

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	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)
000	0000 0000 0000	KRAJ	Kraj programa
1xx	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>
2xx	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>
3xx	0011 <i>bbbb bbbb</i>	AUM <i>dd</i>	AK → <i>dd</i> ili [<i>dd</i>] = [AK]
5xx	0101 <i>bbbb bbbb</i>	MUA <i>dd</i>	<i>dd</i> → AK ili [AK] = [<i>dd</i>]
6xx	0110 <i>bbbb bbbb</i>	SKOK <i>dd</i>	<i>dd</i> → BI ili [BI] = <i>dd</i>
7xx	0111 <i>bbbb bbbb</i>	NSKOK <i>dd</i>	ako je [AK] = 0 <i>dd</i> → BI; u protivnom, [BI] + 1 → BI
8xx	1000 <i>bbbb bbbb</i>	PSKOK <i>dd</i>	ako je [AK] ≥ 0 <i>dd</i> → BI; u protivnom, [BI] + 1 → BI
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 <i>bbbb bbbb</i>	MNO # <i>dd</i> MNO <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>
Bxx	1011 <i>bbbb bbbb</i>	KOL # <i>dd</i> KOL <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>
Cxx	1100 <i>bbbb bbbb</i>	OST # <i>dd</i> OST <i>dd</i>	[AK] mod <i>dd</i> → AK, gde je [AK] oznaka za sadržaj (vrednost) akumulatora [AK] mod [<i>dd</i>] → AK, gde je [<i>dd</i>] oznaka za vrednost na memorijskoj adresi <i>dd</i>

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 DEL #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	DEL #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) 041 (*iznos*) i 042 (p) se koriste za čuvanje unetih podataka. Memorijske adrese 043-049 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 DEL #100 ; [AK] = [AK] / 100 = $p / 100$ (računamo $p\%$ tako što delimo p sa 100: [AK] = $15/100 = 15\% = 0,15$)

05 MNO Iznos ; [AK] = [AK] * *iznos* = $(p / 100) * iznos$ (računamo za koliko se plata povećala: [AK] = $(15/100) * 240 = 36$)

06 SAB Iznos ; [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] = Srednja	[42] = Prijemni	[44] = Međurezultat	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	

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, na primer 240, [AK] = 240
 01 AUM 41 ; *Prvi* = [AK], u ovom primeru Iznos = 240
 02 ULAZ ; Korisnik unosi drugi broj, na primer 215, [AK] = 215
 03 AUM 42 ; *Drugi* = [AK], u ovom primeru *Drugi* = 215
 04 ODU 41 ; [AK] = [AK] – [41] = *Drugi* – *Prvi*, u ovom primeru [AK] = 215 – 240 = -35
 05 PSKOK 08; Ako je [AK] ≥ 0, [BI] = 08, tj. sledeća instrukcija je na adresi 08 ako je vrednost akumulatora nenegativna (u ovom slučaju nije)
 06 MUA 41 ; Pošto smo stigli dovdde, 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 ; Pošto smo stigli dovdde, razlika *Drugi* – *Prvi* ≥ 0, tj. *Drugi* je veći, pa [AK] = [42] = *Drugi* (u ovom primeru, tj. pod a) to se neće desiti)
 09 AUM 43 ; [43] = [AK] = Maksimum (u ovom primeru 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 (<i>ekran</i>)
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	-35	240	215	0	
05	PSKOK 08	-35	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, na primer 140, $[AK] = 140$
 01 AUM 41 ; $Prvi = [AK]$, u ovom primeru Iznos = 140
 02 ULAZ ; Korisnik unosi drugi broj, na primer 315, $[AK] = 315$
 03 AUM 42 ; $Drugi = [AK]$, u ovom primeru $Drugi = 315$
 04 ODU 41 ; $[AK] = [AK] - [41] = Drugi - Prvi$, u ovom primeru $[AK] = 315 - 140 = 175$
 05 PSKOK 08; Ako je $[AK] \geq 0$, $[BI] = 08$, tj. sledeća instrukcija je na adresi 08 ako je vrednost akumulatora nenegativna (u ovom slučaju jeste)
 06 MUA 41 ; U ovom slučaju instrukcije 06 i 07 su nedostižne jer je razlika $Drugi - Prvi$ veća od nule, tj. preskočene su
 07 SKOK 09 ; U ovom slučaju instrukcije 06 i 07 su nedostižne jer je razlika $Drugi - Prvi$ veća od nule, tj. preskočene su
 08 MUA 42 ; Pošto smo stigli dođe, 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 $Ocena \geq 6$, skok na obradu
 05 SKOK 00 ; Pošto *Ocena* nije ni nula ni 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									
03 ODU #6		06	MUA 42	0	9	0	0	0	
04 PSKOK 06		07	SAB 41	9	9	0	0	0	
05 SKOK 00									
06 MUA 42		08	AUM 42	9	9	9	0	0	
07 SAB 41		09	MUA 43	0	9	9	0	0	
08 AUM 42									
09 MUA 43		10	SAB #1	1	9	9	0	0	
10 SAB #1		11	AUM 43	1	9	9	1	0	
11 AUM 43									
12 SKOK 00		12	SKOK 00	1	9	9	1	0	
13 MUA 43	2	00	ULAZ	5	9	9	1	0	
14 NSKOK 19		01	NSKOK 13	5	9	9	1	0	
15 MUA 42									
16 KOL 43		02	AUM 41	5	5	9	1	0	
17 AUM 44									
18 IZLAZ		03	ODU #6	-1	5	9	1	0	
19 KRAJ		04	PSKOK 06	-1	5	9	1	0	
		05	SKOK 00	-1	5	9	1	0	
	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	ciklus/ petlja	PC/ BI	CIR/RI	AK	[41] = Ocena	[42] = ZbirOcena	[43] = BrojOcena	[44] = Prosek	IZLAZ (ekran)
00 ULAZ		07	SAB 41	16	7	9	1	0	
01 NSKOK 13		08	AUM 42	16	7	16	1	0	
02 AUM 41		09	MUA 43	1	7	16	1	0	
03 ODU #6		10	SAB #1	2	7	16	1	0	
04 PSKOK 06		11	AUM 43	2	7	16	2	0	
05 SKOK 00		12	SKOK 00	2	7	16	2	0	
06 MUA 42		12	SKOK 00	2	7	16	2	0	
07 SAB 41	4	00	ULAZ	0	7	16	2	0	
08 AUM 42		01	NSKOK 13	0	7	16	2	0	
09 MUA 43		01	NSKOK 13	0	7	16	2	0	
10 SAB #1	Kraj ciklusa								
11 AUM 43		13	MUA 43	2	7	16	2	0	
12 SKOK 00		14	NSKOK 19	2	7	16	2	0	
13 MUA 43		14	NSKOK 19	2	7	16	2	0	
14 NSKOK 19		15	MUA 42	16	7	16	2	0	
15 MUA 42		15	MUA 42	16	7	16	2	0	
16 KOL 43		16	KOL 43	8	7	16	2	0	
17 AUM 44		17	AUM 44	8	7	16	2	8	
18 IZLAZ		18	IZLAZ	8	7	16	2	8	8
19 KRAJ		19	KRAJ	8	7	16	2	8	8

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 15 ; Ako je osnovu nula i rezultat je nula, idemo na kraj
 03 MUA 43 ; [AK] = [43] = 0
 04 SAB #1 ; [AK] = [AK] + 1 = 0 + 1 = 1
 05 AUM 43 ; [43] = *Stepen* = 1, tj. početna vrednost stepena je 1 (neutralni element množenja)
 06 ULAZ ; Korisnik unosi eksponent
 07 NSKOK 15 ; Ako je eksponent jednak nuli, rezultat je već na memorijskoj adresi 43 (*Stepen*)
 08 ODU #1 ; U protivnom, [AK] = [AK] - 1, smanjujemo eksponent za 1
 09 AUM 42 ; [42] = *Eksponent*
 10 MUA 43 ; [AK] = [43] = 1
 11 MNO 41 ; [AK] = [AK] * [41], trenutnu vrednost stepena množimo osnovom
 12 AUM 43 ; [43] = [AK], čuvamo novu vrednost stepena
 13 MUA 42 ; [AK] = [42], vraćamo eksponent u akumulator
 14 SKOK 07 ; i skaćemo na proveru da li je eksponent jednak nuli
 15 MUA 43 ; [AK] = [43], vraćamo stepen u akumulator
 16 IZLAZ ; Ispisujemo stepen
 17 KRAJ

	PC/BI	CIR/RI	AK	[41] = <i>Osnova</i>	[42] = <i>Eksponent</i>	[43] = <i>Stepen</i>	IZLAZ (ekran)
	00	ULAZ	2	0	0	0	
	01	AUM 41	2	2	0	0	
	02	NSKOK 15	2	2	0	0	
	03	MUA 43	0	2	0	0	

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

Program	PC/BI	CIR/RI	AK	[41] = <i>Osnova</i>	[42] = <i>EkspONENT</i>	[43] = <i>Stepen</i>	IZLAZ (ekran)
00 ULAZ	09	AUM 42	0	2	0	4	
01 AUM 41	10	MUA 43	4	2	0	4	
02 NSKOK 15	11	MNO 41	8	2	0	4	
03 MUA 43	12	AUM 43	8	2	0	8	
04 SAB #1	13	MUA 42	0	2	0	8	
05 AUM 43	14	SKOK 07	0	2	0	8	
06 ULAZ	07	NSKOK 15	0	2	0	8	
07 NSKOK 15	15	MUA 43	8	2	0	8	
08 ODU #1	16	IZLAZ	8	2	0	8	8
09 AUM 42	17	KRAJ	8	2	0	8	8
10 MUA 43							
11 MNO 41							
12 AUM 43							
13 MUA 42							
14 SKOK 07							
15 MUA 43							
16 IZLAZ							
17 KRAJ							