```
env body
  ;reg declarations
    arr char \q,\a,\z,\w,\s,\x,\e,\d,\c,\r,\f,\v,\t,\g,\b,\y,\h,\n,\u,\j,\m,\i
    ,\k,\o,\l,\p
    arr CHAR \Q,\A,\Z,\W,\S,\X,\E,\D,\C,\R,\F,\V,\T,\G,\B,\Y,\H,\N,\U,\J,\M,\I
    ,\K,\O,\L,\P
    arr digits \0,\1,\2,\3,\4,\5,\6,\7,\8,\9
    arr downshift \_
    str noone \n\o\o\n\e
    arr null noone
    arrrem null noone;getting empty arr
    arr space \ ;
    arr tab \  ;here it's TAB!
    arr newline \
;end of new line
    arr begincomment \/\*
    arr endcomment \*\/
    arr include \#\i\n\c\l\u\d\e
    arr incopen \<
    arr incclose \>
    arr types \i\n\t, \l\o\n\g\ \l\o\n\g, \c\h\a\r, \f\l\o\a\t, \d\o\u\b\l\e,
    \s\h\o\r\t
    bor T.c \i\n\t, \l\o\n\g\ \l\o\n\g, \c\h\a\r, \f\l\o\a\t, \d\o\u\b\l\e,
    \s\h\o\r\t
    bor T.pas \L\o\n\g\I\n\t, \I\n\t\6\4, \S\h\o\r\t\I\n\t, \S\i\n\g\l\e,
    \C\o\m\p, \S\m\a\l\l\I\n\t
    arr arrayscopeopen \[
    arr arrayscopeclose \]
    arr blockscopeopen \{
    arr blockscopeclose \}
    arr scopeopen \(
    arr scopeclose \)
    arr comma \,
    obj tabul 1 space 1 tab 1 newline 1 null
    obj spaces 100 tabul
    obj fsym 1 char 1 CHAR 1 downshift
    obj sym 1 fsym 1 digits
    obj name 1 fsym, 100 sym 1 null
    obj funcname 1 name, 1 spaces, 1 scopeopen
    obj arrayname 1 name, 1 spaces, 1 arrayscopeopen
  ;endreg
  opnrdf file \p\r\o\g\A\.\c
  opnwrf out \p\r\o\g\A\.\p\a\s
  met cycle
  ;reg bodycycle
    ;reg scan spaces, tabs and newlines to trash
    scnfltfil trash spaces file
      nop
    ;endreg
    obj ocomments 1 begincomment
    scnfltfil comments ocomments file
      jmp next1
    ;reg comments
      psh file
      obj endcomm 1 endcomment
      psh endcomm
      psh out
```

```
      run Comments
      jmp cycle
    ;endreg
    met next1
    obj oinclude 1 include, 1 spaces, 1 incopen
    scnfltfil skip oinclude file
      jmp next2
    ;reg include
      psh file
      obj endcomm 1 incclose
      psh endcomm
      psh out
      run Include
      jmp cycle
    ;endreg
    met next2
    obj otype 1 spaces, 1 types, 1 spaces
    scnfltfil type otype file
      jmp exit1
    ;reg type
      scnfltfil function funcname file
        nop
      psh out
      psh file
      psh type
      psh function
      psh spaces
      psh T.pas
      psh T.c
      psh arrayname
      psh name
      psh types
      run Function
      jmp cycle
    ;endreg
  ;endreg
  met exit1
  cls file
  cls out
  end
endenv
env Comments
  str newline \
;
  pop out
  pop oend.
  pop file
  str S \(\*
  str F \*\)
  wrt out S
  met met1
  scnfltfil end. oend. file
  jmp readchar
  wrt out F
  wrt out newline
  end
  met readchar
```

```
  scndef stri 1 file
  wrt out stri
  jmp met1
endenv
env Include
  pop out
  pop oend.
  pop file
  met met1
  scnfltfil end. oend. file
  jmp readchar
  end
  met readchar
  scndef stri 1 file
  jmp met1
endenv
env Function
  ;reg pops
    pop types
    pop name
    pop arrayname
    pop Tc
    pop Tpas
    pop spaces
    pop function
    pop type
    pop file
    pop out
  ;endreg
  ;reg pushs
    psh file
    psh name
    psh arrayname
    psh spaces
    psh types
    psh Tc
    psh Tpas
  ;endreg
  run ReadVarInArgs
  pop arraynames
  pop stringnames
  str string \f\u\n\c\t\i\o\n\
  wrt out string
  ;reg write header
    prt string function 0
    wrt out string
    str string \(
    wrt out string
    ;reg write vars
      vctgsz stringnames size
      mth x 0
      jmp start1
      met cycle1
      ;reg cycle1
        tsteql x size
          jmp start1
        str string \;\
```

```
        wrt out string
        met start1
        tsteql size x
          jmp exit1
        mth n size 1 -
        mth size n
        vctget stringnames string n
        wrt out string
        jmp cycle1
      ;endreg
      met exit1
      vctgsz arraynames size
      jmp start2
      met cycle2
      ;reg cycle2
        tsteql x size
          jmp start2
        str string \;\
        wrt out string
        met start2
        tsteql size x
          jmp exit2
        mth n size 1 -
        vctget arraynames string n
        wrt out string
        jmp cycle2
      ;endreg
      met exit2
      str string \)\ \:\ ;
      wrt out string
      prt string type 1
      borgtn Tc num string
      borgts Tpas string num
      wrt out string
      str string \;\
;
      wrt out string
    ;endreg
  ;endreg
  ;mne vsyo nadoelo.
  ;budu ostavlyat' lish' zagolovji funkzij!!!
  str noone \n\o\o\n\e
  arr null noone
  arrrem null noone;getting empty arr
  arr space \ ;
  arr tab \  ;here it's TAB!
  arr newline \
;end of new line
  arr blockscopeopen \{
  arr blockscopeclose \}
  obj tabul 1 space 1 tab 1 newline 1 null
  obj spaces 100 tabul
  obj openscope 1 spaces 1 null, 1 blockscopeopen
  obj closescope 1 spaces 1 null, 1 blockscopeclose
  mth i 0
  mth zero 0
  met cycle01
```

```
  tsteql i zero
    jmp readopen01
  jmp exit01
  met readopen01
  scnfltfil o openscope file
    jmp next01
  mth x i 1 +
  mth i x
  jmp exit01
  met next01
  nxt file 1
  jmp cycle01
  met exit01
  met cycle02
  tsteql i zero
    jmp exit02
  scnfltfil o openscope file
    jmp skip021
  mth x i 1 +
  mth i x
  met skip021
  scnfltfil o closescope file
    jmp skip022
  mth x i 1 -
  mth i x
  jmp cycle02
  met skip022
  nxt file 1
  jmp cycle02
  met exit02
  end
endenv
;Tpas Tc types spaces arrayname name file
;arraynames stringnames
env ReadVarInArgs
  arr comma \,
  arr digits \0,\1,\2,\3,\4,\5,\6,\7,\8,\9
  obj number 10 digits
  ;reg get args
    pop Tpas
    pop Tc
    pop types
    pop spaces
    pop arrayname
    pop name
    pop file
  ;endreg
  vct arraynames 0
  vct varnames 0

  ;reg add vectors to varnames
    vct x 0
    vctpshbck varnames x;1
    vct x 0
    vctpshbck varnames x;2
    vct x 0
    vctpshbck varnames x;3
```

```
    vct x 0
    vctpshbck varnames x;4
    vct x 0
    vctpshbck varnames x;5
    vct x 0
    vctpshbck varnames x;6`
  ;endreg
opnwrf log \l\o\g\.\t\x\t
  str x \x
  arr null x
  arrrem null x
  obj otype 1 spaces 1 null, 1 comma 1 null, 1 spaces 1 null, 1 types, 1 spaces 1
   null
  met begin1
  ;reg read data
    scnfltfil type otype file
      jmp end1
    scnfltfil array arrayname file
      jmp iwillreadvar
    ;reg read array
      ;reg prepare actions
        prt myarray array 0
        str tmp \ \:\ \A\r\r\a\y\ \[;last space
        txtctn tmp2 myarray tmp
        str myarray tmp2
        arr openscope \[
        arr closescope \]
      ;endreg
      obj dimension 1 spaces, 1 openscope, 1 spaces, 1 number, 1 spaces, 1
      closescope
      prv file 1
      met readdims
      scnfltfil dim dimension file
        jmp ireadalldims
      ;reg add new dimension
        str tmp \0\.\.
        prt tmp2 dim 3
        txtctn tmp3 tmp tmp2
        str tmp \,
        txtctn tmp2 tmp3 tmp
        txtctn tmp myarray tmp2
        str myarray tmp
      ;endreg
      jmp readdims
      met ireadalldims
      ;reg finish actions
        txtlen tmp myarray
        mth l tmp 1 -
        txtdel myarray l 1
        str tmp \]\ \o\f\ ;last space
        txtctn tmp2 myarray tmp
        prt tmp type 3
        borgtn Tc num tmp
        borgts Tpas tmp num
        txtctn myarrray tmp2 tmp
        vctpshbck arraynames
      ;endreg
```

```
    jmp begin1
  ;endreg
  met iwillreadvar
  ;reg read varname
    scnfltfil varn name file
      nop
    prt stype type 3
    borgtn Tc numb stype
    mts snumb numb
    wrt log snumb
    str stringi \;
    wrt log stringi
    vctget varnames vector numb
    prt sname1 varn 0
    prt sname2 varn 1
    txtctn sname sname1 sname2
    vctpshbck vector sname
    str stringi \;\;
    wrt log stringi
    vctgsz vector llen
    mts stringi llen
    wrt log stringi
    str stringi \~\~
    wrt log stringi
  ;endreg
  jmp begin1
;endreg
met end1
cls log
vct stringnames 0
vctgsz varnames size
mth zero 0
met finishcycle1
;reg cincatenation name and type
  tsteql zero size
    jmp finish
  mth x size 1 -
  mth size x
  vctget varnames vector x
  vctgsz vector size2

  tsteql zero size2
    jmp skip_finishcycle2
  ;reg cycle two
    str string \x
    txtdel string 0 1
    met finishcycle2
    tsteql zero size2
      jmp finish2
    mth x size2 1 -
    mth size2 x
    vctget vector el x
    txtctn string2 string el
    str string \,\ ;
    txtctn el string2 string
    str string el
    jmp finishcycle2
```

```
      met finish2
    ;endreg
    txtlen x string
    mth len x 2 -
    txtdel string len 2
    str string2 \ \:\ ;
    ;prt stype type 3
    ;borgtn Tc num stype
    borgts Tpas stype size
    txtctn string3 string2 stype
    txtctn el string string3
    vctpshbck stringnames el

    mts ssize size
    txtctn string2 el ssize
    opnwrf log \l\o\g\2\.\t\x\t
    wrt log string2
    cls log

    met skip_finishcycle2
  ;endreg
  jmp finishcycle1
  met finish
  psh stringnames
  psh arraynames
  end
endenv
```