

Instrukcija (heksadekadno)	Instrukcija (binarni zapis)	Instrukcija (simbolički)	Značenje instrukcije (xx, bbbb i dd su redom oznake za heksadekadne, binarne i dekadne brojeve)
000	0000 0000 0000	KRAJ	Kraj programa
1xx	0001 bbbb bbbb	SAB #dd SAB dd	Ako ima # ispred dd, saberi vrednost akumulatora (AK) i vrednost dd; ako nema # ispred dd, saberi vrednost AK i vrednost sa memorijske adrese dd. Rezultat upiši u AK.
2xx	0010 bbbb bbbb	ODU #dd ODU dd	Ako ima # ispred dd, oduzmi od vrednosti AK vrednost dd; ako nema # ispred dd, oduzmi vrednost sa memorijske adrese dd od vrednosti AK. Rezultat upiši u AK.
3xx	0011 bbbb bbbb	AUM dd	Upiši vrednost iz akumulatora na memorijsku adresu dd (iz Akumulatora U Memoriju)
5xx	0101 bbbb bbbb	MUA #dd MUA dd	Ako ima # ispred dd, kopiraj u akumulator vrednost dd; ako nema # ispred dd, kopiraj u akumulator vrednost sa memorijske adrese dd (iz Memorije U Akumulator)
6xx	0110 bbbb bbbb	SKOK dd	Sledeća instrukcija programa je na memorijskoj adresi dd (Skok na instrukciju dd)
7xx	0111 bbbb bbbb	NSKOK dd	Ako je vrednost akumulatora jednaka nuli, sledeća instrukcija programa je na memorijskoj adresi dd; u protivnom, pređi na sledeću instrukciju u programu
8xx	1000 bbbb bbbb	PSKOK dd	Ako je vrednost akumulatora veća ili jednaka nuli (pozitivna ili nula), sledeća instrukcija programa je na memorijskoj adresi dd; u protivnom pređi na sledeću instrukciju u programu
901	1001 0000 0001	ULAZ	Ulaz sa tastature (korisnik unosi vrednost koja se upisuje u akumulator)
902	1001 0000 0010	IZLAZ	Izlaz na ekran (sadržaj akumulatora se ispisuje na ekran)
Axx	1010 bbbb bbbb	MNO #dd MNO dd	Ako ima # ispred dd, pomnoži vrednost akumulatora (AK) i vrednost dd; ako nema # ispred dd, pomnoži AK i vrednost sa memorijske adrese dd; rezultat upiši u AK.
Bxx	1011 bbbb bbbb	KOL #dd KOL dd	Ako ima # ispred dd, podeli AK i vrednost dd; ako nema # ispred dd, podeli AK i vrednost sa memorijske adrese dd. Celobrojni količnik upiši u akumulator.
Cxx	1100 bbbb bbbb	OST #dd OST dd	Ako ima # ispred dd, podeli AK i vrednost dd; ako nema # ispred dd, podeli AK i vrednost sa memorijske adrese dd. Ostatak pri celobrojnom deljenju upiši u akumulator.

Instrukcija (heksadekadno)	Instrukcija (binarni zapis)	Instrukcija (simbolički)	Značenje instrukcije (<i>xx</i> , <i>bbbb</i> i <i>dd</i> su redom oznake za heksadekadne, binarne i dekadne brojeve)
<i>000</i>	0000 0000 0000	KRAJ	Kraj programa
<i>1xx</i>	0001 <i>bbbb bbbb</i>	SAB # <i>dd</i> SAB <i>dd</i>	[AK] + <i>dd</i> → AK, gde je [AK] oznaka za sadržaj (vrednost) akumulatora [AK] + [<i>dd</i>] → AK, gde je [<i>dd</i>] oznaka za vrednost na memorijskoj adresi <i>dd</i>
<i>2xx</i>	0010 <i>bbbb bbbb</i>	ODU # <i>dd</i> ODU <i>dd</i>	[AK] - <i>dd</i> → AK, gde je [AK] oznaka za sadržaj (vrednost) akumulatora [AK] - [<i>dd</i>] → AK, gde je [<i>dd</i>] oznaka za vrednost na memorijskoj adresi <i>dd</i>
<i>3xx</i>	0011 <i>bbbb bbbb</i>	AUM <i>dd</i>	[AK] → <i>dd</i> ili [<i>dd</i>] = [AK]
<i>5xx</i>	0101 <i>bbbb bbbb</i>	MUA # <i>dd</i> MUA <i>dd</i>	<i>dd</i> → AK ili [AK] = <i>dd</i> [<i>dd</i>] → AK ili [AK] = [<i>dd</i>]
<i>6xx</i>	0110 <i>bbbb bbbb</i>	SKOK <i>dd</i>	<i>dd</i> → BI ili [BI] = <i>dd</i>
<i>7xx</i>	0111 <i>bbbb bbbb</i>	NSKOK <i>dd</i>	ako je [AK] = 0, <i>dd</i> → BI, tj. [BI] = <i>dd</i> ; u protivnom, [BI] + 1 → BI, tj. [BI] = [BI] + 1
<i>8xx</i>	1000 <i>bbbb bbbb</i>	PSKOK <i>dd</i>	ako je [AK] ≥ 0, <i>dd</i> → BI, tj. [BI] = <i>dd</i> ; u protivnom, [BI] + 1 → BI, tj. [BI] = [BI] + 1
<i>901</i>	1001 0000 0001	ULAZ	ulaz sa tastature (korisnik unosi vrednost koja se upisuje u akumulator)
<i>902</i>	1001 0000 0010	IZLAZ	izlaz na ekran (sadržaj akumulatora se ispisuje na ekran)
<i>Axx</i>	1010 <i>bbbb bbbb</i>	MNO # <i>dd</i> MNO <i>dd</i>	[AK] * <i>dd</i> → AK, tj. [AK] = [AK] * <i>dd</i> [AK] * [<i>dd</i>] → AK, tj. [AK] = [AK] * [<i>dd</i>]
<i>Bxx</i>	1011 <i>bbbb bbbb</i>	KOL # <i>dd</i> KOL <i>dd</i>	[AK] / <i>dd</i> → AK, tj. [AK] = [AK] / <i>dd</i> (celobrojni količnik) [AK] / [<i>dd</i>] → AK, tj. [AK] = [AK] / [<i>dd</i>]
<i>Cxx</i>	1100 <i>bbbb bbbb</i>	OST # <i>dd</i> OST <i>dd</i>	[AK] mod <i>dd</i> → AK, tj. [AK] = [AK] mod <i>dd</i> (ostatak pri celobrojnem deljenju) [AK] mod [<i>dd</i>] → AK, tj. [AK] = [AK] mod [<i>dd</i>]

Skraćenica	PC/BI	CIR/RI	AK	[41]
Značenje skraćenice	Program Counter / Brojač instrukcija	Current Instruction Register / Registar instrukcije	Akumulator	Sadržaj memorijske adrese 41

Radi jednostavnosti, sve adrese i vrednosti korišćene u programima navedene su kao dekadni brojevi.

Program: Korisnik unosi prirodan broj koji predstavlja veličinu datoteke izraženu u KiB. Program ispisuje kolika je veličina datoteke izražen u bitovima. Koriste se formule: 1 KiB = 1024 B i 1 B = 8 b.

00 ULAZ Korisnik unosi veličinu datoteke u KiB

01 MNO #1024 $[AK] = [AK] * 1024 = \text{Veličina} * 1024$, u ovom primeru $[AK] = 4 * 1024 = 4096$ (B)

02 MNO #8 $[AK] = [AK] * 8 = (\text{Veličina} * 1024) * 8$, u ovom primeru $[AK] = 4096 * 8 = 32768$ (b)

03 IZLAZ Ispisujemo veličinu datoteke u bitovima, u ovom primeru 32768

04 KRAJ

PC/BI	CIR/RI	AK	IZLAZ (ekran)
00	ULAZ	4	
01	MNO #1024	4096	
02	MNO #8	32768	
03	IZLAZ	32768	32768
04	KRAJ	32768	32768

Program: Korisnik unosi dva prirodna broja:

- Broj poena dobijen na osnovu uspeha u srednjoj školi od 0 do 100 (*Srednja*);
- Broj poena osvojen na prijemnom ispitu od 0 do 100 (*Prijemni*).

Rezultat treba da bude ukupan broj poena koji se računa po sledećoj formuli: $(2 * \text{Srednja} + 3 * \text{Prijemni}) / 5$. Memorijske adrese (dekadne) 41 (*Srednja*) i 42 (*Prijemni*) se koriste za čuvanje unetih podataka. Memorijske adrese 43-49 su na raspolaganju za čuvanje međurezultata.

00 ULAZ ; Korisnik unosi poene iz srednje škole, na primer 75, $[AK] = 75$
 01 AUM 41 ; $Srednja = [AK]$, u ovom primeru $Srednja = 75$
 02 ULAZ ; Korisnik unosi poene sa prijemnog, na primer 65, $[AK] = 65$
 03 AUM 42 ; $Prijemni = [AK]$, u ovom primeru $Prijemni = 65$
 04 MNO #3 ; $[AK] = [AK] * 3 = Prijemni * 3$, u ovom primeru $[AK] = 3 * 65 = 195$
 05 AUM 44 ; Međurezultat $Prijemni * 3$ čuvamo na memorijskoj adresi 44, tj. $[44] = 195$
 06 MUA 41 ; $[AK] = Srednja$, u ovom primeru $[AK] = 75$
 07 MNO #2 ; $[AK] = [AK] * 2 = Srednja * 2$, u ovom primeru $[AK] = 2 * 75 = 150$
 08 SAB 44 ; $[AK] = [AK] + [44]$, u ovom primeru $[AK] = 150 + 195 = 345$
 09 KOL #5 ; $[AK] = [AK] / 5$, u ovom primeru $[AK] = 345 / 5 = 69$
 10 IZLAZ ; Ispisujemo ukupan broj poena, u ovom primeru 69
 11 KRAJ ; Kraj programa

PC/BI	CIR/RI	AK	[41] = <i>Srednja</i>	[42] = <i>Prijemni</i>	[44] = <i>Međurezultat</i>	IZLAZ (ekran)
00	ULAZ	75	0	0	0	
01	AUM 41	75	75	0	0	
02	ULAZ	65	75	0	0	
03	AUM 42	65	75	65	0	
04	MNO #3	195	75	65	0	
05	AUM 44	195	75	65	195	
06	MUA 41	75	75	65	195	
07	MNO #2	150	75	65	195	
08	SAB 44	345	75	65	195	
09	KOL #5	69	75	65	195	
10	IZLAZ	69	75	65	195	69
11	KRAJ	69	75	65	195	69

Program: Korisnik unosi dva prirodna broja:

- iznos plate (na primer, 240) koji treba povećati za $p\%$ (na primer, $15\% = 15/100 = 0,15$) i
- broj p (100 puta veći od procenta, u ovom primeru 15)

Rezultat treba da bude novi iznos plate posle uvećanja za $p\%$. Memorijske adrese (dekadne) 41 (*iznos*) i 42 (p) se koriste za čuvanje unetih podataka. Memorijske adrese 43-49 su na raspolaganju za čuvanje međurezultata.

00 ULAZ ; Korisnik unosi iznos plate, na primer 240, [AK] = 240
 01 AUM 41 ; $iznos = [AK]$, u ovom primeru $iznos = 240$
 02 ULAZ ; Korisnik unosi 100 puta veći procenat, tj. broj p , na primer 15, [AK] = 15
 03 AUM 42 ; $p = [AK]$, u ovom primeru $p = 15$
 04 MNO 41 ; $[AK] = [AK] * iznos = p * iznos$ (računamo za koliko se plata povećala: $[AK] = 15 * 240 = 3600$)
 05 KOL #100 ; $[AK] = [AK] / 100 = (p * iznos) / 100$ (računamo $p\%$ tako što množimo sa p i delimo sa 100: $[AK] = 3600 / 100 = 36$)
 06 SAB 41 ; $[AK] = [AK] + iznos = (p / 100) * iznos + iznos$ (računamo novu platu: $[AK] = 36 + 240 = 276$)
 07 IZLAZ ; Ispisujemo novu platu
 08 KRAJ ; Kraj programa

PC/BI	CIR/RI	AK	[41] = iznos	[42] = p	[44] = Međurezultat	IZLAZ (ekran)
00	ULAZ	240	0	0	0	
01	AUM 41	240	240	0	0	
02	ULAZ	15	240	0	0	
03	AUM 42	15	240	15	0	
04	MNO 41	3600	240	15	0	
05	KOL #100	36	240	15	0	
06	SAB 41	276	240	15	0	
07	IZLAZ	276	240	15	0	276
08	KRAJ	276	240	15	0	276

Program: Korisnik unosi dva prirodna broja, program čuva veći od njih na memorijskoj adresi 43 (*Maksimum*) i ispisuje ga. Memorijske adrese (dekadne) 41 (*Prvi*) i 42 (*Drugi*) se koriste za čuvanje unetih podataka.

00 ULAZ ; Korisnik unosi prvi broj, u primeru pod a) 240, [AK] = 240
 01 AUM 41 $Prvi = [AK]$, u primeru pod a) Iznos = 240
 02 ULAZ ; Korisnik unosi drugi broj, u primeru pod a) 215, [AK] = 215
 03 AUM 42 $Drugi = [AK]$, u primeru pod a) $Drugi = 215$
 04 ODU 41 $[AK] = [AK] - [41] = Drugi - Prvi$, u primeru pod a) $[AK] = 215 - 240 = -35$
 05 PSKOK 08; Ako je $[AK] \geq 0$, [BI] = 08, tj. sledeća instrukcija je na adresi 08 ako je vrednost akumulatora nenegativna (u primeru pod a) nije)
 06 MUA 41 Ako smo stigli dovde (u primeru pod a) jesmo), razlika $Drugi - Prvi$ je manja od nule, tj. *Prvi* je veći, pa $[AK] = [41] = Prvi = 240$
 07 SKOK 09 [BI] = 09, tj. sledeća instrukcija je na adresi 09
 08 MUA 42 Ako smo stigli dovde (u primeru pod a) to se neće desiti), razlika $Drugi - Prvi \geq 0$, tj. *Drugi* je veći, pa $[AK] = [42] = Drugi$
 09 AUM 43 $[43] = [AK] = \text{Maksimum}$ (u primeru pod a) 240)
 10 IZLAZ ispisujemo maksimum
 11 KRAJ

Prikažimo tablicom kako se izvršava dati program za različite unete brojeve: a) 240 i 215; b) 140 i 315.

a) Na početku su sve memorijske adrese i registri jednaki nuli, ekran je prazan. Korisnik unosi redom brojeve 240 i 215

PC/BI	CIR/RI	AK	[41] = <i>Prvi</i>	[42] = <i>Drugi</i>	[43] = <i>Max</i>	IZLAZ (ekran)
00	ULAZ	240	0	0	0	
01	AUM 41	240	240	0	0	
02	ULAZ	215	240	0	0	
03	AUM 42	215	240	215	0	
04	ODU 41	-25	240	215	0	
05	PSKOK 08	-25	240	215	0	
06	MUA 41	240	240	215	0	
07	SKOK 09	240	240	215	0	
09	AUM 43	240	240	215	240	
10	IZLAZ	240	240	215	240	240
11	KRAJ	240	240	215	240	240

b) Na početku su sve memorijske adrese i registri jednaki nuli, ekran je prazan. Korisnik unosi redom brojeve 140 i 315

- 00 ULAZ ; Korisnik unosi prvi broj, u primeru pod b), $[AK] = 140$
 01 AUM 41 $Prvi = [AK]$, u ovom primeru Iznos = 140
 02 ULAZ ; Korisnik unosi drugi broj, u primeru pod b) 315, $[AK] = 315$
 03 AUM 42 $Drugi = [AK]$, u primeru pod b) $Drugi = 315$
 04 ODU 41 $[AK] = [AK] - [41] = Drugi - Prvi$, u primeru pod b) $[AK] = 315 - 140 = 175$
 05 PSKOK 08; Ako je $[AK] \geq 0$, $[BI] = 08$, tj. sledeća instrukcija je na adresi 08 ako je vrednost akumulatora nenegativna (u primeru pod b) jeste)
 06 MUA 41 U primeru pod b) instrukcije 06 i 07 su nedostižne jer je razlika $Drugi - Prvi$ veća od nule, tj. preskočene su
 07 SKOK 09 U primeru pod b) instrukcije 06 i 07 su nedostižne jer je razlika $Drugi - Prvi$ veća od nule, tj. preskočene su
 08 MUA 42 Ako smo stigli dovd (u primeru pod b) jesmo), razlika $Drugi - Prvi \geq 0$, tj. $Drugi$ je veći, pa $[AK] = [42] = Drugi = 315$
 09 AUM 43 $[43] = [AK] = \text{Maksimum}$ (u ovom primeru 315)
 10 IZLAZ ispisujemo maksimum
 11 KRAJ

PC/BI	CIR/RI	AK	$[41] = Prvi$	$[42] = Drugi$	$[43] = Max$	IZLAZ (ekran)
00	ULAZ	140	0	0	0	
01	AUM 41	140	140	0	0	
02	ULAZ	315	140	0	0	
03	AUM 42	315	140	315	0	
04	ODU 41	175	140	315	0	
05	PSKOK 08	175	140	315	0	
08	MUA 42	315	140	315	0	
09	AUM 43	315	140	315	315	
10	IZLAZ	315	140	315	315	315
11	KRAJ	315	140	315	315	315

Program: Korisnik unosi svoje ocene (5-10) jednu po jednu i signalizira računaru da je završio unosom nule. Računar na osnovu unetih ocena izračunava prosečnu ocenu, upisuje je na memorijskoj adresi 44 (*Prosek*) i ispisuje na izlaz. Memorijske adrese (dekadne) 41 (*Ocena*), 42 (*ZbirOcena*), 43 (*BrojOcena*) i 44 (*Prosek*) se koriste za čuvanje unetih podataka (na početku su jednake nuli). Testirati program pod pretpostavkom da je korisnik uneo: 9 5 7 0.

```

00  ULAZ      ; Korisnik unosi ocenu
01  NSKOK 13 ; Ako je uneta nula, korisnik je završio sa unosom, idemo da računamo prosek ako je uneta bar jedna prelazna ocena (veća od 5)
02  AUM 41   ; Čuvamo ocenu u memoriji,  $Ocena = [41] = [AK]$ 
03  ODU #6   ; Računamo razliku  $Ocena - 6$ 
04  PSKOK 06 ; Ako je  $Ocena \geq 6$ , tj.  $Ocena - 6 \geq 0$ , skok na obradu
05  SKOK 00  ; Pošto  $Ocena$  nije prelazna, ne ulazi u prosek, vraćamo se na unos
06  MUA 42   ; Ocena je prelazna, pa ulazi u zbir,  $[AK] = [42] = ZbirOcena$ 
07  SAB 41   ;  $[AK] = [AK] + [41] = ZbirOcena + Ocena$ 
08  AUM 42   ;
09  MUA 43   ; Ocena je prelazna, pa je brojimo
10  SAB #1   ;  $[AK] = [AK] + 1 = BrojOcena + 1$ 
11  AUM 43   ;
12  SKOK 00  ; Vraćamo se na unos
13  MUA 43   ; Računamo prosek ako je bar jedna ocena veća od 5
14  NSKOK 19 ; Nijedna ocena nije veća od petice
15  MUA 42   ;
16  KOL 43   ;  $[AK] = [AK] / [42] = ZbirOcena / BrojOcena$ 
17  AUM 44   ;  $Prosek = [AK]$ 
18  IZLAZ
19  KRAJ

```

	Ciklus / petlja	PC /BI	CIR/RI	AK	[41] = <i>Ocena</i>	[42] = <i>ZbirOcena</i>	[43] = <i>BrojOcena</i>	[44] = <i>Prosek</i>	IZLAZ (ekran)
	1	00	ULAZ	9	0	0	0	0	
		01	NSKOK 13	9	0	0	0	0	
		02	AUM 41	9	9	0	0	0	

Program	Ciklus / petlja	PC /BI	CIR/RI	AK	[41] = Ocena	[42] = ZbirOcena	[43] = BrojOcena	[44] = Prosek	IZLAZ (ekran)
00 ULAZ		03	ODU #6	3	9	0	0	0	
01 NSKOK 13		04	PSKOK 06	3	9	0	0	0	
02 AUM 41		06	MUA 42	0	9	0	0	0	
03 ODU #6		07	SAB 41	9	9	0	0	0	
04 PSKOK 06		08	AUM 42	9	9	9	0	0	
05 SKOK 00		09	MUA 43	0	9	9	0	0	
06 MUA 42		10	SAB #1	1	9	9	0	0	
07 SAB 41		11	AUM 43	1	9	9	1	0	
08 AUM 42		12	SKOK 00	1	9	9	1	0	
09 MUA 43									
10 SAB #1									
11 AUM 43									
12 SKOK 00	2	00	ULAZ	5	9	9	1	0	
13 MUA 43		01	NSKOK 13	5	9	9	1	0	
14 NSKOK 19		02	AUM 41	5	5	9	1	0	
15 MUA 42		03	ODU #6	-1	5	9	1	0	
16 KOL 43		04	PSKOK 06	-1	5	9	1	0	
17 AUM 44		05	SKOK 00	-1	5	9	1	0	
18 IZLAZ									
19 KRAJ									
	3	00	ULAZ	7	5	9	1	0	
		01	NSKOK 13	7	5	9	1	0	
		02	AUM 41	7	7	9	1	0	
		03	ODU #6	1	7	9	1	0	
		04	PSKOK 06	1	7	9	1	0	
		06	MUA 42	9	7	9	1	0	

Program: Korisnik unosi dva prirodna broja, osnovu a i izložilac (eksponent) n i ispisuje kao rezultat stepen a^n , prethodno sačuvan na memorijskoj adresi 43 (*Stepen*). Memorijske adrese (dekadne) 41 (*Osnova*), 42 (*Eksponent*) i 43 (*Stepen*) se koriste za čuvanje unetih podataka (na početku su *Osnova* i *Eksponent* jednaki nuli, dok je *Stepen* jednak 1). Testirati program pod pretpostavkom da je korisnik uneo brojeve 2 i 3.

```
00  ULAZ      ; Korisnik unosi osnovu
01  AUM 41    ; [41] = Osnova = 2
02  NSKOK 14 ; Ako je osnova nula i rezultat je nula, idemo na kraj
03  MUA #1    ; [AK] = 1
04  AUM 43    ; [43] = Stepen = 1, tj. početna vrednost stepena je 1 (neutralni element množenja)
05  ULAZ      ; Korisnik unosi Eksponent
06  NSKOK 14 ; Ako je Eksponent (odnosno Brojač u narednim ciklusima) jednak nuli, rezultat je već na memorijskoj adresi 43 (Stepen)
07  ODU #1    ; U protivnom, [AK] = [AK] - 1, smanjujemo Brojač, tj. počinjemo da brojimo unazad od (Eksponent - 1) do nule
08  AUM 42    ; [42] = Brojač
09  MUA 43    ; [AK] = [43] = 1
10  MNO 41    ; [AK] = [AK] * [41], trenutnu vrednost Stepena množimo Osnovom
11  AUM 43    ; [43] = [AK], čuvamo novu vrednost Stepena
12  MUA 42    ; [AK] = [42], vraćamo Brojač u akumulator
13  SKOK 06   ; i skaćemo na proveru da li je Brojač stigao do nule
14  MUA 43    ; [AK] = [43], vraćamo Stepen u akumulator
15  IZLAZ     ; Ispisujemo Stepen
16  KRAJ
```

Program	Ciklus / petlja	PC/BI	CIR/RI	AK	[41] = Osnova	[42] = Brojač	[43] = Stepen	IZLAZ (ekran)
00 ULAZ 01 AUM 41 02 NSKOK 14 03 MUA #1 04 AUM 43 05 ULAZ 06 NSKOK 14 07 ODU #1 08 AUM 42 09 MUA 43 10 MNO 41 11 AUM 43 12 MUA 42 13 SKOK 06 14 MUA 43 15 IZLAZ 16 KRAJ		00	ULAZ	2	0	0	0	
		01	AUM 41	2	2	0	0	
		02	NSKOK 14	2	2	0	0	
		03	MUA #1	1	2	0	0	
		04	AUM 43	1	2	0	1	
		05	ULAZ	3	2	0	1	
		06	NSKOK 14	3	2	0	1	
		07	ODU #1	2	2	0	1	
		08	AUM 42	2	2	2	1	
		09	MUA 43	1	2	2	1	
		10	MNO 41	2	2	2	1	
		11	AUM 43	2	2	2	2	
		12	MUA 42	2	2	2	2	
		13	SKOK 07	2	2	2	2	
		06	NSKOK 14	2	2	2	2	
		07	ODU #1	1	2	2	2	
		08	AUM 42	1	2	1	2	
		09	MUA 43	2	2	1	2	
		10	MNO 41	4	2	1	2	
		11	AUM 43	4	2	1	4	
		12	MUA 42	1	2	1	4	

Program	Ciklus / petlja	PC/BI	CIR/RI	AK	[41] = Osnova	[42] = Brojač	[43] = Stepen	IZLAZ (ekran)
00 ULAZ		13	SKOK 06	1	2	1	4	
01 AUM 41		06	NSKOK 14	1	2	1	4	
02 NSKOK 14		07	ODU #1	0	2	1	4	
03 MUA #1		08	AUM 42	0	2	0	4	
04 AUM 43		09	MUA 43	4	2	0	4	
05 ULAZ		10	MNO 41	8	2	0	4	
06 NSKOK 14		11	AUM 43	8	2	0	8	
07 ODU #1		12	MUA 42	0	2	0	8	
08 AUM 42		13	SKOK 06	0	2	0	8	
09 MUA 43		06	NSKOK 14	0	2	0	8	
10 MNO 41		14	MUA 43	8	2	0	8	
11 AUM 43		15	IZLAZ	8	2	0	8	8
12 MUA 42		16	KRAJ	8	2	0	8	8
13 SKOK 06								
14 MUA 43								
15 IZLAZ								
16 KRAJ								