

1	INIT1	LDA	STAB	;	Tabellenpointer initialisieren
2		STA	PTAB	;	~
3		LDA	SSTOR	;	Storagepointer initialisieren
4		STA	PSTOR	;	~
5		LDA	NULL	;	Hilfsvariable 0 setzen
6		STA	NUM	;	~
7	INIT2	LDA	NULL	;	Zahl 0 laden und an aktuelle adresse im
					Pointer speichern
8		STA,I	PSTOR	;	~
9		LDA	PSTOR	;	Pointer inkrementieren und rückspeichern
10		ADD	ONE	;	~
11		STA	PSTOR	;	~
12		SUB	SSTOR	;	Startadresse abziehen
13		CMP	NINE	;	Mit 9 vergleichen
14		JLE	INIT2	;	Wenn kleiner gleich 9 muss speicher
					weiter initialisiert werden.
15		JMP	INPUT	;	Sonst weiter
16	INPUT	RDA		;	Tastaturpuffer einlesen
17		CMP	ASCOFF	;	Mit ASCII offset von 0 vergleichen.
18		JLT	EXCEPT	;	ASCII codes unter 0 keine Zahlen ->
					Exception
19		SUB	ASCOFF	;	ASCII offset abziehen
20		CMP	NINE	;	Exception für input groesser 9
21		JGT	EXCEPT	;	~
22		JMP	SAVE	;	Weiter zu speichern
23	EXCEPT	LDA	NULL	;	Bei Exception eine 0 speichern
24		JMP	SAVE	;	
25					
26					
27					;---Speichern aller Inputs in die Tabelle---
28	SAVE	STA,I	PTAB;		Wenn exceptionchecks passiert, speichern
					an aktuellen Tabellenplatz
29		LDA	PTAB;		Tabellenpointer inkrementieren und
					rückspeichern
30		ADD	ONE;		~
31		STA	PTAB;		~
32		SUB	STAB;		Pointerstart abziehen -> gibt Anzahl der
					inputs bis jetzt -1
33		CMP	INMAX;		Vergleichen mit Maximaler Anzahl an inputs
34		JGE	CLEAN1;		Falls gleich, ist die Maximale Anzahl an
					inputs erreicht -> weiter zum cleanup und Ausgabe
35		JMP	INPUT;		Sonst nächsten input lesen.
36	CLEAN1	LDA	STAB;		Tabellenpointer auf Startadresse setzen
37		STA	PTAB;		Danach weiter zum Comparator
38		JMP	COMPAR;		
39					
40					
41					;---Zählen, wie oft jede Ziffer vorkommt---
42	COMPAR	LDA,I	PTAB;		Den Wert an der Adresse im
					Tabellenpointer laden
43		ADD	SSTOR;		Startadresse addieren
44		STA	PSTOR;		in den pointer speichern
45		LDA,I	PSTOR;		Zahl an Adresse inkrementieren
46		ADD	ONE;		~
47		STA,I	PSTOR;		Inkrementierten wert Rückspeichern
48		LDA	PTAB;		Tabellenpointer inkrementieren
49		ADD	ONE;		~
50		STA	PTAB;		Rückspeichern
51		SUB	STAB;		Adressoffset abziehen und mit der
					maximalen inputanzahl vergleichen
52		CMP	INMAX;		~
53		JGE	OUTPUT;		Wenn groesser, output, sonst wiederholen
54		JMP	COMPAR;		
55					
56					
57					;---Ausgabe der Ziffern in geordneter Reihenfolge---
58	OUTPUT	LDA	NUM;		Num dient als momentan betrachtete Zahl,
					wurde als 0 initialisiert.
59		ADD	SSTOR;		Startadresse des Storage addieren und in
					den Pointer rückspeichern
60		STA	PSTOR;		~
61		LDA,I	PSTOR;		Zahl im betrachteten Counter als
					Loopzähler -> die Betrachtete Zahl wird so oft ausgegeben.

62	STA	VLOOP;	~
63	JMP	OUTLOOP;	Sprung zum Outloop
64			
65	OUTLOOP	LDA VLOOP;	
66		CMP ONE;	Vor Ausführung der Ausgabe schauen, wie
		oft noch ausgegeben werden soll	
67		JGE NUMOUT;	Wenn eine oder mehr Ausgaben, Sprung zur
		Ausgabe	
68		LDA NUM;	Wenn Vloop erschöpft, betrachtete Zahl
		laden und mit 0 vergleichen	
69		CMP NINE;	~
70		JEQ HALT;	Für 9 als betrachtete Zahl und
71		ADD ONE;	Sonst incrementieren,
72		STA NUM;	Rückspeichern
73		JMP OUTPUT;	und zurück zum Output
74	NUMOUT	LDA NUM;	Betrachtete Zahl laden,
75		ADD ASCOFF;	ASCII-offset addieren
76		WRA;	und Accumulator ausgeben
77		LDA VLOOP;	VLOOP decrementieren
78		SUB ONE;	~
79		STA VLOOP;	und Rückspeichern
80		JMP OUTLOOP;	zurück zu outloop
81	HALT	HLT;	Programm beenden
82		;---Storage---	
83			
84		;---Konstanten---	
85	NULL	DEC 0;	
86	ONE	DEC 1;	
87	NINE	DEC 9;	
88	ASCOFF	DEC 48;	Offset der ASCII 0 zur regulären 0
89	INMAX	DEC 4;	Maximale Anzahl an inputs minus eins
		(also maximale speicheradresse)	
90		;---Zwischenspeicher der Inputs (beliebig zu erweitern)---	
91	STAB	DEF TAB0;	Startadresse der Tabelle
92	PTAB	HEX 0;	Tabellenpointer
93	TAB0	DEC 0;	
94	TAB1	DEC 0;	
95	TAB2	DEC 0;	
96	TAB3	DEC 0;	
97	TAB4	DEC 0;	
98		;---Counter der Einzelnen Zahlen---	
99	NUM	DEC 0;	Hilfsvariable für die Ausgabe
100	SSTOR	DEF C0;	Startadresse des Storage
101	PSTOR	HEX 0;	Speicherpointer
102	C0	DEC 0;	
103	C1	DEC 0;	
104	C2	DEC 0;	
105	C3	DEC 0;	
106	C4	DEC 0;	
107	C5	DEC 0;	
108	C6	DEC 0;	
109	C7	DEC 0;	
110	C8	DEC 0;	
111	C9	DEC 0;	
112		;---Hilfsvariablen---	
113	VLOOP	DEC 0;	Loop-Variable für die Ausgabe
114			
115			