Prva in najhujša je, da je ta način primeren le za kratke programe, ki jih ne bomo več dopolnjevali ali popravljali. Vrste s strojnim kodom največkrat ni mogoče izpisati na zaslon z ukazom LIST, pogosto tudi ni mogoče uporabiti ukaza EDIT (Caps shift + 1). S tem je popravljanje oz. dopolnjevanje skoraj onemogočeno. Vzrok so nekateri kodi (npr. - desetiško - 10, 6, 13 in drugi), ki imajo čudne učinke na izpis.

Z začetnim naslovom programa v REM-u včasih ni bilo težav. (naslov prvega zloga za znakom REM je bil 23760). Zdaj pa, ko na Spectrum priključimo mikrotračno enoto, se program v basicu (s tem pa tudi REM vrsta ter strojni program) med obdelavo pomika v pomnilniku navzgor in navzdol. S tem je uporaba absolutnih naslovov onemogočena, programiranje pa zelo okrnjeno. Za uporabo programa v REMu si je treba v tem primeru pomagati z ukazoma DEF FN ter FN in z uporabo sistemske spremenljivke PROG izračunati dejanski naslov strojnega programa. Med manjše pomanjkljivosti prištejmo še, da bo ukaz NEW seveda izbrisal tudi REM s strojnim programom vred (kar se ne zgodi, če je program nad RAMTOP).

Prednost uporabe REM vrstice je, da je strojni program tako del programa v basicu. Zato je vedno dovolj le en LOAD, SAVE ali VERIFY ukaz. Vendar to ne odtehta omenjenih težav, zato je shranjevanje strojnega koda v REM stavke le redkokdaj ustrezna rešitev.

## UPORABA PROGRAMSKEGA PROSTORA

Prednosti in pomanjkljivosti takšnega shranjevanja so skoraj enake kot za REM stavke. Razliki: ni težav pri uporabi ukaza LIST, pa tudi možnost shranjevanja nekoliko daljših strojnih programov je laže izvedljiva. Ker je v tem primeru začetni naslov strojne kode popolnoma spremenljiv (odvisen od dolžine basic programa) tudi brez uporabe mikrotračnih enot, je ta način shranjevanja še nekoliko manj priporočljiv kot raba REM vrstic. Zato ga le omenjamo in se z njim ne bomo natančneje ukvarjali.

## UPORABA TISKALNIKOVEGA VMESNEGA POMNILNIKA

Tiskalnikov "buffer" je 256 zlogov dolgo področje, ki sega od naslova 23296 do 23551. Kadar ne uporabljamo tiskalnika, sameva ta prostor prazen in neizkoriščen. Zato je precej uporaben za shranjevanje - pod pogojem, da je strojni program krajši od 256 zlogov ter da ne uporabljamo tiskalnika ali paralelnega vmesnika. Vsaka uporaba ukazov LPRINT, LLIST ali COPY namreč izprazni tiskalnikov prostor. Sicer pa se ta način ne razlikuje dosti od shranjevanja v pomnilniku nad RAMTOPom. Prednost je le, da nam tako ostane na voljo ves drug pomnilniški prostor.



## POSREDOVANJE PODATKOV STROJNEMU PROGRAMU

Zdaj pa nekaj popolnoma drugega. Oglejte si ukaz

```
DEF FN F(X) = USR 29340
```

Recimo, da se na naslovu 29340 začenja nek strojni program. Kadarkoli bomo v basicu uporabili ukaz FN F(X), bomo z njim pognali ta strojni program. "Lepo in prav," boste rekli, "a kakšno korist imamo od tega ? Zakaj ne bi še naprej uporabljali dobrega starega načina RANDOMIZE USR 23340 ?" No, pomislite na X v oklepaju - kaj se dogaja z njim ?

Ena od sistemskih spremenljivk (poglejte 25. poglavje priročnika za Spectrum), ki se imenuje DEFADD, shranjuje naslov spremenljivk, ki so navedene v ukazu FN. Primer:

```
DEF FN F(A,B,C,A$,B$,C$) = USR naslov
```

Uporabljenih je 6 spremenljivk in nekje v pomnilniku je spravljenih 6 skupin podatkov o teh spremenljivkah. DEFADD hrani naslov prve skupine podatkov. Poglejmo, kako so te skupine sestavljene:

```
                [ A ][   ][   1. spremenljivka    ][ , ]
1.skupina       [41 ][0E ][00 ][00 ][aa ][aa ][00 ][2C ]
                  ↑
                (DEFADD)
                [ B ][   ][   2. spremenljivka    ][ , ]
2.skupina       [42 ][0E ][00 ][00 ][bb ][bb ][00 ][2C ]
                [ C ][   ][   3. spremenljivka    ][ , ]
3.skupina       [43 ][0E ][00 ][00 ][cc ][cc ][00 ][2C ]
                [ A ][ $ ][   ][   ][ naslov ][dolžina ][ , ]
4.skupina       [41 ][24 ][0E ][?? ][   ][   ][   ][   ][2C ]
                [ B ][ $ ][   ][   ][ naslov ][dolžina ][ , ]
5.skupina       [42 ][24 ][0E ][?? ][   ][   ][   ][   ][2C ]
                [ C ][ $ ][   ][   ][ naslov ][dolžina ][ ) ]
6.skupina       [43 ][24 ][0E ][?? ][   ][   ][   ][   ][29 ]
```