A>ED RND.ASM

```
jp start

bdos    equ     5

; versch. Unterprogramme
;

ci:     push hl
        ld c,1
        call bdos       ;zeichen lesen
        pop hl
        ret
co:     push hl
        ld c,2          ;e=wert
        call bdos
        pop hl
        ret

print:  push hl
        ld c,9          ;de -> string
        call bdos
        pop hl
        ret

meldg:
        defb 0dh,0ah,'Eingabe der Recordnr:$'

zahlein:
        ld de,meldg
        call print
        ld hl,0         ;start wert
        call ci         ;erstes
        cp '0'
        jr c,zahlein    ;neuer versuch
        cp '9'+1
        jr nc,zahlein
wdh:    ld d,h
        ld e,l
        add hl,hl
        add hl,hl
        add hl,de
        add hl,hl       ;hl :=  hl * 10
        and 0fh         ;zahl
        ld c,a
        ld b,0
        add hl,bc       ;hl := hl + neue zahl
        call ci         ;zahl
        cp '0'
        jr c,finzahl
        cp '9'+1
        jr nc,finzahl
        jr wdh          ;erneut ausfuehren
finzahl:ret             ;hl = zahlenwert


msgb:
defb 0dh,0ah,'Alter Inhalt:',0dh,0ah,'$'

bufaus:                 ;alten Inhalt ausgeben
        ld de,msgb
        call print

        ld hl,buffer    ;zeichenbuffer
        ld b,128        ;max count
lopbuf: ld e,(hl)       ;zeichen holen
        push bc
        call co
        pop bc
        inc hl
        djnz lopbuf     ;alle 128 zeichen
        ret             ;dann fertig

msgb1:
defb 0dh,0ah,'Neuen Inhalt eingeben',0dh,0ah,'$'

bufein:                 ;neuen Inhalt in
                        Buffer einlesen
        ld de,msgb1
        call print
        ld hl,buffer    ;erst mit leerzeichen
                        vorbelegen
        ld de,buffer+1
        ld bc,128-1
        ld (hl),' '     ;leerzeichen
        ldir
        ld hl,buffer    ;nun neueingabe
        call ci
        cp 0dh
        jr z,carset
        ld b,128        ;max count
lopein: and 7fh
        cp 1            ;CTRL-A beendet
        jr z,carset
        cp ' '          ;> dann ok sonst ende
        jr c,einfin
        ld (hl),a
        inc hl
        push bc
        call ci
        pop bc
        djnz lopein     DEC B, wenn #0 zurück
einfin: xor a           ;kein carry
        ret             ;ok fertig
carset: scf
        ret             ;ende bedingung

msgw:
defb 0dh,0ah,'Buffer neu belegt wird
geschrieben',0dh,0ah,'$'


start:
        ld sp,stack
        ld c,26         ;neue dma - adresse
        ld de,buffer
        call bdos
        ld c,15         ;eroeffnen der Datei
        ld de,fcbn      ;wenn schon da
        call bdos       ;dann ok sonst anlegen
        cp 0ffh         ;=ff dann nicht da
        jr nz,weiter
        ld c,22         ;anlegen noetig
        ld de,fcbn
        call bdos       ;nun muss es klappen
        cp 0ffh         ;bei fehler verlassen
        call z,0        ;hier meldung moeglich
weiter: call zahlein    ;nach hl fuer rndpos
        ld (rndpos),hl  ;nur r0,r1 belegen
        ld c,33         ;versuch zu lesen
        ld de,fcbn
        call bdos       ;wenn schon da anzeigen
        cp 0            ;=0 dann ok
        jr nz,neuan     ;neu anlegen
        call bufaus     ;ausgabe des alten buffers
neuan:  call bufein     ;neuer eingeben nach buffer
        cp 0dh
        jr z,weiter     ;nicht zurueckschreiben
        cp 1
        jr z,finale
        ld de,msgw
        call print
        ld c,34         ;wenn nicht cr
        ld de,fcbn      ;zurueckschreiben
        call bdos
        cp 0
        jr z,weiter     ;das ganze immer weiter so.
finale:
        ld c,16         ;close
        ld de,fcbn
        call bdos
        call 0          ;bei fehler write



fcbn:
        defb 0,'RNDDAT  ','TXT',0,0,0,0
        defs 16
        defb 0          ;cur rec
rndpos: defb 0,0,0              ;rnd zeiger hier
;               lsb .. msb
buffer: defs 128                ;zeichenbuffer
        defs 200                ;stack
stack:  defs 1
end
```

Bild 2. Beispiel für die RND-Befehle