



664 83 757  
 664 83 745  
 49408 44 49452  
 408 155 563  
 664 157 721

"falsch"  
 auf Stack  
 "wahr"  
 aus Stack  
 SWAP  
 GRÖßER  
 holen

>> <<  
 KLEINER  
 >NOT<<

"falsch"  
 speichern  
 "wahr"  
 speichern  
 holen  
 retten  
 holen  
 retten  
 ENDE

>> ANDcc  
 2 Werte holen  
 Lo 1  
 AND Lo 2  
 retten  
 Hi 1  
 AND Hi 2  
 retten  
 speichern

>> ORcc  
 2 Werte holen  
 retten

>> XORcc  
 2 Werte holen

>> +cc  
 2 Werte holen  
 2 Werte addieren  
 Ergebnis  
 speichern

>> -cc  
 2 Werte holen  
 Subtraktion  
 Ergebnis  
 speichern

>> \*cc  
 2 holen  
 0 ins  
 Ergebnis

M  
u  
l  
t  
i  
p  
l  
i  
k  
a  
t  
i  
o  
n

Division

D  
i  
v  
i  
s  
i  
o  
n

>> /cc

>> MODcc

JMP \$002A speichern

C251 >> Rcc  
 von RET-Stack holen  
 auf Stack  
 C257 >> >Rcc  
 von Stack holen  
 auf RET-Stack  
 C25D >> DOcc  
 SWAP  
 auf RET-Stack  
 "

664  
234

664  
256  
920

920

182  
32  
224

3

C266

664 838 664  
102 147  
766 811

664 664  
120 210  
844 880

RET → Stack  
REF Stack-Zähler  
zurück

Byte holen  
Null  
ausgeben  
erhöhen  
↳ 0, LOOP  
ENDE

C26E

RET → Stack  
DUP  
RET → STACK  
SWAP (→ r1-r2-r1)  
Subtr. r2-r1  
DROP  
DROP

C2FB »TYPE«

Wert holen  
Low retten (Länge)  
Wert holen (low index)  
Zähler  
Byte holen  
und ausgeben  
Zähler  
Länge erreicht?  
nein, LOOP  
ENDE

C283 »LOOP«

LDA #400  
LDX #401  
speichern  
1

+LOOP

JSR #003F holen  
INC #02  
SP+2

C310 »EXPECT«

Wert holen  
Low ablegen (Länge)  
Wert holen  
Low  
High  
Zähler auf 0  
Eingabe  
ablegen  
CR?  
ja  
Zähler erhöhen  
Länge erreicht  
nein, LOOP  
ENDE

C327 »EMIT«

JSR #003F holen

C330 »DEPTH«

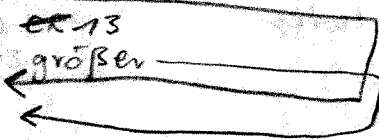
Stack Pointer  
12  
nach X (Low-Byte d. Wertes)  
Null  
ablegen + RTS  
holen  
Low  
High } ablegen  
holen

C338 »GET«

JSR #003F holen  
Low nach Akku  
ausgeben  
EAB7 Tastaturabfrage

C33E »!«

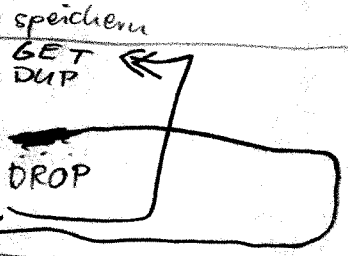
Highbyte an  
Low-Byte Adr+1  
V=0  
ablegen  
ENDE



C34C »C!«

holen  
ablegen  
holen  
für Indizierung  
Low-Byte  
im Speicher  
ablegen

C35C »@«



C35C »@«

holen  
ablegen  
für Indizierung  
holen  
nach x

ablegen  
für Indizierung

4

+1  
 holen  
 ablegen  
 Wert holen  
 ablegen  
 für Indizierung  
 holen  
 nach X  
 High-Byte 0  
 ablegen  
 holen

CS7F >>CALL<<  
 {Register} wie bei  
 {laden} BASIC-Befehl  
 aufrufen SPS

CS8E >>PICK<<  
 holen  
 Stapelzeiger reihen  
 zwischenspeichern  
 Low  
 #2  
 nach Stapel  
 holen  
 Zwischenspeicher  
 in Stapelzeiger  
 Wert ablegen

CSA5 >>ROT<<  
 holen  
 {reihen Wert 1}  
 holen  
 {reihen Wert 2}  
 holen  
 {reihen Wert 3}  
 {Wert 2}  
 speichern  
 {holen Wert 1}  
 speichern  
 {holen Wert 3}  
 speichern

DB  
 Komma  
 LOAD/SAVE Parameter holen  
 OPEN #X,1  
 Flag  
 LOAD

4  
 für  
 "LOAD-SYSTEM"