

čune; v tem primeru mora poiskati vrednost string variable, ki se imenuje n\$, to pa je katerokoli ime, ki ste ga vtiskali v zadnjem poizkusu. Ukaz INPUT deluje ravno tako kot LET n\$ = n\$ tako, da je vrednost n\$ nespremenjena.

Sedaj za primerjavo vtiskajte:

n\$

tokrat brez brisanja narekovajev. Variabla n\$ ima vrednost "n\$". Če želite uporabiti STOP pri inputu stringa, morate najprej vrniti kursor na začetek vrste.

Poglejte ponovno RUN 100, ki smo ga že imeli. Računalnik skače na vrstico 100. Ali bi lahko rekli: GO TO 100 namesto RUN 100? V tem primeru je odgovor pozitiven, čeprav je razlika. RUN 100 predvsem briše vse variable in ekran ter nato deluje kot GO TO 100. GO TO ne briše ničesar! Lahko se zgodi, da delate neki program brez, da bi brisali eno samo variabla. Tukaj je GO TO nujen, RUN bi bil lahko katastrofalen; zato je bolje, da se navadite avtomatsko vtiskavati RUN, kadar delate neki program.

Še ena razlika je tudi v tem, da lahko vtiskate RUN brez števila vrste programa, GO TO pa jo mora vedno imeti.

Oba programa sta vstavljena z ukazom STOP v input vrstici (ko je računalnik pričakoval vnos podatkov). Včasih pomotoma vpišete program, ki se ne da ustaviti. Vtiskajte:

```
200 GO TO 200
```

```
RUN 200
```

Izgleda, da je sedaj vse v redu, in da računalnik dela brez prestanka, dokler ga ne izključite, vendar obstaja manj drastičen način, da to storite. Pritisnete CAPS SHIFT skupaj s tipko SPACE, nad katero je napisano BREAK. Program se bo ustavil z obvestilom: L BREAK in to program.

Na koncu vsakega ukaza program pregleduje, ali so tipke CAPS SHIFT in SPACE pritisnjene in če so, se ustavi. Tipko BREAK lahko uporabljate tudi, kadar koristite kasetofon ali printer ter ostalo, kar lahko priključite računalniku, vendar samo v času, ko računalnik pričakuje njihovo delovanje, oni pa ne