```
'2:CPU.PRG       v0.2.0a, 7 августа 2022
'Процессор, исполняющий команды ПМК

'Авторское право (c) Васильев Илья Владимирович, г. Москва
'Это свободная программа, защищённая авторским правом
'Распространяется под лицензией GNU GPL v3 или старше

OPTION STRICT          'Все переменные должны быть объявлены
OPTION DEFINT          'Массивы по умолчанию целые

CONST #Hex$="0123456789abcdef"

VAR flgReplace=0       'Флаг перезаписи

VAR shBody$            'Мантисса вводимого числа или "" (флаг ввода числа)
VAR shExp$             'Порядок вводимого числа или "" (флаг ввода порядка)
VAR shSign,shSignExp   'Знаки мантиссы и порядка вводимого числа
VAR shFixed            'Команда , фиксирует запятую (флаг введённой запятой)
VAR shFormat$          'Признак ввода угла: G/M/S
VAR shExpBack$         'Старый порядок, со знаком, для изменения ВП или ""


'Считать мантиссу из A$, как число, и перевести её в градусы, если flgLand
DEF GetBody#(A$,flgLand)
  VAR I, iD=0
  CASE shFormat$
    WHEN "M":
      VAR rM#=0
      I=INSTR(A$,"°")
      IF I>=0 THEN iD=VAL(MID$(A$,0,I)):A$=MID$(A$,I+1)
      I=INSTR(A$,"'"):IF I>=0 THEN A$[I]="."
      rM#=VAL(A$)
      WHILE rM#>=60:DEC rM#,60:INC iD:WEND 'Должно быть меньше 60 минут
      IF flgLand THEN shFormat$="G":RETURN iD+rM#/60
      RETURN iD+rM#/100
    WHEN "S":
      VAR iM=0, rS#=0
      I=INSTR(A$,"°")
      IF I>=0 THEN iD=VAL(MID$(A$,0,I)):A$=MID$(A$,I+1)
      I=INSTR(A$,"'")
      IF I>=0 THEN iM=VAL(MID$(A$,0,I)):A$=MID$(A$,I+1)
      I=INSTR(A$,"""):IF I>=0 THEN A$[I]="."
      rS#=VAL(A$)
      WHILE rS#>=60:DEC rS#,60:INC iM:WEND 'Должно быть меньше 60 секунд
      WHILE iM>=60: DEC iM,60: INC iD:WEND 'Должно быть меньше 60 минут
      IF flgLand THEN shFormat$="G":RETURN iD+(iM*60+rS#)/3600
      RETURN iD+(iM*100+rS#)/10000
    WHEN "G":  I=INSTR(A$,"°")  'Градусы и их десятичные доли
    OTHERWISE: I=INSTR(A$,",")  'Заменяем десятичную запятую на точку
  ENDCASE
  IF I>=0 THEN A$[I]="."
  RETURN VAL(A$)
END


'Поместить введённое число в RX
'- лучше точная реализация в строке, без промежуточной двоичной арифметики
DEF Finalize
  IF shBody$!="" THEN
    VAR Tmp$
    IF shExp$=="" THEN
      Tmp$=DecCheck$(GetBody#(shBody$,0))
    ELSE
      VAR E=VAL(shExp$)
      IF shSignExp!=0 THEN E=-E                    'Учитываем знак порядка
      IF shExpBack$!="" THEN INC E,VAL(shExpBack$) 'Старый порядок до ВП
      Tmp$=DecCheck$(GetBody#(shBody$,1)*POW(10,E))
    ENDIF
    IF VAR("0:ERROR")==0 THEN
      IF shSign!=0 && !IsZero(Tmp$) THEN
        CASE shFormat$ 'Ставим отрицательный знак мантиссы
          WHEN "G":  Tmp$[0]="g"
          WHEN "M":  Tmp$[0]="m"
          WHEN "S":  Tmp$[0]="s"
          OTHERWISE: Tmp$[0]="-"
        ENDCASE
      ELSE
        IF shFormat$!="" THEN Tmp$[0]=shFormat$
      ENDIF
      SetX$ Tmp$
    ENDIF
    shBody$=""
  ENDIF
END


'Разобрать формат Г: Г,г
COMMON DEF DecryptD A$ OUT strBody$,chExpSign$,strExp$
  chExpSign$=A$[15]:strExp$=MID$(A$,16,2):VAR I=VAL(strExp$)
  IF chExpSign$!=" " OR I==0 THEN 'Отрицательный порядок
    strBody$=A$[1]+"°"+MID$(A$,2,13)
    IF I==0 THEN strExp$="":chExpSign$=""
  ELSEIF I>13 THEN
    strBody$=A$[1]+"°"+MID$(A$,2,13)
  ELSEIF I==13 THEN
    strBody$=MID$(A$,1,14)+"°"
    strExp$="":chExpSign$=""
  ELSE
    strExp$="":chExpSign$=""
    strBody$=MID$(A$,1,I+1)+"°"+MID$(A$,I+2,13-I)
  ENDIF
  I=LAST(strBody$)       'Убрать завершающие нули в дробной части
  WHILE strBody$[I]=="0":DEC I:WEND
  strBody$=LEFT$(strBody$,I+1)
END


'Разобрать формат М: Г,ММм
COMMON DEF DecryptM A$ OUT strBody$,chExpSign$,strExp$
  chExpSign$=A$[15]:strExp$=MID$(A$,16,2):VAR I=VAL(strExp$)
  IF I==0 THEN 'Нулевой порядок, можно убрать в положительный порядок
    IF A$[1]=="0" THEN strBody$="" ELSE strBody$=A$[1]+"°"
    strExp$="":chExpSign$=""
    IF A$[2]!="0" THEN Push strBody$,A$[2]
    Push strBody$,A$[3]+"'"+MID$(A$,4,11)
  ELSEIF chExpSign$!=" " THEN 'Отрицательный порядок
    CASE I
      WHEN 1:     strBody$=A$[1]+A$[2]+"'"+MID$(A$,3,12)
                  strExp$="":chExpSign$=""
      WHEN 2:     strBody$=A$[1]+"'"+MID$(A$,2,13)
                  strExp$="":chExpSign$=""
      OTHERWISE:  strBody$=A$[1]+"'"+MID$(A$,2,13)
                  strExp$=RIGHT$("0"+STR$(I-2),2)
    ENDCASE
  ELSEIF I>13 THEN
```

```
 0  2  4  6  8  10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68 70 72 74
PRG2 CPU.PRG                                                                                    a  13:03
```

```
    ENDCASE
  ELSEIF I>13 THEN
    strBody$=A$[1]+"°"+MID$(A$,2,13)
  ELSE
    strBody$=MID$(A$,1,I+1)+"°"
    strExp$="":chExpSign$=""
    CASE I
      WHEN 13:    ' ГГГГГГГГГГГГГ
        PUSH strBody$,"0'"
      WHEN 12:    ' ГГГГГГГГГГГГМ
        IF A$[14]!="0" THEN PUSH strBody$,A$[14]
        PUSH strBody$,"0'"
      WHEN 11:    ' ГГГГГГГГГГГММ
        IF A$[13]!="0" THEN PUSH strBody$,A$[13]
        PUSH strBody$,A$[14]+"'"
      OTHERWISE:  ' ,ГММм,,
        IF A$[I+2]!="0" THEN PUSH strBody$,A$[I+2]
        PUSH strBody$,A$[I+3]+"'"+MID$(A$,I+4,11-I)
    ENDCASE
  ENDIF
  I=LAST(strBody$)          'Убрать завершающие нули в дробной части
  WHILE strBody$[I]=="0":DEC I:WEND
  strBody$=LEFT$(strBody$,I+1)
END

'Разобрать формат MC: Г,ММССс
COMMON DEF DecryptMS A$ OUT strBody$,chExpSign$,strExp$
  chExpSign$=A$[15]:strExp$=MID$(A$,16,2):VAR I=VAL(strExp$)
  IF I==0 THEN 'Нулевой порядок
    IF A$[1]=="0" THEN strBody$="" ELSE strBody$=A$[1]+"°"
    strExp$="":chExpSign$=""
    IF A$[2]!="0" THEN
      PUSH strBody$,A$[2]+A$[3]+"'"
    ELSEIF A$[3]!="0" OR strBody$!="" THEN
      PUSH strBody$,A$[3]+"'"
    ENDIF
    IF A$[4]!="0" THEN PUSH strBody$,A$[4]
    PUSH strBody$,A$[5]+""+MID$(A$,6,9)
  ELSEIF chExpSign$!=" " THEN 'Отрицательный порядок
    CASE I
      WHEN 1:     strBody$=A$[1]+A$[2]+"'"
                  IF A$[3]!="0" THEN PUSH strBody$,A$[3]
                  PUSH strBody$,A$[4]+""+MID$(A$,5,10)
                  strExp$="":chExpSign$=""
      WHEN 2:     strBody$=A$[1]+"'"
                  IF A$[2]!="0" THEN PUSH strBody$,A$[2]
                  PUSH strBody$,A$[3]+""+MID$(A$,4,11)
                  strExp$="":chExpSign$=""
      WHEN 3:     strBody$=A$[1]+A$[2]+""+MID$(A$,3,12)
                  strExp$="":chExpSign$=""
      WHEN 4:     strBody$=A$[1]+""+MID$(A$,2,13)
                  strExp$="":chExpSign$=""
      OTHERWISE:  strBody$=A$[1]+""+MID$(A$,2,13)
                  strExp$=RIGHT$("0"+STR$(I-4),2)
    ENDCASE
  ELSEIF I>13 THEN
    strBody$=A$[1]+"°"+MID$(A$,2,13)
  ELSE
    strBody$=MID$(A$,1,I+1)+"°"
    strExp$="":chExpSign$=""
    CASE I
      WHEN 13:    ' ГГГГГГГГГГГГГ
        PUSH strBody$,"0'0""
      WHEN 12:    ' ГГГГГГГГГГГГМ
        IF A$[14]!="0" THEN PUSH strBody$,A$[14]
        PUSH strBody$,"0'0""
      WHEN 11:    ' ГГГГГГГГГГГММ
        IF A$[13]!="0" THEN PUSH strBody$,A$[13]
        PUSH strBody$,A$[14]+"'0""
      WHEN 10:    ' ГГГГГГГГГГММС
        IF A$[12]!="0" THEN PUSH strBody$,A$[12]
        PUSH strBody$,A$[13]+"'"
        IF A$[14]!="0" THEN PUSH strBody$,A$[14]
        PUSH strBody$,"0""
      WHEN 9:     ' ГГГГГГГГГММСС
        IF A$[11]!="0" THEN PUSH strBody$,A$[11]
        PUSH strBody$,A$[12]+"'"
        IF A$[13]!="0" THEN PUSH strBody$,A$[13]
        PUSH strBody$,A$[14]+""
      OTHERWISE:  ' ,ГММСС,,
        IF A$[I+2]!="0" THEN PUSH strBody$,A$[I+2]
        PUSH strBody$,A$[I+3]+"'"
        IF A$[I+4]!="0" THEN PUSH strBody$,A$[I+4]
        PUSH strBody$,A$[I+5]+""+MID$(A$,I+6,9-I)
    ENDCASE
  ENDIF
  I=LAST(strBody$)          'Убрать завершающие нули в дробной части
  WHILE strBody$[I]=="0":DEC I:WEND
  strBody$=LEFT$(strBody$,I+1)
END

'Дать мантиссу, порядок десятичного числа, а также их знаки
'Также разобрать гексы, строки и углы
'+ строки возвращаются в кодировке МК-261
COMMON DEF DecryptDec A$ OUT chSign$,strBody$,chExpSign$,strExp$
  chSign$=A$[0]
  CASE chSign$
    WHEN "$":
      strBody$=MID$(A$,1)
    WHEN "#":
      strBody$=MID$(A$,1,4)+":"+MID$(A$,5,4)
'     strExp$=""
    WHEN "G": WHEN "g":
      DecryptD A$ OUT strBody$,chExpSign$,strExp$
    WHEN "M": WHEN "m":
      DecryptM A$ OUT strBody$,chExpSign$,strExp$
    WHEN "S": WHEN "s":
      DecryptMS A$ OUT strBody$,chExpSign$,strExp$
    WHEN " ": WHEN "-":
      chExpSign$=A$[15]
      strExp$=MID$(A$,16,2):VAR I=VAL(strExp$)
      IF chExpSign$!=" " THEN 'Плавающая запятая, отрицательный порядок
        strBody$=A$[1]+","+MID$(A$,2,13)
      ELSEIF I>13 THEN        'Плавающая запятая, положительный порядок
        strBody$=A$[1]+","+MID$(A$,2,13)
      ELSE                    'Естественная запятая
        strExp$=""
        strBody$=MID$(A$,1,I+1)+","+MID$(A$,I+2,13-I)
      ENDIF
      I=LAST(strBody$)        'Убрать завершающие нули в дробной части
      WHILE strBody$[I]=="0":DEC I:WEND
      strBody$=LEFT$(strBody$,I+1)
  ENDCASE
```

```
        strBody$=LEFT$(strBody$,I+1)
    ENDCASE
  END
  
  
  'Дать мантиссу, порядок десятичного числа в RX, а также их знаки
  COMMON DEF DecryptRX OUT chSign$,strBody$,chExpSign$,strExp$
    IF shBody$=="" THEN
      DecryptDec GetX$() OUT chSign$,strBody$,chExpSign$,strExp$
    ELSE
      if shSign==0 THEN chSign$=" " ELSE chSign$="-"
      strBody$=shBody$
      IF shExp$=="" THEN
        chExpSign$=" ":strExp$=""
      ELSE
        if shSignExp==0 THEN chExpSign$=" " ELSE chExpSign$="-"
        strExp$=shExp$
      ENDIF
    ENDIF
  END
  
  
  ' Ввод в стек гекса N
  COMMON DEF DoGex N
    IF shBody$!="" THEN Finalize:flgReplace=0
    IF VAR("0:ERROR")==0 THEN
      IF flgReplace==0 THEN StackUpFast
      SetX$ MkGex$(N)
    ENDIF
    flgReplace=0
  END
  
  
  ' Ввод в стек строки A$
  COMMON DEF DoStr A$
    IF shBody$!="" THEN Finalize:flgReplace=0
    IF VAR("0:ERROR")==0 THEN
      IF flgReplace==0 THEN StackUpFast
      SetX$ "$"+A$
    ENDIF
    flgReplace=0
  END
  
  
  'КОП 00..09 -- Ввод цифры 0..9
  DEF Digit N
    VAR L=14 ' Максимальное количество значащих цифр мантиссы
    IF INSTR(shBody$,".")>=0 THEN INC L
    IF INSTR(shBody$,"%")>=0 THEN INC L
    IF INSTR(shBody$,"'")>=0 THEN INC L
    IF INSTR(shBody$,"~")>=0 THEN INC L
    IF shBody$=="" THEN
      IF flgReplace==0 THEN StackUpFast
      shBody$=#Hex$[N]+","
      shSign=0
      shFixed=0
      shFormat$=""
      shExp$=""
      shExpBack$=""
    ELSEIF shExp$!="" THEN
      shExp$=shExp$[1]+#Hex$[N]
    ELSEIF LEN(shBody$)<L THEN
      IF shFixed==0 THEN
        IF shBody$=="0," THEN shBody$=","
        shBody$[LAST(shBody$)]=#Hex$[N]+","
      ELSE
        PUSH shBody$,#Hex$[N]
      ENDIF
    ENDIF
    flgReplace=1
  END
  
  
  'КОП 0A -- Ввод десятичной запятой
  DEF Comma
    IF shBody$=="" THEN
      IF flgReplace==0 THEN StackUpFast
      shBody$="0,"
      shSign=0
      shFixed=1
      shFormat$=""
      shExp$=""
    ELSEIF shExp$!="" THEN
      VAR("0:ERROR")=11  'Ошибка: ВП ,
    ELSE
      shFixed=1
    ENDIF
    flgReplace=1
  END
  
  
  'КОП 0B -- Смена знака
  'Не меняет признак перезаписи
  '- проверить ввод чисел после /-/
  '- проверить переполнение гекса
  DEF Negate
    IF shBody$=="" THEN
      IF GetX$()[0]=="#" THEN
        GexNot GetX$()
        GexInc GetX$()
      ELSE
        SetX$ MathNeg$(GetX$()) 'умеет изменять знак строк, углов и гексов
      ENDIF
    ELSEIF shExp$=="" THEN
      shSign=1-shSign:Finalize
    ELSE
      shSignExp=1-shSignExp
    ENDIF
  END
  
  
  'КОП 0C -- Ввод порядка
  '- продумать ВП при вводе угла
  '- продумать ВП при изменении порядка гекса и строки
  DEF EE
    VAR chSign$, strBody$, chExpSign$, strExp$
    IF shBody$=="" THEN 'Изменение порядка результата
      IF IsZero() THEN 'Результат ноль
        shBody$="1,"
        shSign = 0
        shExpBack$=""
      ELSE
        LandRX
        DecryptRX OUT chSign$,strBody$,chExpSign$,strExp$
        shBody$=strBody$
```

```
      DecryptRX OUT chSign$,strBody$,chExpSign$,strExp$
      shBody$=strBody$
      shSign = (chSign$="-")
      IF strExp$="00" THEN shExpBack$="" \
      ELSE shExpBack$ = chExpSign$+strExp$
    ENDIF
    shExp$="00"
    shSignExp=0
  ELSEIF shExp$!="" THEN
    Finalize    'Дважды ВП
    IF VAR("0:ERROR")==0 THEN
      DecryptRX OUT chSign$,strBody$,chExpSign$,strExp$
      shBody$=strBody$
      shSign = (chSign$="-")
      shExp$ = "00"
      shSignExp=0
      IF strExp$="00" THEN shExpBack$="" \
      ELSE shExpBack$ = chExpSign$+strExp$
    ENDIF
  ELSE 'ВП после ввода мантиссы
    shExp$="00"
    shSignExp=0
    shExpBack$=""
  ENDIF
  shFixed=1:shFormat$=""
  flgReplace=1
END

'КОП 1f -- ГМС
COMMON DEF DoDMS
  IF shBody$="" THEN
    IF flgReplace==0 THEN StackUpFast
    shBody$="0°"
    shSign=0
    shFixed=1
    shFormat$="G"
    shExp$=""
    shExpBack$=""
    shSignExp=0
  ELSEIF shExp$=="" THEN
    VAR I=LAST(shBody$)
    CASE shFormat$
      WHEN "":
        IF shBody$[I]=="," THEN
          shBody$[I]="°":shFormat$="G":shFixed=1
        ELSE
          I=INSTR(shBody$,",")
          IF I>=0 THEN shBody$[I]="°":shFormat$="G":shFixed=1
        ENDIF
      WHEN "G":
        IF shBody$[I]=="°" THEN shBody$[I]="′" ELSE PUSH shBody$,"′"
        shFormat$="M":shFixed=1
      WHEN "M":
        IF shBody$[I]=="′" THEN shBody$[I]="″" ELSE PUSH shBody$,"″"
        shFormat$="S":shFixed=1
      WHEN "S":
        IF shBody$[I]=="″" THEN
          IF INSTR(shBody$,"′")==-1 AND INSTR(shBody$,"°")==-1 THEN
            shBody$[I]=",":shFormat$="":shFixed=0
          ELSE
            VAR T$
            IF INSTR(shBody$,"′")>=0 THEN
              shFormat$="M":T$=POP(shBody$)
            ELSE
              IF INSTR(shBody$,"°")>=0 THEN shFormat$="G":T$=POP(shBody$)
            ENDIF
          ENDIF ' ;,." без ′ и °
        ENDIF ' ..."
    ENDCASE 'shFormat$
  ENDIF
END

'КОП 27 -- Аварийный останов
DEF KSUB
  Finalize
  IF VAR("0:ERROR")==0 THEN VAR("0:ERROR")=9
  flgReplace=0
END

'КОП 29 -- Аварийный останов
DEF KDIV
  Finalize
  IF VAR("0:ERROR")==0 THEN VAR("0:ERROR")=10
  flgReplace=0
END

'КОП 40..4f -- Запись содержимого регистра X в регистр R
COMMON DEF DoM R
  Finalize
  IF VAR("0:ERROR")==0 THEN STO R
  flgReplace=0
END

'КОП 55 -- Обновление экрана
DEF KSCR
  Finalize
  IF VAR("0:ERROR")==0 THEN RefreshAVT
  flgReplace=0
END

'КОП 60..6f -- Вызов в X содержимого регистра R
COMMON DEF DoRM R
  Finalize
  IF VAR("0:ERROR")==0 THEN RCL R
  flgReplace=0
END

'КОП b0..bf -- Косвенная запись содержимого регистра X
'              по содержимому адресного регистра R
COMMON DEF DoKM R
  Finalize
  IF VAR("0:ERROR")==0 THEN KSTO1 R
  flgReplace=0
END

'КОП fb -- Косвенная запись содержимого регистра X
'          по содержимому адресных регистров от 0 до ffff
```

0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68 70 72 74

PRG2 CPU.PRG

CPU.PRG v0.2a (5/5)