enota. Oglejmo si primer, ki nam pokaže kaj se dogaja:

```
LET moja starost = INT (RND 1ØØ): INPUT ("jaz  sem"; moja starost;".");
"Koliko let imate?", vaša leta
```

moja starost - je v oklepajih, zato se njena starost izpiše,
vaša starost - ni v oklepajih, zato morate njeno vrednost vpisati.

Vse, kar ukaz INPUT piše, se pomika proti spodnjemu delu ekrana, ki deluje v glavnem neodvisno od zgornjega dela ekrana. Njegove vrste se v odnosu proti zgornji vrsti spodnje polovice označujejo s številkami, pa čeprav se je ta del pomaknil proti stvarnemu TV ekranu (kar se nam dogaja, če vtipkate množico INPUT podatkov).

Da bi videli, kaj AT počne v ukazih INPUT, izpeljite to:

```
1Ø INPUT "To je vrsta 1.", a$; AT Ø, Ø; "To je vrsta Ø.", a$;
AT 2, Ø; "To je vrsta 2.", a$; AT 1, Ø; "To je še vedno vrsta 1.", a$
```

(vsakič, ko se ustavi pritisnete ENTER). Ko je izpisano "To je vrsta 2", se spodnji del ekrana pomakne proti navzgor, da bi napravil prostor: ker pa se tudi oznake s števili pomaknejo navzgor, so vrste teksta iste številke. Poizkusite sedaj to:

```
1Ø FOR n=Ø TO 19: PRINT AT n,Ø;n;: NEXT n
2Ø INPUT AT Ø,Ø; a$; AT 1,Ø; a$; AT 2,Ø; a$; AT 3,Ø; AT 4,Ø; a$; AT 5,Ø; a$
```

Dokler gre spodnji del ekrana proti navzgor, zgornji del ne bo ogrožen vse do takrat, ko bo spodnji del zagrozil, da bo pričel pisati v isti vrsti kot PRINT pozicija. Da bi se to preprečilo, začne zgornji del s scrolliranjem.

Obstoja še ena lastnost ukaza INPUT, ki jo do sedaj še nismo videli, imenuja pa se LINE INPUT. Predstavlja drugačen način vpisovanja stirng variabl. Če pred ime string variable vpišete LINE, npr. tako:

```
INPUT LINE a$
```

računalnik ne bo zapisal narekovajev, kar sicer dela za string variablami, vendar si bo predstavljal, da so tam. Če vtipkate:

```
maček
```