

Untitled

program medAD

```

' *****
' *****

dim temp as byte
dim test as byte
dim col as byte
dim flag1, flag2, flag3, flag4, flag5, flag6, flag7, flag8, flag9, flag10, flag11,
flag12, flag13, flag14, flag15, flag16, flag17, flag18, flag19, flag20, flag21,
flag22, flag23, flag24, flag25, flag26, flag27, flag28 as boolean

dim sec_counter as integer

' adherence data
  dim sec, m, h, dy, dt, mh, y as byte
  dim boxnumber as byte [500]
  dim boxtime as integer [500]
  ' dim boxinfo as byte [100][6]
  dim array_pos as byte
  dim fmin_counter as integer
' end adherence data
' patient data
  dim pName as string [20]
  dim pSS as string [11]
  dim pHeight as string [5]
  dim pWeight as string [5]
  dim pDOB as string [10]
  dim pGender as string [6]
  dim alarm_start_date as byte [3]
  dim alarm_end_date as byte [3]
  dim cdr as boolean ' correct date range
  dim alarm_active, alarm1_active, alarm2_active, alarm3_active, alarm4_active as
boolean
  dim alarm_time1 as byte [2]
  dim alarm_time2 as byte [2]
  dim alarm_time3 as byte [2]
  dim alarm_time4 as byte [2]
  dim start_date_time as byte [5]
' end patient data
sub procedure interrupt
  intcon3.int2f = 0
  sec_counter = sec_counter + 1
  if sec_counter = 5 then '60sec * 15 mins ' needs to be changed back to 900
    fmin_counter = fmin_counter + 1
    sec_counter = 0
  end if
end sub

end sub

' initializes serial port for 9600, N, 8, 1
sub procedure Ser_Init
  TRISC.RX = 1
  TRISC.TX = 1
  TXSTA = 0x20
  RCSTA = 0x90
  SPBRG = 12 ' 12 for 8 MHz, 38 for 24 MHz clock
end sub

' returns 1 if received byte is available, 0 for no byte available
sub function Ser_Data_Ready as byte
  result = PIR1.RCIF

```

```

end sub

' writes one byte to serial port
sub procedure Ser_Write(dim b as byte)
    while (PIR1.TXIF = 0)
        wend
    TXREG = b
end sub

' returns available byte from serial port or 0 if no byte is available
sub function Ser_Read as byte
    if (PIR1.RCIF = 0) then
        result = 0
    else
        result = RCREG
    end if
end sub

' allows a way to simply output a string, instead of outputting bytes one by one
sub procedure Ser_Write_String (dim byref toout as string[20])
    dim count as integer
    dim x as integer
    x = strlen(toout)
    ' Ser_Init
    for count = 0 to x -1
        Ser_Write(toout[count])
    next count
end sub

sub procedure Ser_Write_Time()
    dim count as integer
    ' dim x as integer
    ' x = strlen(boxtime)
    ' if array_pos > 0 then

    for count = 0 to array_pos
        Ser_Write(boxnumber[count])
        Ser_Write(boxtime[count])
        ' Ser_Write(boxinfo[count][0])
        ' Ser_Write(boxinfo[count][1])
        ' Ser_Write(boxinfo[count][2])
        ' Ser_Write(boxinfo[count][3])
        ' Ser_Write(boxinfo[count][4])
        ' Ser_Write(boxinfo[count][5])
        ' Ser_Write(", ")
    next count
    ' end if
end sub

sub procedure rtc_read()

    I2c_start()
    I2c_Wr($d0)
    I2C_Wr(0)
    I2C_Stop
    I2C_Start
    I2c_Wr($d1)
    sec = I2C_Rd(1)
    m = I2C_Rd(1)
    h = I2C_Rd(1)
    dy = I2C_Rd(1)
    dt = I2C_Rd(1)
    mh = I2C_Rd(1)

```

```

y = I2C_Rd(0)
I2C_stop
end sub

sub procedure read_string(dim leng as byte, dim byref inputstr as string[20])
    dim counter as byte
    'dim temp as byte
    'take length from case statement, then read from serial port x times
    inputstr = ""

    for counter = 1 to leng
        while Ser_Data_Ready <> 1

            wend
            temp = Ser_Read ()
            strappendsuf(inputstr, temp)
        next counter
    end sub

sub procedure check_boxes()

    '.... box 1 .....
    portd.0 = 0
    if (portb.4 = 0) then
        if flag1 = 0 then
            portd.2 = 1
            boxnumber[array_pos] = 1 ' correspondin to box number
            boxtime[array_pos] = fi fmi n_counter
            array_pos = array_pos + 1
            Ser_Write("1")
            flag1 = 1
        end if
    end if
    if (portb.4 = 1) then
        portd.2 = 0
        flag1 = 0
    end if
    '..... box 2 .....
    if (portb.5 = 0) then
        if flag2 = 0 then
            portd.2 = 1
            boxnumber[array_pos] = 2
            boxtime[array_pos] = fi fmi n_counter
            array_pos = array_pos + 1
            Ser_Write("2")
            flag2 = 1
        end if
    end if
    if (portb.5 = 1) then
        portd.2 = 0
        flag2 = 0
    end if
    '..... box 3 .....
    if (portb.6 = 0) then
        if flag3 = 0 then
            boxnumber[array_pos] = 3
            boxtime[array_pos] = fi fmi n_counter
            array_pos = array_pos + 1
            Ser_Write("3")
            flag3 = 1
        end if
    end if
    if (portb.6 = 1) then

```

```

    flag3 = 0
end if
'''' box 4 ''''
if (portb.7 = 0) then
    if flag4 = 0 then
        portd.2 = 1
        boxnumber[array_pos] = 4
        boxtime[array_pos] = fi fmi n_counter
        array_pos = array_pos + 1
        Ser_Write("4")
        flag4 = 1
    end if
end if
if (portb.7 = 1) then
    portd.2 = 0
    flag4 = 0
end if

portd.0 = 1
portd.1 = 0

'''' box 5 ''''
if (portb.4 = 0) then
    if flag5 = 0 then

        boxnumber[array_pos] = 5 ' correspondin to box number
        boxtime[array_pos] = fi fmi n_counter
        array_pos = array_pos + 1
        Ser_Write("5")
        flag5 = 1
    end if
end if
if (portb.4 = 1) then
    flag5 = 0
end if
'''' box 6 ''''
if (portb.5 = 0) then
    if flag6 = 0 then
        boxnumber[array_pos] = 6
        boxtime[array_pos] = fi fmi n_counter
        array_pos = array_pos + 1
        Ser_Write("6")
        flag6 = 1
    end if
end if
if (portb.5 = 1) then
    flag6 = 0
end if
'''' box 7 ''''
if (portb.6 = 0) then
    if flag7 = 0 then

        boxnumber[array_pos] = 7
        boxtime[array_pos] = fi fmi n_counter
        array_pos = array_pos + 1
        Ser_Write("7")
        flag7 = 1
    end if
end if
if (portb.6 = 1) then
    flag7 = 0
end if
'''' box 8 ''''

```

```

if (portb.7 = 0) then
  if flag8 = 0 then
    portd.2 = 1
    boxnumber[array_pos] = 8
    boxtime[array_pos] = fi fmi n_counter
    array_pos = array_pos + 1
    Ser_Write("8")
    flag8 = 1
  end if
end if
if (portb.7 = 1) then

  flag8 = 0
end if

portd.1 = 1
portd.2 = 0
'''' box 9 ''''
if (portb.4 = 0) then
  if flag9 = 0 then
    boxnumber[array_pos] = 9 ' correspondin to box number
    boxtime[array_pos] = fi fmi n_counter
    array_pos = array_pos + 1
    Ser_Write("9")
    flag9 = 1
  end if
end if
if (portb.4 = 1) then
  flag9 = 0
end if
'''' box 10 ''''
if (portb.5 = 0) then
  if flag10 = 0 then

    boxnumber[array_pos] = 10
    boxtime[array_pos] = fi fmi n_counter
    array_pos = array_pos + 1
    Ser_Write_string("10")
    flag10 = 1
  end if
end if
if (portb.5 = 1) then
  flag10 = 0
end if
'''' box 11 ''''
if (portb.6 = 0) then
  if flag11 = 0 then
    boxnumber[array_pos] = 11
    boxtime[array_pos] = fi fmi n_counter
    array_pos = array_pos + 1
    Ser_Write_string("11")
    flag11 = 1
  end if
end if
if (portb.6 = 1) then
  portd.2 = 0
  flag11 = 0
end if
'''' box 12 ''''
if (portb.7 = 0) then
  if flag12 = 0 then
    boxnumber[array_pos] = 12
    boxtime[array_pos] = fi fmi n_counter

```

```

        array_pos = array_pos + 1
        Ser_Write_string("12")
        flag12 = 1
    end if
end if
if (portb.7 = 1) then
    flag12 = 0
end if

portd.2 = 1
portd.3 = 0
'''' box 13 ''''
if (portb.4 = 0) then
    if flag13 = 0 then
        boxnumber[array_pos] = 13 ' correspondin to box number
        boxtime[array_pos] = fi fmi n_counter
        array_pos = array_pos + 1
        Ser_Write_string("13")
        flag13 = 1
    end if
end if
if (portb.4 = 1) then
    flag13 = 0
end if
'''' box 14 ''''
if (portb.5 = 0) then
    if flag14 = 0 then

        boxnumber[array_pos] = 14
        boxtime[array_pos] = fi fmi n_counter
        array_pos = array_pos + 1
        Ser_Write_String("14")
        flag14 = 1
    end if
end if
if (portb.5 = 1) then
    flag14 = 0
end if
'''' box 15 ''''
if (portb.6 = 0) then
    if flag15 = 0 then
        boxnumber[array_pos] = 15
        boxtime[array_pos] = fi fmi n_counter
        array_pos = array_pos + 1
        Ser_Write_string("15")
        flag15 = 1
    end if
end if
if (portb.6 = 1) then

    flag15 = 0
end if
'''' box 16 ''''
if (portb.7 = 0) then
    if flag16 = 0 then
        boxnumber[array_pos] = 16
        boxtime[array_pos] = fi fmi n_counter
        array_pos = array_pos + 1
        Ser_Write_string("16")
        flag16 = 1
    end if
end if
if (portb.7 = 1) then

```

```

    flag16 = 0
end if

    portd.3 = 1
    portd.4 = 0
    '.... box 17 .....'
if (portb.4 = 0) then
    if flag17 = 0 then
        boxnumber[array_pos] = 17 ' correspondin to box number
        boxtime[array_pos] = fi fmi n_counter
        array_pos = array_pos + 1
        Ser_Write_string("17")
        flag17 = 1
    end if
end if
if (portb.4 = 1) then
    flag17 = 0
end if
    '.... box 18 .....'
if (portb.5 = 0) then
    if flag18 = 0 then

        boxnumber[array_pos] = 18
        boxtime[array_pos] = fi fmi n_counter
        array_pos = array_pos + 1
        Ser_Write_string("18")
        flag18 = 1
    end if
end if
if (portb.5 = 1) then
    flag18 = 0
end if
    '.... box 19 .....'
if (portb.6 = 0) then
    if flag19 = 0 then
        boxnumber[array_pos] = 19
        boxtime[array_pos] = fi fmi n_counter
        array_pos = array_pos + 1
        Ser_Write_string("19")
        flag19 = 1
    end if
end if
if (portb.6 = 1) then

    flag19 = 0
end if
    '.... box 20 .....'
if (portb.7 = 0) then
    if flag20 = 0 then
        boxnumber[array_pos] = 20
        boxtime[array_pos] = fi fmi n_counter
        array_pos = array_pos + 1
        Ser_Write_string("20")
        flag20 = 1
    end if
end if
if (portb.7 = 1) then
    flag20 = 0
end if

    portd.4 = 1
    portd.5 = 0

```

```

'''''' box 21 ''''
if (portb.4 = 0) then
  if flag21 = 0 then
    boxnumber[array_pos] = 21 ' correspondin to box number
    boxtime[array_pos] = fi fmi n_counter
    array_pos = array_pos + 1
    Ser_Write_string("21")
    flag21 = 1
  end if
end if
if (portb.4 = 1) then
  flag21 = 0
end if
'''''' box 22 ''''
if (portb.5 = 0) then
  if flag22 = 0 then

    boxnumber[array_pos] = 22
    boxtime[array_pos] = fi fmi n_counter
    array_pos = array_pos + 1
    Ser_Write_string("22")
    flag22 = 1
  end if
end if
if (portb.5 = 1) then
  flag22 = 0
end if
'''''' box 23 ''''
if (portb.6 = 0) then
  if flag23 = 0 then
    boxnumber[array_pos] = 23
    boxtime[array_pos] = fi fmi n_counter
    array_pos = array_pos + 1
    Ser_Write_string("23")
    flag23 = 1
  end if
end if
if (portb.6 = 1) then

  flag23 = 0
end if
'''''' box 24 ''''
if (portb.7 = 0) then
  if flag24 = 0 then
    boxnumber[array_pos] = 24
    boxtime[array_pos] = fi fmi n_counter
    array_pos = array_pos + 1
    Ser_Write_string("24")
    flag24 = 1
  end if
end if
if (portb.7 = 1) then
  flag24 = 0
end if

portd.5 = 1
portd.6 = 0
'''''' box 25 ''''

if (portb.4 = 0) then
  if flag25 = 0 then
    boxnumber[array_pos] = 25 ' correspondin to box number
    boxtime[array_pos] = fi fmi n_counter

```

```

    array_pos = array_pos + 1
    Ser_Write_string("25")
    flag25 = 1
end if
end if
if (portb.4 = 1) then
    flag25 = 0
end if
'''''' box 26 ''''''
if (portb.5 = 0) then
    if flag26 = 0 then

        boxnumber[array_pos] = 26
        boxtime[array_pos] = fi fmi n_counter
        array_pos = array_pos + 1
        Ser_Write_string("26")
        flag26 = 1
    end if
end if
if (portb.5 = 1) then
    flag26 = 0
end if
'''''' box 27 ''''''
if (portb.6 = 0) then
    if flag27 = 0 then
        boxnumber[array_pos] = 27
        boxtime[array_pos] = fi fmi n_counter
        array_pos = array_pos + 1
        Ser_Write_string("27")
        flag27 = 1
    end if
end if
if (portb.6 = 1) then

    flag27 = 0
end if
'''''' box 28 ''''''
if (portb.7 = 0) then
    if flag28 = 0 then
        boxnumber[array_pos] = 28
        boxtime[array_pos] = fi fmi n_counter
        array_pos = array_pos + 1
        Ser_Write_string("28")
        flag28 = 1
    end if
end if
if (portb.7 = 1) then
    flag28 = 0
end if

' wend if PORTD.3 = 0 then
' if (flag = 0) then ' flag = 0 not opened before
'     array_pos = array_pos + 1
'     boxnumber[array_pos] = 1 ' would be box number .. i.e. 1, 2, 3
'     boxtime[array_pos] = fi fmi n_counter
'     boxinfo[array_pos][0] = dy
'     boxinfo[array_pos][1] = dt
'     boxinfo[array_pos][2] = mh
'     boxinfo[array_pos][3] = y
'     boxinfo[array_pos][4] = h
'     boxinfo[array_pos][5] = m

```

```

'      array_pos = array_pos + 1
'      'Ser_Write("z")
'      end if
'      flag = 1
' end if
end sub

sub function convert(dim toconvert as byte) as byte
    dim t,o, xbcd as byte

    t = toconvert div 10
    o = toconvert mod 10
    xbcd = (t*16) + o
    result = xbcd
end sub

sub procedure check_date()
    if (mh >= convert(alarm_start_date[0])) and (dy >= convert(alarm_start_date[1]))
and (y >= convert(alarm_start_date[2])) then
        if(mh <= convert(alarm_end_date[0])) and ( dy <= convert(alarm_end_date[1]))
and (y <= convert(alarm_end_date[2])) then
            cdr = true

        else
            cdr = false

        end if
    end if
end sub

sub procedure check_alarms()
    dim alarmON as boolean
    alarmON = false
    ' Ser_Write(m)
'   if (h = convert(alarm_time1[0])) then
'       if (m = convert(alarm_time1[1])) then
'           portd.2 = 1
'       end if
'   end if
    if (alarm_active = true) and (cdr = true ) then
        if (h = convert(alarm_time1[0])) and (m = convert(alarm_time1[1])) then
            alarmON = true
        end if
        if (h = convert(alarm_time2[0])) and (m = convert(alarm_time2[1])) then
            alarmON = true
        end if
        if (h = convert(alarm_time3[0])) and (m = convert(alarm_time3[1])) then
            alarmON = true
        end if
        if (h = convert(alarm_time4[0])) and (m = convert(alarm_time4[1])) then
            alarmON = true
        end if
    end if

    if alarmON = true then
        porta.0 = 1
        'Ser_Write("0")
    else
        porta.0 = 0
    end if
end sub

```

```

    ' Ser_Write(m)
end if
end sub

sub procedure wait_for_ser()
    while Ser_Data_Ready <> 1

        wend
    end sub
sub function btoB(dim toconvert as byte) as byte
    Dim x, y As Byte
    Dim output As Byte

    x = toConvert div 16
    y = toConvert Mod 16
    output = 10 * x + y
    result = output
end sub

main:
    dim leng as byte
    dim xtemp as byte
    dim ctemp as integer
    temp = 0

    'Intcon.gie = 1 ' enable interrupts
    'Intcon3.int2ie = 1 ' enable external interrupt 2
    OSCCON = 0x72 'sets internal osc to 8MHz

    INTCON2.NOT_RBPU = 0 ' turns on all pull up resistors
    ADCON1 = 15 'sets to digital
    trisa.0 = 0

    TRISD = 0 ' Configure all pins for output
    portd = 255 ' turn on all pins on portd
    portd.2 = 0

    TRISB.7 = 1 ' 1 for input
    Trisb.6 = 1
    trisb.5 = 1
    trisb.4 = 1

    ' TRISD.2 = 0 ' Configure pins of PORTB as output
    ' PORTD.2 = 0 ' Turn ON LED on PORTB.1

    ' TRISD.3 = 1 ' set for input
    ' TRISD.1 = 0 ' for output
    Ser_Init ' initiates serial communications
    delay_ms(100)
    I2C_Init(100000)
    delay_ms(100)
    col = 0
    test = 0
    flag1 = 0
    flag2 = 0
    flag3 = 0
    flag4 = 0

    flag5 = 0
    flag6 = 0
    flag7 = 0
    flag8 = 0

```

Untitled

```
fl ag9 = 0
fl ag10 = 0
fl ag11 = 0
fl ag12 = 0

fl ag13 = 0
fl ag14 = 0
fl ag15 = 0
fl ag16 = 0

fl ag17 = 0
fl ag18 = 0
fl ag19 = 0
fl ag20 = 0

fl ag21 = 0
fl ag22 = 0
fl ag23 = 0
fl ag24 = 0

fl ag25 = 0
fl ag26 = 0
fl ag27 = 0
fl ag28 = 0

array_pos = 0

sec_counter = 0
fi fmi n_counter = 0

pName = "xxxx"
pSS = "xxx-xx-xxxx"
pGender = "xxxx"
pWei ght = "xxx"
pHei ght = "xx"
pDOB = "xx-xx-xx"
cdr = true
al arm1_acti ve = fal se
al arm2_acti ve = fal se
al arm3_acti ve = fal se
al arm4_acti ve = fal se
al arm_acti ve = fal se

al arm_start_date[0] = 0
al arm_start_date[1] = 0
al arm_start_date[2] = 0

al arm_end_date[0] = 0
al arm_end_date[1] = 0
al arm_end_date[2] = 0

rtc_read()
del ay_ms(100)
start_date_ti me[0] = mh ' month
start_date_ti me[1] = dy ' day
start_date_ti me[2] = y ' year
start_date_ti me[3] = h ' hour
start_date_ti me[4] = m

del ay_ms(100)

' Ser_Wri te(m)
```

```

'Ser_Write(h)
'Ser_Write(start_date_time[3])
'Ser_Write(start_date_time[4])
delay_ms(500)
ctemp = 0
while true
  if (ctemp = 10000) then

    rtc_read()
    ctemp = 0
  end if
  ctemp = ctemp + 1
  'delay_ms(1000) ' will be taken out later

if Ser_Data_Ready then
  select case Ser_Read
    case "n" ' patient name
      while Ser_Data_Ready <> 1

        wend
        pname = ""
        leng = Ser_Read ()
        read_string(leng, pname)
      case "s" ' ss number
        wait_for_ser()
        pSS = ""
        leng = Ser_Read
        read_string(leng, pss)
      case "x" ' clear patient data
        pname = ""
        pss = ""
        pDOB = ""
        pweight = ""
        pheight = ""
        pGender = ""
      case "q"
        Ser_Write(strlen(pname))
        Ser_Write_string(pname)
        'Ser_Write_Time()
      case "d" ' date of birth
        wait_for_ser()
        pDOB = ""
        leng = Ser_Read
        read_string(leng, pDOB)
      case "w"
        wait_for_ser()
        pWeight = ""
        leng = Ser_Read
        read_string(leng, pWeight)
      case "h"
        wait_for_ser()
        pHeight = ""
        leng = Ser_Read
        read_string(leng, pHeight)
      case "g"
        wait_for_ser()
        pGender = ""
        leng = Ser_Read
        read_string(leng, pGender)
      case "S" ' send everything
        Ser_Write(strlen(pname))
        Ser_Write_string(pname)
        ,,,

```

Untitled

```
Ser_Write(strlen(pDOB))
Ser_Write_String(pDob)
,,,

Ser_Write(strlen(pss))
Ser_Write_string(pss)
,,,

Ser_Write(strlen(pgender))
Ser_Write_string(pgender)
,,,

Ser_Write(strlen(pWeight))
Ser_Write_String(pweight)
,,,

Ser_Write(strlen(pheight))
Ser_Write_String(pheight)
,,,

Ser_Write(btoB(start_date_time[0])) ' month
'del ay_ms(100)
Ser_Write(btoB(start_date_time[1])) ' date
'del ay_ms(100)
Ser_Write(btoB(start_date_time[2])) ' year
'del ay_ms(100)
Ser_Write(btoB(start_date_time[3])) ' hour
'del ay_ms(100)
Ser_Write(btoB(start_date_time[4])) ' min
.....

Ser_Write(alarm_time1[0])
Ser_Write(alarm_time1[1])

Ser_Write(alarm_time2[0])
Ser_Write(alarm_time2[1])

Ser_Write(alarm_time3[0])
Ser_Write(alarm_time3[1])

Ser_Write(alarm_time4[0])
Ser_Write(alarm_time4[1])
,,,

Ser_Write(array_pos)
Ser_Write_Time()
case "W"
' Ser_Write("1")
wait_for_ser()
alarm_time1[0] = Ser_Read()
' Ser_Write(":")
wait_for_ser()
alarm_time1[1] = Ser_Read()
' Ser_Write(".")
case "X"
wait_for_ser()
alarm_time2[0] = Ser_Read()
wait_for_ser()
alarm_time2[1] = Ser_Read()
case "Y"
wait_for_ser()
alarm_time3[0] = Ser_Read()
wait_for_ser()
alarm_time3[1] = Ser_Read()
case "Z"
wait_for_ser()
alarm_time4[0] = Ser_Read()
wait_for_ser()
alarm_time4[1] = Ser_Read()
```

Untitled

```
case "["
  wait_for_ser()
  alarm_start_date[0] = Ser_Read() ' month
  wait_for_ser()
  alarm_start_date[1] = Ser_Read() ' date
  wait_for_ser()
  alarm_start_date[2] = Ser_Read() ' year
case "]"
  wait_for_ser()
  alarm_end_date[0] = Ser_Read() ' month
  wait_for_ser()
  alarm_end_date[1] = Ser_Read() ' date
  wait_for_ser()
  alarm_end_date[2] = Ser_Read() ' year
case "A"
  alarm_active = true
  Ser_Write_String("Alarm activated")
case "a"
  alarm_active = false
  Ser_Write_String("Alarm deactivated")
case "t"
  portd.2 = 1
  delay_ms(200)
  portd.2 = 0
  delay_ms(200)
  portd.2 = 1
  delay_ms(200)
  portd.2 = 0
end select
end if
check_boxes()
check_date()
check_alarms()

' Ser_Write(mh)
' Ser_Write(dy)
' Ser_Write(y)

' Ser_Write(convert(alarm_start_date[0]))
' Ser_Write(convert(alarm_start_date[1]))
' Ser_Write(convert(alarm_start_date[2]))
' Ser_Write(convert(alarm_end_date[0]))
' Ser_Write(convert(alarm_end_date[1]))
' Ser_Write(convert(alarm_end_date[2]))

' ser_write(h)
' ser_write(m)
' Ser_Write(alarm_time1[0])
' Ser_Write(alarm_time1[1])
' Ser_Write(convert(alarm_time1[0]))
' Ser_Write(convert(alarm_time1[1]))
wend
```

end.