[G],SRR);
408  $GOTO SW[QR[G]];
409  A:
410  STRINGCOMP (SRR,HA[G],P);
411  EQA:=P>0;
412  $GOTO VEN;
413  B:
414  STRINGCOMP (SRR,HA[G],P);
415  EQA:=P<0;
416  $GOTO VEN;
417  C:
418  EQA:= NOT STRINGEQUAL (SRR,HA[G]);
419  $GOTO VEN;
420  D:
421  STRINGCOMP (SRR,HA[G],P)
422  EQA:=P=1;
423  $GOTO VEN;
424  E:
425  EQA:= STRINGEQUAL (SRR,HA[G]))
426  $GOTO VEN;
427  F:
428  STRINGCOMP(SRR,HA[G],P);
429  EQA:=P=-1;
430  VEN:
431  $END;
432  $BOOL PRO EQC(G);
433  VAL G;INT G;
434  $BEG
435  INT P;
436  $VI SW:=A,B,C,D,E,F;
437  P:=HII(DB,Q1[G]);
438  $GOTO SW[QR[G]];
439  A:
440  EQC:=P>HC[G];
441  $GOTO KE;
442  B:
443  EQC:=P< HC[G];
444  $GOTO KE;
445  C:
446  EQC:=P# HC[G];
447  $GOTO KE;
448  D:
449  EQC:=P>HC[G];
450  $GOTO KE;
451  E:
452  EQC:=P=HC[G];
453  $GOTO KE;
454  F:
455  EQC:=P< HC[G];
456  KE:
457  $END;
458  $BOOL PRO EQR(G);
459  VAL G;INT G;
460  $BEG
461  $REAL P;
```

```
463   P:=H[R (DO,O:I[G]);
464   GOTO SW[OR[G]];
465   A:
466   EQR:=P> HR[G];
467   GOTO KE;
468   B:
469   EQR:=P<HR[G];
470   GOTO KE;
471   C:
472   EQR:=P#HR[G];
473   GOTO KE;
474   D:
475   EQR:=P>HR[G];
476   GOTO KE;
477   E:
478   EQR:=P=HR[G];
479   GOTO KE;
480   F:
481   EQR:=P<HR[G];
482   KE:END;
483   WT BEG PRIOR(OP);
484   VAL OP;
485   WT OP;
486   PRIOR:=IF OP=26 THEN 3 ELSE
487   IF OP=27 THEN 4 ELSE
488   IF OP=78 THEN 5 ELSE
489   IF OP=69 THEN 1 ELSE
490   IF OP=73 THEN 2 ELSE
491   0;
492   BOO BEG READOP(P);
493   VAL P; WT P;
494   READOP:= IF P=26 THEN (SV[V] OR SV[V-1]) ELSE
495   IF P=27 THEN (SV[V] AND SV[V-1]) ELSE
496   IF P=78 THEN NOT SV[V]
497   ELSE
498   IF P=69 THEN (SV[V] EQV SV[V-1]) ELSE
499   (SV[V-1] IMP SV[V]);
500   COM INTERPRET PRO VYHODNOCENI PRAVIDLA;
501   I:=V:=Z:=0;
502   ALFA:
503   I:=I+1;
504   Y:=X[I];
505   IF Y=16 THEN GOTO OMEGA;
506   IF Y<0 THEN
507   BEG
508   I:=ABS(Y);
509   SV[V+1]:=
510   IF QT[I]=65 THEN EQA(I) ELSE
511   IF QT[I]= 67 THEN EQC(I) ELSE
512   EQR(I);
513   V:=V+1;
514   GOTO ALFA;
515   END;
516   IF
517   Y # 37 AND Y # 38
518   AND Y # 26 AND Y # 27
519   AND
520   Y#78 AND Y#69 AND Y#73
521   AND Y # 16
522   THEN VCH (27,ZF);
523   IF Z=0 THEN BEG
524   Z:=1;
525   SZ[1]:=Y;
526   GOTO ALFA;
527   END;
```

```
528  LAB3;
529  IF Y=38 AND SZ[Z]=37
530  THEN
531  BEG
532  Z:=Z-1;
533  GOTO ALFA;
534  END;
535  IF
536  Y=37 OR SZ[Z]=37
537  THEN
538  BEG
539  SZ[Z+1]:=Y;
540  Z:=Z+1;
541  GOTO ALFA;
542  END;
543  IF
544  Y=38 AND SZ[Z] # 37 THEN
545  BEG
546  IF SZ[Z]=78 THEN
547  SV[V]:=READP(SZ[Z])
548  ELSE
549  BEG
550  SV[V-1]:= READP(SZ[Z]);
551  V:=V-1;
552  END;
553  Z:=Z-1;
554  GOTO LAB3;
555  END;
556  IF
557  PRIOR (Y) < PRIOR (SZ[Z]) THEN
558  BEG
559  IF SZ[Z]=78 THEN
560  SV[V]:=READP(SZ[Z])
561  ELSE
562  BEG
563  SV[V-1]:= READP(SZ[Z]);
564  V:=V-1;
565  END;
566  Z:=Z-1;
567  GOTO LAB3;
568  END;
569  IF PRIOR (Y)> PRIOR(SZ[Z]) THEN
570  BEG
571  SZ[Z+1]:=Y;
572  Z:=Z+1;
573  GOTO ALFA;
574  END;
575  IF PRIOR(Y)=PRIOR(SZ[Z])THEN
576  BEG
577  IF SZ[Z]=78 THEN
578  SV[V]:=READP(SZ[Z])
579  ELSE
580  BEG
581  SV[V-1]:=READP(SZ[Z]);
582  V:=V-1;
583  END;
584  SZ[Z]:=Y;
585  GOTO ALFA;
586  END
587  ELSE
588  VCH(11,2F);
589  OMEGA:
590  IF V # 1 THEN VCH (12,2F);
591  PRAVIDLO:=SV[1];
592  END PRAVIDLO PROCEDURA KONEC;
593  BEG TRASTISK;
```

```
BEG
ADVANCELINE (-2);
MOVESTRING (< 20$R
TRASTISK >, TXT,-1);
WRITELINE (TEXT,-2);
MOVESTRING ( < 20$R TRANSF >, TXT, - 1);
WRITELINE (TEXT,-2);
MOVESTRING(IPPR,TXT,70);
WRITELINE (TEXT,-2);
FOR I:=1 STEP 1 UNT 50 DO
BEG
TREL:=(-1)*I;
MOVEGROUP(OPR[I],TREL);
WRITELINE (TREL,-1);
END;
ADVANCELINE(1);
ADVANCELINE(-2);
MOVESTRING(<10$R POLE SR PO>,TXT,-1);
WRITELINE(TEXT,-2);
FOR I:=1 STEP 1 UNT 65 DO
BEG
TINT:=PO[I];
WRITELINE(TINT,-1);
END;
ADVANCELINE(1);
ADVANCELINE(-2);
MOVESTRING(<10$R POLE SR X>, TXT,-1);
WRITELINE(TEXT,-2);
FOR I:=1 STEP 1 UNT 65 DO
BEG
TINT:=X[I];
WRITELINE(TINT,-1);
END;
END KONEC POMOCNEHO TISKU;
COM ZACATEK VLASTNI VYPOCTOVE CASTI PROGRAMU;
TYPETEXT(< CR SELECTION SR BEG >);
VSE:= FALSE;
ABORT:= FALSE;
ERROR (CHYB);
OPENOUTPUT(POMTISK);
OPENIP;
NEXTIP (NIC);
IF NOT SIP (PSIP) THEN VCH(15,ZK);
IF NOT STRINGEQUAL(<PR>,PSIP) THEN
BEG IF STRINGEQUAL (<BAND>,PSIP) THEN
BEG VSE:= TRUE;
GOTO DDIP;
END
ELSE
VCH(13,ZK);
END;
NEXTIP (CHPR);
IF NOT SIP (IPPR) THEN VCH (14,ZK);
IF STRINGEQUAL(<VIZRL>,IPPR)
OR STRINGEQUAL(<VIZRL>,IPPR)
OR STRINGEQUAL(<VIZ RL>,IPPR)
THEN
BEG
TYPETEXT (<CR ** RL:=PRAVIDLO?>);
OPENINPUT(PASKA);
READFILE(RECDP,TR,KF);
IF KF THEN VCH(24,ZK);
IF NOT STRINGEQUAL(<PRAVIDLO>,DPPR) THEN VCH(24,ZK);
READFILE(RECDP,TR,KF);
IF KF THEN VCH(24,ZK);
MOVESTRING(DPPR,IPPR,-1);
```

```
660  END;
661  NEXTIP (DAL);
662  ODIP;
663  IF NOT SIP(PSIP) THEN VCH(16,ZK);
664  IF NOT STRINGEQUAL (<BAND>,PSIP) THEN VCH(16,ZK);
665  NEXTIP (CHLB);
666  IF NOT IIP (A) THEN VCH(17,ZK);
667  IF A=0 OR ABS (A)>7 THEN VCH(18,ZK);
668  NEXTIP (CHLB1);
669  IF NOT SIP (LABDB) THEN VCH(19,ZK);
670  NEXTIP (CHLB2);
671  IF NOT IIP(B) THEN VCH(20,ZK);
672  IF ABS(B)>7 OR B=0 THEN VCH(21,ZK);
673  NEXTIP (CHLB3);
674  IF NOT SIP (LABPS) THEN VCH(22,ZK);
675  DAL;
676  ZEROGROUP (POLE REL);
677  TRANSF;
678  A:=ABS(A);
679  OPENINPUT (FDB);
680  IF B<0 THEN BEG
681  B:=ABS(B);REOPEN (FPS) END
682  ELSE OPENOUTPUT (FPS);
683                                          COM ZACATEK VYKONNE CASTI
684                                             PRAVIDLO MA HODNOTU
685                                               =TRUE PODMINKA VYBERU SPLNENA
686                                               =FALSE PODMINKA NESPLNENA
687                                             VSE=TRUE PODMINKA CHYBI
688                                          ;
689  OPAK: READFILE (FDB,TR,KF);
690  IF KF THEN GOTO ZF;
691  IF PRAVIDLO THEN MOVEFILE (FDB,FPS);
692  GOTO OPAK;
693                                          COM KONEC VYKONNE CASTI;
694  NIC;
695  VSE:=TRUE;
696  GOTO DAL;
697  CHPR:
698  VCH(23,ZK);
699  CHLB:
700  VCH(23,ZK);
701  CHLB1:
702  VCH(23,ZK);
703  CHLB2:
704  VCH(23,ZK);
705  CHLB3:
706  VCH(23,ZK);
707  ZF:CLOSE (FPS,TRUE);
708  CLOSE (FDB,TRUE);
709  ZK:
710  IF ABORT  THE TRASTISK;
711  IF ABORT THEN
712  TYPETEXT(< CO CHYBY SE VIZ SE TISKARNA >);
713  TYPETEXT(< CO SELECTION SE END>);
714  END;
```