

## RETRO KUTAK

### Lekcija četvrta

Dobrodošli na četvrti čas "Male škole programiranja na Commodore 64 u BASIC-u". Danas ćemo nastaviti sa upoznavanjem mogućnosti i naredbi C64 računara. Naravno kao i obično daćemo neke savete i primere za ono što učimo.

### Počnimo sa malim savetima i objašnjenjima

#### Kako doći do zvuka iz bejzika C64

Kao što smo u prethodnoj lekciji objasnili ne postoje naredbe za zvuk kod standardnog bejzika za C64. Da bi smo došli do istog moramo manipulirati vrednostima memorijske mape koje su zadužene za to. C64 poseduje trokanalni zvuk (u isto vreme čujemo tri različita zvuka) i memorijska mapa koja obuhvata sva tri kanala kreće se u opseku od 54272 do 54300. Da bi smo dobili zadržavajuću muziku ili neverovatne zvučne efekte treba imati muzičkog iskustva, iskustva sa radom na muzičkim sintisajzerima, znati koje oblike zvuka C64 podržava, odrediti rastuće, zadržavajuće, opadajuće vrednosti tona i mnogo toga. Takođe nije moguće dobiti zvuk nekog posebnog kvaliteta putem bejzika zbog njegove velike sporosti posebno u odnosu na mašinar. Mi ćemo se zadržati na najosnovnijem objašnjenju.

Pa da krenemo. U početku ćemo se objasniti kako treba podesiti prvi kanal koja će se podešavanja primeniti i na drugi i treći. Da bi smo resetovali zvuk i omogućili da isti čujemo moramo prvo izbrisati (postaviti na nulu) sve memorijske lokacije povezane sa zvukom (na isti način prekidamo zvuk). To ćemo postići jednostavnom petljom:

```
10 FOR I=54272 TO 54300:POKE I,0:NEXT I
```

Zatim ćemo uneti vrednosti u sledeće memorijske lokacije:

```
20 POKE 54296,15 (jačina zvuka do maksima - od 0 nečujno do 15 najglasnije važi  
za sva tri kanala)
```

Kod prvog kanala podesićemo sledeće:

```
30 POKE 54278,240 (aktiviramo stalno održavanje zvuka - setovanjem 4 viša bita dobijamo  
vrednost 240. Zvuk se najglasnije i konstantno čuje)
```

```
40 POKE 54276,17 (podesićemo oblik zvuka na triangl setovanjem petog bita (16) i također  
setovati prvi bit za aktiviranje zuka ukupno (17), ako želimo testerasti oblik  
(33),za pulsni oblik (65),a za šum (129). Slobodno ekperimentišite dobićete  
zvukove različitih instrumenata ili zanimljivih zvučnih efekata, probajte  
manipulaciju i prvih četiri bita, biće vam zabavno)
```

```
50 POKE 54272,100: POKE 54273,20 (niži i viši bajt visine tona - dva osmobitna registra koja daju  
ukupno 65535 kombinacija - najbitnije stavke jer ovde  
dobijamo notu koju želimo u rang 8 oktava. Naravno prvi  
bajt na 54272 ima mali uticaj na visinu tona dok drugi na  
54273 čini pravu promenu)
```

Sada za drugi kanal isto sve to samo memorijske adrese uvećavamo za 7 u odnosu na prvi kanal i povećajmo visinu tona na 50

```
100 POKE 54285,240
```

```
110 POKE 54283,17
```

```
120 POKE 54279,100: POKE 54280,50
```

Kod trećeg kanala samo uvećamo memorijske adrese za 14 u odnosu na prvi kanal i povećajmo visinu tona na 70  
200 POKE 54292,240  
210 POKE 54290,17  
220 POKE 54286,100: POKE 54287,70

Naravno primetili ste da memorijske adrese date u primeru nisu redosledne i da neke fale. Nedostajuće adrese služe za još finija podešavanja karakteristika zvuka ali zadržimo se na ovom objašnjenju.

Kada startujemo program dobićemo zvuk na sva tri kanala različite visine koji će se neprekidno čuti. Da bi smo zaustavili ovu kakofoniju dovoljno je isprazniti sve parametre zvuka kao i što smo uradili na početku sa petljom:

```
FOR I=54272 TO 54300:POKE I,0:NEXT I
```

Imajte u vidu da uvek pre nego što želite da radite sa zvukom trebate putem ove petlje resetovati memorijske lokacije ili zvuk koji dobijete nećete moći pravilno da kontolishete ili ga jednostavno neće ni biti.

## **Krenimo sa današnjom lekcijom**

### **Danas ćemo naučiti nekoliko novih funkcija**

#### **ABS - Apsolutna vrednost**

Format funkcije je:

ABS (A)

Ova funkcija daje absolutnu vrednost argumenta "A" bilo broja ili promenljive. Sve negativne vrednosti postaju pozitivne a pozitivne se ne menjaju. Jednom rečju dobijamo broj bez predznaka

Primer:

```
PRINT ABS(-10),ABS(-5.4),ABS(7)
```

Dobijamo rezultat 10, 5.4, 7

#### **SGN - Signum vrednost**

Format funkcije je:

SGN (A)

Ova funkcija daje signum vrednost argumenta "A" bilo broja ili promenljive. Šta je to signum vrednost? Sve negativne vrednosti daju -1, nula daje 0, a pozitivne vrednosti 1

Primer:

```
PRINT SGN(-10),ABS(0),ABS(7)
```

Dobijamo rezultat -1, 0, 1

### **Poradimo malo i na logičkim operacijama**

Logičke operacije služe kod ispitivanja da bi se moglo povezati više međusobno zavisnih uslova. Na primer treba nešto uraditi ako je rezultat prvog ispitivanja promenljive X jednak nuli ali takođe mora da bude zadovoljen uslov da je i promenljiva Y jednaka nuli. Ili je potrebno da se nešto desi ako je bilo koja promenljiva X ili Y različita od nekog zadatog broja itd. Ovo važi i za tekstualne promenljive gde se upoređuju tekstovi

#### **AND - logičko množenje (I)**

Formati operacije:

X uslov AND Y uslov

X\$ uslov AND Y\$ uslov

X\$ uslov AND Y\$ uslov AND X uslov AND Y uslov

Ova logička operacija koristi se kada vršimo ispitivanje da li zadate promenljive odgovaraju uslovima koji svi moraju da budu zadovoljeni da bi odradili željenu stvar.

Primer:

```
IF X=0 AND Y=0 THEN PRINT "PROMENLJIVE X I Y IMAJU VREDNOST NULA"
```

Takođe ova naredba može izvršiti logičko množenje nad binarnim vrednostima dva broja ili dve numeričke promenljive. (Npr.broj 3 je binarno dat kao 11 a broj 5 kao 101). Kada upotrebimo logičko množenje dobijamo sledeće

```
0011
AND 0101 (0*0=0, 0*1=0, 1*0=0, 1*1=1)
-----
0001 što predstavlja decimalnu vrednost 1
```

### **OR - logičko sabiranje (ILI)**

Formati operacije:

X uslov OR Y uslov

X\$ uslov OR Y\$ uslov

X\$ uslov OR Y\$ uslov OR X uslov OR Y uslov

Ova logička operacija koristi se kada vršimo ispitivanje da li zadate promenljive odgovaraju uslovima od kojih makar jedan mora da budu zadovoljeni da bi odradili željenu stvar.

Primer:

```
IF X=0 OR Y=0 THEN PRINT "JEDNA OD PROMENLJIVE X ILI Y IMA VREDNOST NULA"
```

Takođe ova naredba može izvršiti logičko sabiranje nad binarnim vrednostima dva broja ili numeričke dve promenljive. (Uzećemo gornji primer gde je broj 3 binarno dat kao 11 a broj 5 kao 101). Kada upotrebimo logičko sabiranje dobijamo sledeće

```
0011
OR 0101 (0+0=0, 0+1=1, 1+0=1, 1+1=1-da ne bude zabune ovo je logičko a ne stvarno sabiranje)
-----
0111 što predstavlja decimalnu vrednost 7
```

### **NOT - logička negacija (NE)**

Format operacije:

NOT uslov

Ova logička operacija koristi se za promenu tačnosti uslova i radi za razliku od prethodnih operacija samo sa jednim

Primer:

```
IF NOT A<>B THEN PRINT "PROMENLJIVA A NIJE RAZLICITA OD PROMENLJIVE B
ZNACI DA JE JEDNAKA"
```

### **Vreme je za nove naredbe i primere.**

### **DIM - rezervisanje prostora u memoriji za smeštanje višedimenzionalnih numeričkih i tekstuanih promenljivi**

Formati naredbe su:

DIM A(X) - jednodimenziona matrica numeričkih promenljivih

DIM A(X,Y,Z,V, ...) - višedimenzionalna matrica numeričkih promenljivih

DIM A\$(X) - jednodimenziona matrica tekstualnih promenljivih

DIM A\$(X,Y,Z,V, ...) - višedimenzionalna matrica tekstualnih promenljivih

Ova naredba definiše jednodimenzionu ili višedimenzionu matricu numeričkih ili tekstualnih promenljivih. Matrica može imati više argumenata i na taj način možemo napraviti virtualni prostor koji ima dužinu, širinu, visinu, čak i vreme. Izuzetno korisna naredba za razne matrične proračune i ispitivanja ili kod tekstualnih igrica koje zahtevaju kretanje kod prostor i dimenzije.

Format naredbe:

DIM A(4) - formiraće niz od A(0),A(1),A(2),A(3),A(4) i u svaku od ovako formiranih matricnih numeričkih promenljiva možemo upisati neki broj

DIM A\$(3,3) - formiraće niz od A\$(0,0),A\$(0,1),A\$(0,2),A\$(0,3),A\$(1,0),A\$(1,1),A\$(1,2),A\$(1,3), A\$(3,0),A\$(3,1),A\$(3,2),A\$(3,3) i u svaku od ovako formiranih matricnih tekstualnih promenljiva možemo upisati neki proizvoljni tekst/string

Prilikom korišćenja ove naredbe moramo da imamo u vidu da rezervisanje memorije na ovaj način zauzima veliku količinu iste tako da ne treba preterivati sa veličinom matrice ili ako koristimo u programu više matrica. Dodeljivanje vrednosti ovim matricnim promenljivama vrši se na isti način kao i kod običnih promenljivih.

Primer:

```
10 DIM A(10)
20 FOR I=1 TO 10
30 R=INT(RND(0)*30)+1
40 FOR M=1 TO 10
50 IF A(M)=R THEN GOTO 30
60 NEXT M
70 A(I)=R
80 PRINT R,
90 NEXT I
```

Šta ovde imamo? U programskom redu 10 definišemo jednodimenzionu matricu koja sadrži 10 elemenata od A(1),...do A(10). Linija 20 pokreće petlju od 10 ciklusa. Linija 30 generiše slučajni broj od 1 do 30 i vrednost smešta u promenljivu R. Linija 40 pokreće drugu petlju od takođe 10 ciklusa. Ova petlja će uzeti svaki element u nizu koji se formira svakim prolazom u glavnoj petlji i proveriti. Linija 50 uzima svoju promenljivu M za argument matrice i ispituje da li neka od promenljiva u nizu sadrži upravo generisani slučajni broj koji se nalazi u promenljivi R i ako sadrži vraća programski tok u liniju 30 gde se generiše drugi slučajni broj. Linija 60 zatvara drugu petlju koja vrši ispitivanje da li postoji duplikat broja. Ako nema duplikata u liniji 70 se dodeljuje slučajni broj u matricu čiji je argument promenljiva I iz glavne petlje i koja će se ispitivati u programskim linijama 50,60 i 70. Linija 80 štampa rezultat promenljive R. Linija 90 zatvara glavnu petlju.

Startovanjem programa dobijamo deset različitih brojeva koji se nikad ne ponavljaju.

### **GOSUB, RETURN - rad sa podprogramima**

Ove naredbe idu uvek zajedno

Format naredbi:

GOSUB broj programskog reda

RETURN

Ukoliko u programu imamo potrebu za istim stalnim ispitivanjima, proračunima ili prikazima ove naredbe drastično štede vreme pisanja programa i količine memorije koja se troši. Dovoljno je da programer napravi grupu naredbi počev od neke udaljene programske linije (kod manjih programa npr.od programske linije 100 a kod većih počev od 1000,10000 i sl.) koje će stalno pozivati, a kada se unete naredbe izvrše naredba RETURN vratiće programski tok na glavni program od sledeće naredbe koja dolazi posle naredbe GOSUB. Imajte u vidu da se glavni program mora završiti naredbom END ili beskonačnom petljom inače će doći do podprograma, izvršiti ga i kada dođe do naredbe RETURN koja nije pozvana naredbom GOSUB doći će do ispisa greške ?RETURN WITHOUT GOSUB ERROR.

Primer:

```
10 FOR I=1 TO 10
20 PRINT "UNESITE BROJ KOJI ZELITE DA STEPENUJETE"
30 INPUT B
40 GOSUB 100
```

```

50 PRINT "REZULTAT STEPENOVANJA SA 2 JE ";S
60 PRINT "REZULTAT STEPENOVANJA SA 3 JE ";K
70 END
100 S=B↑2: K=B↑3
110 RETURN

```

A sada poslastica dana napravimo jednu malu tekstualnu igricu. Šta kažete da zalutamo u 81 soba i u jednoj da bude izlaz. Imamo za indikatore da smo blizu izlazu "Toplo","Vruće" i "Vrelo". Koristićemo dimenzionu matricu 9x9 koju ćemo ograničiti po X i Y osi od 1 do 9. Tako nam ostaje neiskorišćene lokacije X i Y za vrednost 0 (Ovo je samo zbog primera koji ukazuje da zadatu matricu možemo iskoristiti i ograničiti po svojim potrebama). Takođe napravićemo da kretanje bude ciklično. To znači da kada dođemo do poslednje odnosno prve sobe bilo po kojoj osi pređemo u prvu odnosno poslednju sobu po datoj osi (npr.iz 79 ide se na sobu 19, ili iz 81 na 89 itd). Na kraju ćemo dodati po neki video i zvučni efekat. Ovaj program ćemo objašnjavati korak po korak

```
10 PRINT "(SHIFT+CTRL/HOME)"
```

Briše se ekran

```
20 DIM A$(9,9)
```

Definišemo tekstualnu matricu veličine 9x9

```
30 GOSUB 1000
```

Odlazimo na podprogram u programskoj liniji 1000 koji generiše za X i Y promenljive (kordinate soba) slučajne brojeve od 1 do 9

```
40 IF X<2 OR X>8 OR Y<2 OR Y>8 THEN GOTO 30
```

Ovde ograničavamo X i Y kordinatu da ne mogu biti manje od 2 odnosno veće od 8. Ukoliko jeste idemo na liniju 30 gde se ponovo generišu slučajnu brojevi. Ovo ograničenje je potrebno zbog kasnijih ispitivanja gde se izlaz nalazi.

```
50 A$(X,Y)="IZLAZ"
```

U matricu slučajno generisanih kordinata smeštamo tekst "IZLAZ" koju ćemo kasnije ispitivati da li je uslov zadovoljen i da li smo izašli iz soba.

```
60 XI=X:YI=Y:PO=0
```

Smeštamo u promenljivama XI i YI kordinate koje su slučajno generisane (ovo će biti kordinate izlaza) i koje ćemo kasnije upoređivati sa tekućim kordinatama sobe u koje se trenutno nalazimo

```
65 REM PRINT XI,YI
```

Ovaj red nije aktivan a ako ga aktivirate (obrišete REM naredbu) videćete kordinate izlaza. Služi za testiranje programa

```
70 PRINT "(CTRL+2)":POKE 53280,2:POKE 53281,8
```

Ovde podešavamo boju karaktera(bela),boju okvira (crvena),boju pozadina (braon). Naravno boje možete podesiti po svom ukusu

```
80 PRINT "DOBRODOSLI"
```

Dobrodošlica što da ne?

```
90 GOSUB 1000:IF X=XI AND Y=YI THEN GOTO 90
```

Ponovni odlazak na podprogram za generisanje slučajnih X i Y kordinata. Ove kordinate biće kordinate sobe u koje se nalazimo na početku igre. Iako je šansa mala ispitujemo da li su te kordinate identične sa kordinatama izlaza a ako jesu ponovo se generišu. Kordinate ne smeju da budu identične inače ste već u sobi gde je izlaz i gde je tu zabava?

```
100 PRINT "NALAZITE SE U SOBI",X;Y
```

```
110 PRINT "MOGUCE KOMANDE:W-GORE,S-DOLE,A-LEVO,D-DESNO"
```

```
120 PRINT "UNESITE U KOM SMERU ZELITE DA IDETE"
```

Ova tri programska reda ispusuju malo uputstvo šta dalje.Za kretanje sobama koristimo slova W-gore, S-dole, A-levo, D-desno (stvar navike kod današnjih igrača)

130 IN=0

Promenljivu koja ima ulogu indikatora IN postavljamo na nulu. Pomoću nje ćemo znati da li je komanda koju unosimo dozvoljena ili ne

140 INPUT SM\$

Unosimo komandu o kretanju u promenljivu SM\$

150 GOSUB 1100

Odlazimo na podprogram u programskoj liniji 1100 koji ispituje da li SM\$ sadrži ispravnu komandu i ako sadrži omogućava da se krećemo po sobama. Takođe setuje indikator IN na 1 ukoliko je komanda koju smo uneli ispravna

160 IF IN=0 THEN PRINT "POGRESNA KOMANDA":GOTO 140

Ukoliko je komanda koju smo uneli ispravna indikator IN je 1. Ukoliko ima vrednost 0 nije i vraćamo se na programsku liniju 140 gde unosimo novu komandu

170 GOSUB 1200

Odlazimo na podprogram gde utvđujemo koliko smo daleko od izlaza. Ukoliko smo udaljeni od izlaza manje od 4 po X ili po Y kordinati ispisuje se poruka "Toplo", za manje od 3 poruka je "Vruće" a za manje od 2 "Vrelo". Ovde možemo po našoj želji povećati veličinu pretrage povećanjem broja za ispitivanje udaljenosti.

180 PO=PO+1

Uvećavamo promenljivu koja sadrži broj pokušaja za 1

190 IF A\$(X,Y)="IZLAZ" THEN GOTO 1300

Ukoliko smo uspeali da dođemo do kordinata sobe u kojoj matrična promenljiva sadrži tekst "Izlaz" to je to POBEDILI SMO. Odlazimo na programsku liniju 1300 od koje ćemo nadalje dobiti poruku o pobedi, imati malo ludovanje boja i čuće se rastući zvuk. Ovim završavamo program.

200 PRINT "(CTRL+2)":GOTO 100

U ovom programskom redu bojimo kurzor u belo (ako mu se u međuvremenu promenila boja) i vraćamo se na programski red 100 gde se ponovo ispusuje kordinate soba u kojoj se nalazimo, traži nova komanda i tako u krug dok ne dođemo do izlaza

990 END

Završavamo glavni program mada programski tok i onako nikada neće doći do ovog programskog reda ali uradimo to školski

1000 X=INT(RND(0)\*9)+1

Krenuli smo sa prvim podprogramom. Ovde se generiše slučajni broj za X kordinatu sobe od 1 do 9

1010 Y=INT(RND(0)\*9)+1

Ovde generišemo slučajni broj za Y kordinatu sobe od 1 do 9

1020 RETURN

Vraćamo se na glavni tok programa posle naredbe GOSUB koja je pozvala ovaj podprogram

1100 IF SM\$="D" THEN X=X+1:IN=1

Krećemo sa drugim podprogramom. Ovde ispitujemo da li promenljiva SM\$ sadrži komandu "D" koja uvećava X kordinatu i koja će nas pokrenuti desno u sobama (primer 34 na 44). Indikator IN postaje jedinica što znači da smo uneli ispravnu komadnu

1110 IF X>9 THEN X=1

U slučaju da je kordinata X veća od 9 upisujemo za kordinatu X vrednost 1 (ciklično kretanje)

1120 IF SM\$="A" THEN X=X-1:IN=1

Ovde ispitujemo da li promenljiva SM\$ sadrži komandu "A" koja umanjuje X kordinatu i koja će nas pokrenuti levo u sobama (primer 34 na 24). Indikator IN postaje jedinica

1130 IF X<1 THEN X=9

U slučaju da je kordinata X manja od 1 upisujemo za kordinatu X vrednost 9

1140 IF SM\$="S" THEN Y=Y-1:IN=1

Ispitujemo da li promenljiva SM\$ sadrži komandu "S" koja umanjuje Y kordinatu i koja će nas pokrenuti dole u sobama (primer 34 na 33). Indikator IN postaje jedinica

1150 IF Y<1 THEN Y=9

U slučaju da je kordinata Y manja od 1 upisujemo za kordinatu Y vrednost 9

1160 IF SM\$="W" THEN Y=Y+1:IN=1

Ispitujemo da li promenljiva SM\$ sadrži komandu "W" koja uvećava Y kordinatu i koja će nas pokrenuti na gore u sobama (primer 34 na 35). Indikator IN postaje jedinica

1170 IF Y>9 THEN Y=1

U slučaju da je kordinata Y veća od 9 upisujemo za kordinatu Y vrednost 1

1180 RETURN

Vraćamo se na glavni tok programa posle naredbe GOSUB koja je pozvala ovaj podprogram

1200 XP=ABS(XI-X)

Evo nas i u trećem podprogramu. Ovde defenišemo promenljivu XP koja nam kazuje koliko smo udaljeni od izlaza po X osi (XI je kordinata izlaza a X trenutna kordinata sobe u koje se nalazimo. Njihovim oduzimanjem dobijamo trenutnu udaljenost po X kordinati). Koristimo naredbu ABS koja će nam dati uvek pozitivnu vrednost (bez obzira na pravac udaljenosti) koju ćemo kasnije uporediti

1210 YP=ABS(YI-Y)

Sve ovo važi i za Y kordinatu. Dakle defenišemo promenljivu YP koja nam kazuje koliko smo udaljeni od izlaza po Y osi (YI je kordinata izlaza a Y trenutna kordinata sobe u koje se nalazimo. Njihovim oduzimanjem dobijamo trenutnu udaljenost po Y kordinati). Također pomoću ABS naredbe dobijamo uvek pozitivnu vrednost (bez obzira na pravac udaljenosti)

1215 REM PRINT "XP";XP;"YP";YP;"X";X;"Y";Y

Ovaj red je neaktivan. Ukoliko ga aktivirate značete tačno koliko ste udaljeni od izlaza po X i Y kordinati. Služi za testiranje

1220 IF XP<2 AND YP<2 THEN PRINT "(CTRL+6)VRELO":GOTO 1280

Ovde vršimo ispitivanje koliko smo udaljeni od izlaza. Ukoliko je manje od 2 po X i Y kordinati onda smo veoma blizu i dobijamo poruku "Vrelo" ispisano zelenim. Onda preskačemo ostala ispitivanja jer i ona bi zadovoljila uslove a nama treba samo jedna poruka i idemo na liniju koja sadrži naredbu RETURN (kraj podprograma)

1230 IF XP<3 AND YP<3 THEN PRINT "(CTRL+8)VRUCE":GOTO 1280

Za poruku "Vruće" ispisano žutim treba da budemo manje od 3 po X i Y kordinati. Idemo na kraj podprograma

1240 IF XP<4 AND YP<4 THEN PRINT "(CTRL+3)TOPLO"

Za poruku "Toplo" ispisano crvenim treba da budemo manje od 4 po X i Y kordinati. Ovde nije potrebno ići na kraj pošto je ovo zadnja programska linija u ovom podprogramu

1250 RETURN

Naravno vraćamo se na prvu naredbu posle GOSUB naredbe koja je pozvala ovaj podprogram

1300 PRINT:PRINT "BRAVO USPELI STE IZ";PO;"POKUSAJA"

Na kraju kada smo u glavnom programu uspeli da nađemo kordinate tekstualne matrice koja sadrži tekst "IZLAZ" odlazimo na ovu programska liniju i dobijamo poruku o pobedi i iz koliko pokušaja smo uspeli

1310 FOR I=54272 TO 54300:POKE I,0:NEXT I

Ovde brišemo sve memorijske adrese zadužene za zvuk i samim tim pripremamo računar sa zvučne efekte. Koristićemo samo prvi kanal

1320 POKE 54296,15

Jačina tona do kraja

1330 POKE 54278,240

Trajanje zvuka neprekidno

1340 POKE 54276,17

Oblik zvuka triangl

1350 FOR I=1 TO 50

Postavljamo petlju za pedeset ciklusa

1360 POKE 54272,124: POKE 54273,19+I

Povećavamo u svakom prolazu visinu tona

1370 POKE 53280,RND(0)\*16

Slučajna boja okvira

```
1380 POKE 53281,RND(0)*16
```

Slučajna boja pozadine

```
1390 NEXT I
```

Kraj petlje

```
1400 FOR I=54272 TO 54300:POKE I,0:NEXT I
```

Prekid zvuka

```
1410 PRINT:PRINT "NASLI STE IZLAZ NA";X;Y
```

Ispis poruke kordinata gde ste našli izlaz

```
1420 POKE 53280,14:POKE 53281,6
```

```
1430 PRINT "(COMMODORE+7)"
```

U ova dva reda vraćamo boje karaktera.okvira i pozadine na podrazumevane vrednosti

```
1440 END
```

I na kraju END. Ova naredba nije potrebna jer je programu kraj ali završimo ga u stilu

Uzmite ovaj program kao početni primer. Pokušajte da u njemu dodate sobe sa npr.nekim zamkama ili nagradama. Ukoliko upadnete u zamku dajte da program ispiše poruku o neuspehu možda i neki prateći blesak boja i zvučni efekat. Što se tiče nagrada možda na kraju uspešnog izlaza to bude broj skupljenih zlatnika po sobama ili slično. Možete dodati i više soba ili sprat na koji ćete se popeti lestvicama ili liftom itd. (matrica bi sadržala tri argumenta npr.A\$(X,Y,Z)). Možda dodate na kraju da vas računar pita želite li novu igru. Najvažnije je da budete kreativni.

Ovim završavamo za danas.

Samo napred. Uživajte u računarskim retro vodama