

Podobno kombinacijo z indeksnim registroma imenujemo sprotno indeksno naslavljanje. To je manj pomembno, kratki zapis ukazov pa se glasi:

```
LD (IX +dis), n
LD (IY +dis), n.
```

#### UPORABA

Poskusimo zdaj v primerih uporabiti nekaj "LD" ukazov. Vemo, da je po povratku iz strojnega programa vrednost funkcije USR enaka vrednosti v registrskem paru BC. Poskusite z naslednjim programom:

```
0E 00 LD C, 0 ;napolni C z 0
C9 RET
```

(Dalej bomo programe zapisovali na ta način, ki ga imenujemo "zbirna oblika" (angleško assembly format). Na levi je strojni kod v šestnajstičnem zapisu, v sredini so krajšave, na desni pa potrebne opombe. Z našim urejevalnikom boste tako programe vnašali brez težav.) Za začetni naslov izberite 30000. Program da registru C vrednost 0; kot veste, je začetna vrednost para BC enaka začetnemu naslovu programa. Kakšen rezultat torej pričakujete?

```
# 0
# 30000
# 29952
```

Poželite program. Ste izbrali pravi odgovor? Če si morda niste čisto na jasnem, zakaj je rezultat takšen, kakršen je, preberite še enkrat poglavje Kako računalniki štejejo.

Da bi vas natančneje seznanili z obravnavanimi ukazi in vam obenem pomagali premagati začetne težave, bomo to in vsa naslednja poglavja zaključili s kratkimi primeri. Toplo vam priporočamo, da jih preizkusite in jih čim pogosteje tudi sami dopolnjujete ali spreminjate. Le tako boste dobili uporabno znanje strojnega jezika.

Izbiri začetnega naslova v primerih prepuščamo vam. Začeli bomo s sprotnim naslavljanjem:

```
06 00 LD B, 0 ;napolni B z 0
OE XX LD C, XX ;XX nadomestite z raznimi vrednostmi
C9 RET ;vrni se v basic
```

Spreminjati smete tudi vrednost v registru B. Poskusite pred vsakim izvajanjem ugotoviti, kakšen bo rezultat.

Nadaljujmo s primerom zunanjega naslavljanja. Spreminjajte vrednost naslova (v primeru je enak 28672) - najbolj zanimive bodo vrednosti, manjše od 16384 (tj. naslovi v ROM-u). Vrednost v izbrani pomnilniški celici najprej naložimo v A (zunanje naslavljanje) in jo od tam prenesemo v C (registrsko naslavljanje):

```
06 00 LD B, 0
3A 00 70 LD A, (naslov) ;napolni A s (trenutno) vsebino
; pomnilniške celice
4F LD C, A
C9 RET
```

Končajmo s primerom za registrsko posredno naslavljanje. Spreminjajte tudi tu vrednost naslova.

```
06 00 LD B, 0
21 00 70 LD HL, naslov
4E LD C, (HL)
C9 RET
```

Prepuščamo vam, da sami poiščete in raziščete primere, ki jih nismo napisali. Kar nekaj jih bo - začnete lahko že z LD (naslov), A, nadaljujete z LD (HL), C in tako naprej. V teh primerih si bo treba nekoliko več pomagati z basicom. Po izvajanju programa boste namreč šele z ukazom PEEK (npr. PEEK (naslov)) lahko ugotovili, če ste dosegli nameravano.