

G R A F 0 9

U v o d v B A S I C

NUMERIČNE SPREMENLJIVKE

Kot numerično spremenljivko lahko shranimo katerokoli število.

Ime spremenljivke je lahko sestavljeno iz poljubne kombinacije črk in števil, prvi znak pa mora biti črka.

Računalnik uporablja le prva dva znaka imena, zato imen-ko so

LUKA, LUMEN, LU2

ne razlikuje. Uporaba imen z več črkami pa nam olajša razumevanje programa, če za imena uporabljamo ustaljena imena ali pojme, ki opisujejo spremenljivko. Ime VSOTA nam nedvomno bolje pojasni, kaj je shranjeno pod tem imenom, kot pa VS.

Računalnik daljša imena sprejme, vendar pa uporabimo na ta način nekaj več prostora v pomnilniku, kot če uporabljamo kratka imena.

ZNAKOVNE SPREMENLJIVKE

Kot znakovne spremenljivke lahko shranjujemo zaporedja alfanumeričnih znakov. Tako zaporedje pogosto imenujemo "STRING".

Kot ime znakovne spremenljivke lahko uporabimo poljubno kombinacijo črk in števil, vendar mora biti prvi znak črka, zadnji znak pa \$.

Enako kot pri numeričnih spremenljivkah tudi tu računalnik uporabi le prva dva znaka imena, zato sprejme

ANTON\$, AN1\$ in AN2\$

kot eno samo spremenljivko AN\$.

LIST

Ukaz LIST nam izpiše na ekran program, ki je shranjen v pomnilniku.

Ker je program običajno predolg, da bi ga v celoti izpisali na ekranu, lahko izpisovanje prekinemo s pritiskom na (SHIFT)(). Izpisovanje se nadaljuje, če pritisnemo katerokoli tipko. Pri tem delu moramo biti hitri.

Da bi izpisali le del programa uporabimo :

LIST n_1 - n_2

n_1 in n_2 sta številki prve in zadnje vrstice, ki naj se izpiše. Primer:

LIST 40-100

izpiše vrstice med 40 in 100

LIST - 80

izpiše vse vrstice od začetka programa do 80

LIST 120 -

izpiše vse vrstice od 120 do konca programa.

RUN

Da bi sprožili izvajanje programa vtipkamo ukaz RUN.

Če ne želimo, da bi se program izvajal od začetka, vtipkamo

RUN n

n je številka vrstice, kjer naj se izvajanje prične

RUN 250

NEW

NEW je ukaz, s katerim brišemo vso vsebino pomnilnika.

Priporočamo, da uporabite NEW vedno, predno vnašale nov program. Na ta način preprečite, da bi ostanki starega programa ovirali izvajanje novega.

LET

Prireditveni ukaz LET uporabimo, da bi spremenljivki dali zaželeno vrednost. Ukaz ima naslednjo obliko :

LET ime spremenljivke = izraz

Ukaz LET je običajni sestavni del BASIC-a, vendar na računalniku GRAF 09 tega ukaza ne uporabljamo, ker ni potreben.

Spremenljivka, ki ji prirejamo vrednost, je definirana z imenom.

Izraz je konstanta, druga spremenljivka ali njihova kombinacija, ki jo povezujejo operatorji (+, -, *, ...). Ker operacije med numenričnimi in znakovnimi spremenljivkami niso možne, morata biti spremenljivka in izraz istorodna.

40 X=X+1

vrednost X se poveča za ena.

10 S=0:N=0:CLS 5

20 PRINT@72,"VTIPKAJ ŠTEVILO";

30 INPUT X:CLS 5

40 S=S+X:N=N+1

50 PRINT@192,"VTIPKAL SI";N;"ŠTEVIL";

60 PRINT@262,"POVPREČNA VREDNOST JE";S/N

70 GOTO 20

PRINT

Z ukazom PRINT izpišemo vrednosti spremenljivk, konstant ali poljubne tekste na ekran. Uporabimo ga tudi za neposredno računanje brez programa.

Če v enem PRINT stavku navedemo več kot en izraz, jih moramo ločiti z vejico (,) ali podpičjem (;).

Vejica povzroči, da se izpis oblikuje v dveh stolpcih širine 15 znakov. Če je prvi izraz daljši od 15 znakov, se kljub temu v celoti izpiše, sledeči izraz pa se izpiše v novi vrsti.

Podpiče povzroči, da se izrazi izpisujejo eden za drugim, loči jih eno prazno mesto.

Stavek PRINT brez navedb izpiše prazno vrstico.

Primeri :

10 CLS

20 PRINT " PRIMERI STAVKA PRINT "

30 PRINT

40 PRINT "PRVI STOLPEC"; "DRUGI STOLPEC"

50 PRINT 142.6; 13.76

60 PRINT 1,2,3,11,34

70 PRINT "TO JE";

80 PRINT "KONEC PRIMERA"

INPUT

Ko pride program do stavka INPUT se ustavi in čaka na podatke, vnešene preko tastature. Stavek INPUT mora vsebovati eno ali več spremenljivk, ki so ločene z vejicami.

```
25 INPUT A,B,FS,H7
```

Gornji stavek zahteva, da vnesemo štiri podatke. To lahko storite tako, da po vsakem podatku pritisnete tipko ENTER, ali pa odtipkate vse štiri vrednosti, ločene z vejicami, kot kaže spodnji primer :

```
146.5,23,ANA,3 (ENTER)
```

Vnašanje podatkov olajšamo, če operaterja opozorimo, kaj mora vtiskati. To lahko storimo s stavkom PRINT ali pa tako, da navodilo vgradimo v stavek INPUT. Navodilo v stavku INPUT ima naslednjo obliko :

```
35 INPUT "VTIPKAJ POLJUBNI ŠTEVILI";N1,N2
```

Navodilo in seznam spremenljivk ločuje podpičje.

Paziti moramo, da imajo vnešeni podatki pravilno obliko : za numerične spremenljivke vnašamo števila, za znakovne spremenljivke pa stringe, sicer se izvajanje programa ustavi, na ekranu pa se izpiše ?REDO, nakar moramo vnos ponoviti brez napake.

Pri vnosu stringov ni nujno, da jih tipkamo v narekovajih. Obe obliki STRING in "STRING" sta možni.

RND

Funkcija RND daje naključna števila.

Funkcija v BASIC-u uporabi eno ali več vrednosti in z njimi izvede računanje, katerega rezultat je ena sama vrednost. Vrednosti (spremenljivke) ki se uporabijo v računanju imenujemo argumenti in jih navajamo v okepaju za imenom funkcije.

Funkcija RND izračuna naključno število, odvisno od vrednosti argumenta. Če je argument enak \emptyset (RND(\emptyset)), se izračuna naključno število v intervalu od 1 do \emptyset .

Če je argument večji od \emptyset (primer : RND(6)), izračuna funkcija celo število med 1 in vrednostjo argumenta. Ker je rezultat naključno število ne moremo vnaprej vedeti, kakšen bo rezultat, oz. katera od možnih vrednosti bo izbrana .

```

10 CLS
20 PRINT@ 8,"VTPKAJ ENO ŠTEVILO";:INPUT N
30 CLS:PRINT@ 194,"TRI NAKLJUČNA ŠTEVILA ZA N=";N
40 PRINT@ 270,RND(N)
50 PRINT@ 302,RND(N)
60 PRINT@ 334,RND(N)
70 GOTO 20

```

CLS

Ukaz CLS uporabimo za brisanje ekrana in izbiranje barve ozadja. Normalna barva ozadja je zelena in je ni potrebno navajati. Vtikamo samo CLS.

Če želimo drugačno ozadje, dodamo ukazu CLS še številko med 0 in 8, ki označuje eno naslednjih barv :

0 - črna	1 - zelena	2 - rumena
3 - modra	4 - rdeča	5 - bela
6 - tirkizna (cian)	7 - vijolična	8 - oranžna

Barve lahko izberemo le, če je na računalnik priključen barvni televizor. Odtenek barve je odvisen od nastavitve na televizorju.

Neodvisno od barve ozadja je tekst v črni ali zeleni barvi.

PRINT @

Ukaz PRINT @ uporabimo, da bi izpisali vsebino ukaza na izbrano mesto na ekranu.

Ekran moramo opazovati kot raster z 16 x 32 polji. Vseh polj je 512, oštevilčena so z 0 do 511. Oštevilčba je prikazana na naslednji strani.

PRINT@ ukaz ima naslednjo obliko :

PRINT@ izraz, seznam spremenljivk

Izraz je lahko število, spremenljivka ali aritmetična formula. Vrednost izraza mora biti med 0 in 511.

Seznam spremenljivk ima enako obliko kot pri ukazu PRINT. Vsebuje števila, spremenljivke, tekste ali formule, medsebojno ločene z vejicami ali podpičji.

Če vzamemo, da je ekran sestavljen iz 16. vrstic, potem da

PRINT@ 32*(LINE - 1),A

izpiše vrednost A na začetku vrstice. Izbira vrstice je odvisna od vrednosti LINE, ki je med 1 in 16.

PRINT USING

Z ukazom PRINT USING lahko določamo format izpisa na ekranu ali tiskalniku. Ukaz ima obliko :

PRINT USING format;seznam spremenljivk

format je ime znakovne konstante ali spremenljivke, ki vsebuje formatne podatke za oblikovanje izpisa. Seznam spremenljivk vsebuje numerične in znakovne spremenljivke, medsebojno ločene z vejicami. Uporabljamo naslednje formatne podatke :

Znak	Pomen	primer	rezultat
#	da položaj številke glede na decimalno piko	"###";147.2	147
.	da položaj decimalne pike	"##.##"34.678	34.67
,	izpiše vejico po vsaki tretji številki	"#####"; 123456	123,456
**	vodilne ničle se namesto z praznimi mesti izpišejo z zvezdicami	"*##.##"; 1.47	*1.4
\$	postavi znak \$ pred število	"\$##.##";12.69	\$12.69
+	če stoji + kot prvi znak v formatu se predznak izpiše pred vrednostjo, če stoji kot zadnji pa za vrednostjo	"+#####"; -12.689	-12.68
↑↑↑↑	izpis z mantiso in potenco števila 10	"###.##↑↑↑↑"; 12.689	12.68- 1.27E+01
!	izpiše se le prvi znak stringa	"!";KREDIT	K
%▼▼%	da število izpisanih zankov iz stringa. Število je enako številu praznih mest	"%▼%";SALDO	SAL

+2

IZPIS NA PRINTER

Če imamo na paralelne vmesniku priključen tiskalnik lahko ipisujemo na ekran ali na tiskalnik. V ta namen sta nam na voljo naslednja ukaza :

PRINT#-2, seznam spremenljivk

PRINT USING#-2,format;seznam spremenljivk

format in seznam spremenljivk imata enako zgradbo kot pri pisanju na ekran.

POS(-2) da trenutni položaj pisalne glave v vrstici, kjer se je izpisovanje končalo in tiskalnik stoji.

LLIST izpiše program na tiskalnik. Uporablja se anako kot ukaz LIST.

Če smo pri vnašanju uporabili (SHIFT)(Ø), lahko izpisujemo na tiskalniku tudi male črke. Male črke lahko uporabljamo le v stringih in REM stavkih, vsi ukazi za računalnik morajo biti pisani z velikimi črkami.

PRINT - DELOVNI LIST

SOUND

Ukaz SOUND proizvede ton izbrane višine in dolžine. Uporablja dva argumenta :

5Ø SOUND V,D

V je število med 1 in 255. Najnižji ton je 1, najvišji 255. C ključu na klavirju odgovarja vrednost 89.

D je lahko število med 1 in 255. Pri 16 traja ton natančno eno sekundo.

GOTO

Ukaz GOTO ima obliko :

4Ø GOTO n

n je številka vrstice, kjer naj se izvajanje programa nadaljuje. Biti mora število, ne pa spremenljivka. Če ni vrstice z navedeno številko se izvajanje programa prekine, na ekranu pa se pojavi ?UL ERROR (Undefined Line).

Ukaz GOTO se takoj izvede, zato ni smiselno navajanje dodatnih ukazov v isti vrstici, ker se tak ukaz ne izvede .

5Ø GOTO 25:PRINT "KONEC OBDELAVE"

izvede samo skok v 25. vrstico. Če želimo, da se tekst v narekovajih pojavi na ekranu uporabimo naslednjo obliko :

5Ø PRINT "KONEC OBDELAVE":GOTO 25

ali

5Ø PRINT "KONEC OBDELAVE"

6Ø GOTO 25

REM

Ukaz REM uporabljamo za vpisovanje pripomb in obrazložitvev v program, s čemer olajšamo analizo in popraviljanje programa v prihodnosti. Vse navedbe, ki sledijo ukazu REM se pri izvajanju programa izpustijo in nimajo vpliva na rezultate.

Namesto REM lahko uporabimo tudi znak (').

1Ø REM TO JE OSNOVNA OBLIKA UKAZA

2Ø $D = B \cdot B - 4 \cdot A \cdot C$: 'DETERMINANTA

3Ø REM V VRSTICI 2Ø VIDIMO DRUGO OBLIKO UKAZA REM

CSAVE
CLOAD
SKIPF

Z ukazom CSAVE zapišemo program na kaseto. Ime programa sme imeti največ 8 znakov.

CSAVE "IME PROGRAMA"

Za zapisovanje podatkovnih datotek na kaseto dodamo ukazu parameter A. Podatki se bodo zapisali v ASCII kodi. Či-
tamo jih z ukazom INPUT#-1

CSAVE "IME DATOTEKE",A
INPUT#-1, seznam spremenljivk

Ukaz CLOAD prebere program s kasete in ga naloži v pomnilnik.

CLOAD "IME PROGRAMA"

Ukaz SKIPF povzroči previjanje kasete do konca programa, katerega ime je v ukazu :

SKIPF "IME PROGRAMA"

EDIT

Ukaz EDIT uporabimo za popraviljanje izbrane vrstice

EDIT št.vrstice (ENTER)

EDIT 5Ø

S tem ukazom vstopimo v editor, kar nam omogoča, da uporabljamo naslednje ukaze :

L	LIST - vrstica se izpiše
C znak	CHANGE - spremeni znak pod kurzorjem
n C znak	spremeni n naslednjih znakov
I	INSERT - vrine vtipkane znake v obstoječ tekst, insert modus
D	DELETE - briše znak pod kurzorjem
n D	briše naslednjih n znakov
H	HACK - briše vse znake desno od kurzorja in se vrne v insert modus
X	EXTEND - podaljša vrstico na njenem koncu in preskoči v insert modus
S znak	SEARCH - poišče prvi navedeni znak v vrsti
n S znak	poišče n-ti znak
K	KILL - briše ostanek vrstice desno od kurzorja
n K znak	briše znake desno od kurzorja do n-tega znaka, navedenega v ukazu
n (tipka za presledek)	prestavi kurzor za n mest naprej, če n ni podan se premakne za eno mesto
n (←)	prestavi kurzor za n mest nazaj, če n ni naveden se premakne za eno mesto v levo
(SHIFT) (↑)	zapusti insert modus in preide v editor
(ENTER)	zapusti editor, shrani vrstico in izpiše OK na ekranu

DEL

Ukaz DEL uporabljamo za brisanje vrstic shranjenega programa.

DEL št. vrstice 1 - št. vrstice 2

S tem ukazom brišemo vse vrstice s številkami med navedenima števloma, vključno z vrsticama, označenima z št. vrstice 1 in št. vrstice 2

Ukaz ima lahko tudi eno naslednjih oblik :

DEL - briše cel program

DEL - 100 briše program od začetka do vrstice 100

DEL 300 - briše program od vrstice 300 do konca

DEL 40 briše le vrstico 40

DEL 10 - 70 briše vrstice od 10 do 70

RENUM

Ukaz RENUM nam omogoča, da spreminjamo numeracijo vrstic v programu. S tem ukazom se menjajo tudi številke vrstic v ukazih za skoke (GOTO ...). Na ta način je zagotovljeno, da se program brezhibno izvede tudi po preštevilčenju vrstic.

RENUM nova vrstica, začetna vrstica, korak

Ukaz RENUM preštevilči vrstice. Parameter nova vrstica da začetno vrednost nove oštevilčbe. Parameter začetna vrstica številko vrstice v stari oštevilčbi, kjer naj se preštevilčba prične. Parameter korak da razliko med štelkami vrstic v novi oštevilčbi.

RENUM	preštevilči cel program. Vrstice dobe številke 10, 20, 30 ...
RENUM 100	preštevilči cel program. Vrstice dobe številke 100, 110, 120 ...
RENUM 100,50,5	prične od vrstice 50. Vrstice dobe številke 100,105,110,115...
RENUM ,,20	preštevilči se cel program. Vrstice dobe številke 10, 30, 50 ...

TRACE

Izvajanje programa lahko spremljamo s TRACE. Na ekranu se izpiše številka vrstice, ki se izvaja. Trace moramo vključiti pred pričetkom izvajanja programa.

TRON vključi TRACE
TROFF izključi TRACE

Oba ukaza se izvajata neposredno in ju ne moremo vgraditi v program.

END
STOP
CONT

Ukaz END zaključi izvajanje programa, računalnik čaka na nov ukaz s tastature.

Ukaz STOP ustavi izvajanje programa v vrstici, kjer se ta ukaz nahaja. Na ekranu se pojavi opozorilo BREAK AT N. N je številka vrstice. Za nadaljevanje izvajanja moramo vtiskati CONT (CONTINUE).

Po ukazu END izvajanja ne moremo nadaljevati. Program lahko le ponovno startamo.

ON ... GOTO

Ukaz ON...GOTO omogoča pogojni skok na eno od v nadaljevanju navedenih vrstic

ON izraz GOTO seznam št. vrstic

Izraz se izračuna in zaokroži, če ni cela vrednost. Program izbere številko vrstice iz seznama, katere položaj v seznamu ustreza vrednosti izraza. Če je slednja enaka 4, potem se izvede skok v vrstico, katere številka je na 4. mestu v seznamu. Izraz ne sme biti negativen. Če ima izraz vrednost \emptyset ali je njegova vrednost večja od števila številok vrstic v seznamu, se ukaz ignorira in izvajanje se nadaljuje z ukazom v naslednji vrstici.

```

10 CLS : PRINT "RESITEV KVADRATNE ENACBE"
20 INPUT "A,B,C";A,B,C:IF A=0 THEN 20
30 R=-B/(2*A):D=B*B-4*A*C:S=SGN(D)
40 P=SQR(D*S)/(2*A)
50 ON S+2 GOTO 80,60,70
60 PRINT "OBE RESITVI STA ENAKI":PRINT R,R:END
70 PRINT "REALNI REŠITVI":PRINT R+P,R-P:END
80 PRINT "KOMPLEKSNI REŠITVI":PRINT R,R:PRINT P,-P:END

```

IF...THEN...ELSE

Ukaz ima naslednjo obliko :

IF pogoj THEN navodilo 1 ELSE navodilo 2

Ukaz preizkusi pogoj, ki je lahko izpolnjen (TRUE) ali neizpolnjen (FALSE). Če je pogoj izpolnjen se izvede navodilo 1, sicer pa navodilo 2.

Pogoj sestavljata dva izraza, ki ju povezuje operator. Kot izraz lahko uporabimo katerikoli izraz iz BASIC-A, vendar morata imati oba izraza isti tip (numerični ali string). Možni so naslednji operatorji :

=	enako	<>	neenako
>	večje kot	<	manjše
>=	večje ali enako	<=	manjše ali enako

Pogoji so lahko povezani z logičnimi operatorji AND, OR in NOT.

pogoj AND pogoj	TRUE če sta izpolnjena oba pogoja
pogoj OR pogoj	TRUE če je izpolnjen vsaj en pogoj
NOT pogoj	TRUE če pogoj ni izpolnjen

Navodilo 1 in navodilo 2 sta lahko katerikoli izraz iz BASIC-a, vključno z nadaljnimi IF stavki.

ELSE je razširitev ukaza in ga zato lahko izpustimo. V tem primeru bo program nadaljeval izvajanje v naslednji vrstici, če pogoj ne bo izpolnjen.

```

10 CLS:PRINT @ 9,"UGANJEVANJE ŠTEVILA":N=RND(100):T=0
20 INPUT "UGANI ŠTEVILKO";G:T=T+1
30 IF G=N THEN PRINT "UGANIL SO PO";T;"POIZKUSIH:END
40 IF G>N THEN PRINT "PRAVA VREDNOST JE MANJŠA" ELSE PRINT
   "PRAVA VREDNOST JE VEČJA"
50 GOTO 20

```

FOR...NEXT...STEP

Ukaz FOR...NEXT sestavljata dva stavka :

FOR ime=izraz TO izraz STEP izraz

in

NEXT ime

S tem ukazom dosežemo, da se del programa n-krat ponovi. Izrazi, ki so lahko konstante, spremenljivke eli aritmetični izrazi, se izračunajo in števec ponovitev šteje od vrednosti prvega izraza do vrednosti drugega s korakom, ki ga definira izraz za STEP. Trenutna vrednost števca je shranjena kot numerična spremenljivka z imenom, ki je navedeno za FOR in jo lahko uporabimo za računanje v zanki. Zanka se izvede vsaj enkrat, četudi je korak večji od različne vrednosti obeh izrazov. STEP lahko izpustimo, v tem primeru se števec v vsaki ponovitvi poveča za 1.

Zanko lahko kjerkoli prekinemo z ukazi GOTO in IF...THEN.

Uporabljamo lahko tudi več zank v zanki, vendar skok v sredo zanke ni dovoljen.

INKEY\$

Ukaz INKEY\$ je funkcija. Ukaz preveri tastaturo, in ugotovi, katera tipka je pritisnjena. Znak s te tipke se uporabi kot string.

Ukaz uporabljamo za shranjevanje znaka (samo enega) v string. tipke (ENTER) ni treba pritisniti.

```

10 CLS: PRINT@0, "PRITISNI ENO OD TIPK IN ";
20 PRINT@32, "POVEDAL TI BOM, KATERA JE";
30 B$ = "NOBENE TIPKE NISI PRITISNIL ";
40 C$ = "PRITISNIL SI TIPKO ";
50 A$ = INKEY$
60 IF A$ = "" THEN PRINT@192,B$;ELSE PRINT@192,C$;A$;
70 FOR D = 1 TO 600:NEXT D:GOTO 50

```


GOSUB
 RETURN
 ON ... GOSUB

Z ukazom GOSUB dosežemo skok na začetek podprograma , ki se konča z ukazom RETURN. Ta povzroči povratak na vrstico v glavnem programu, ki sledi vrstici z ukazom GOSUB.

Za GOSUB navedemo št. vrstice, v kateri se prične podprogram.

GOSUB 1600

Podprogram ima lahko tudi več ukazov RETURN.

Ukaz ON...GOSUB je pogojni skok v enega od podprogramov. Ukaz lahko po strukturi primerjamo z ukazom ON...GOTO.

ON-izraz GOSUB seznam št. vrstic

Če je izraz negativen nam računalnik javi napako. Če je enak 0 ali večji od števila elementov v seznamu se ukaz ignorira, izvajanje se nadaljuje v naslednji vrstici

DIM

Ukaz DIM uporabimo za definiranje polja spremenljivke. Polje je lahko eno ali dvodimenzionalno, numerično ali iz znakovnih spremenljivk. Stavek ima obliko :

DIM ime polja (n), ime polja (n,n)

Ukaz DIM ni potreben, če je polje manjše od 10 elementov.

Če želimo uporabiti element takega polja moramo poleg imena navesti v oklepaju še indeks.

10 DIM A(6,6)

20 FOR I=1 TO 6

30 FOR J=1 TO 6: A(I,J)=I+J: NEXT J

40 NEXT I

DEF FN

Z ukazom DEF FN lahko definiramo novo numenrično funkcijo.

DEF FN ime(spremenljivka)=formula

ime je lahko katerakoli črka od A do Z.

spremenljivka je lahko katerakoli črka od A do Z. Tekom izvajanja se nadomesti z argumentom, ko kličemo funkcijo. Uporabimo lahko le eno spremenljivko.

formula definira operacije, v katerih se uporablja spremenljivka in/ali druge spremenljivke.

DEF FN lahko uporabi eno programsko vrstico. V formuli so lahko uporabljene funkcije, vendar funkcija ne sme klicati samo sebe.

Funkcija mora biti definirana predno jo kličemo, zato navedemo vse DEF FN stavke na začetku programa.

Z naslednjimi funkcijami definiramo nekatere matematične funkcije, ki jih sicer nimamo :

10 DEF FNS(X)=ATN(X/SQR(-X*X+1)):REM ASIN V RADIANH

20 DEF FNC(X)=ATN(X/SQR(-X*X+1))+1.5708:REM ACOS V RADIANH

LINE INPUT

Ukaz LINE INPUT spejme kot podatek celo vrstico, skupaj z vejicami, drugimi ločili in presledki.

LINE INPUT "navodilo";ime stringa

Navodilo je lahko kakršen koli tekst v narekovajih. Običajno z njim povemo operaterju, kakšen podatek mora vnesti. Z enim LINE INPUT lahko vnesemo vsebino ene same spremenljivke. Največja dolžina vrstice, ki jo z LINE INPUT vnesemo v pomnilnik je 255 znakov.

```
10 CLEAR 500:CLS
```

```
20 LINE INPUT "IME IN PRIIMEK";NS
```

```
30 LINE INPUT "NASLOV          ";AS
```

CLEAR

Ukaz CLEAR briše vse spremenljivke in rezervira spominsko območje za stringe.

```
CLEAR 500
```

rezervira 500 Bytov spo mina

Ukaz CLEAR lahko uporabimo tudi za definiranje najvišjega naslova v spominu, ki ga lahko uporabi program v BASIC-u. Na ta način lahko rezerviramo prostor za programe v strojnem jeziku.

```
CLEAR 200,14000
```

rezervira 200 Bytov za stringe in definira 14000 kot najvišji naslov spomske celice. Programi v strojnem jeziku se lahko shranjujejo od 14001 dalje.

Brez Ukaza CLEAR se avtomatsko rezervira 200 Bytov za stringe.

READ
DATA
RESTORE

Ukaz READ prečita naslednje vrednosti iz DATA stavka in jih priredi spremenljivkam iz seznama.

READ seznam spremenljivk

seznam spremenljivk vsebuje imena spremenljivk, ki jim z READ priredimo vrednosti. DATA stavek vsebuje seznam števil, ki se kot podatki uporabijo za definiranje vrednosti spremenljivk.

DATA seznam števil ali stringov

V READ in DATA lahko uporabimo numerične in znakovne spremenljivke. Paziti moramo, da prirejamo podatke istorodnim spremenljivkam.

DATA stavek lahko stoji v katerikoli vrstici programa.

Ukaz RESTORE premakne kazalec, s katerim čitamo DATA stavke na začetek prvega DATA stavka. Na ta način lahko podatke v DATA stavkih večkrat uporabimo. RESTORE nima argumenta ali seznama.

RESTORE

SET

Ukaz SET uporabimo za postavitev točke na ekranu pri majhni ločljivosti.

SET (x,y,f)

x in y sta koordinati na ekranu, x mora biti med 0 in 63, y pa med 0 in 31.

f je kodno število barve in mora biti med 0 in 8

10 CLS0:SET(5,27,8):SET(6,27,8)

20 FOR X=0 TO 6:FOR Y=28 TO 30

30 SET(X,Y,8):NEXT Y,X

40 FOR X=7 TO 63:FOR D=1 TO 200:NEXT D

50 FOR Y=27 TO 30:IF Y=27 THEN RESET(X-2,Y)

60 SET (X,Y,8):RESET(X-7,Y:NEXT Y,X

70 GOTO 70

RESET

Ukaz RESET uporabimo za brisanje točke na ekranu, če je bila postavljena s SET. Uporablja se le pri majhni ločljivosti.

RESET (x,y)

PCLEAR

Z ukazom PCLEAR rezerviramo strani spomina za grafiko v različnih načinih z visoko resolucijo.

PCLEAR n

n mora biti število med 1 in 8. Če ne uporabimo ukaza PCLEAR se rezervirajo 4 strani spomina.

Ker vsaka stran spomina pokrije 1536 Bytov je smiselno, da rezerviramo le toliko strani, kot jih dejansko potrebujemo

SCREEN

Ukaz SCREEN se uporablja za preklapljanje z teksta na grafiko in obratno

SCREEN način, barvna skupina

Način ima lahko vrednost 0 za ekranski tekst in grafiko nizke ločljivosti, ali 1 za grafiko visoke ločljivosti.

Barvna skupina je lahko 0 ali 1. Za ekranski tekst da barvna skupina 0 črne znake na zelenem ekranu, 1 pa črne znake na oranžnem ekranu. Pri grafiki je skupina barv odvisna od izbranega načina, ki ga določa PMODE

PMODE

SCREEN 1,0

SCREEN 1,1

0	črna, zelena	črna, bela
1	zelena, rumena modra, rdeča	bela, cian, lila oranžna
2	črna, zelena	črna, bela
3	zelena, rumena modra, rdeča	bela, cian, lila, oranžna
4	črna, zelena	črna, bela

PMODE

Ukaz PMODE ima obliko

PMODE a,b

a je število med 0 in 4 in definira grafični način.

b število, ki definira stran, ki jo želimo uporabiti.

Ekran je razdelejen na 256 x 192 točk, neodvisno od grafičnega načina. Razlika je samo v velikosti točke.

V vsakem grafičnem načinu imamo dve skupini barv, ki ju izbiramo z ukazom SCREEN.

PMODE	velikost ekrana	velikost točke	št.strani spomina	brvni skupini	
				SCREEN 1,0	SCREEN 1,1
0	128x96	8x8	1	0,1	0,5
1	128x96	8x8	2	1,2,3,4	5,6,7,8
2	192x128	8x8	2	0,1	0,5
3	192x128	8x8	4	1,2,3,4	5,6,7,8
4	256x196	•	4	0,1	0,5

COLOR

Ukaz COLOR menja osnovno barvo in barvo ozadja pri grafiki velike ločljivosti.

COLOR osnovna barva, barva ozadja

PCLS

Z ukazom PCLS brišemo vsebino ekrana in izberemo barvo ozadja pri grafiki visoke ločljivosti.

PCLS f

f je koda barve ozadja

PSET

Ukaz PSET ima v grafiki velike ločljivosti enak pomen in strukturo kot ukaz SET v grafiki nizke ločljivosti.

PSET (x,y,f)

PRESET

Ukaz PRESET ima v grafiki velike ločljivosti enak pomen in strukturo kot ukaz RESET v grafiki nizke ločljivosti

PRESET (x,y)

PCOPY

PCOPY je ukaz s katerim v grafiki visoke ločljivosti kopiramo vsebine strani spomina v drugo stran.

PCOPY original TO kopija

Original in kopija sta števili med 1 in 8 in se nanašajo na strani spomina, ki so predhodno rezervirane z PCLEAR.

PCOPY 3 TO 5

LINE

Ukaz LINE uporabljamo za risanje črt in pravokotnikov.

Ukaz ima obliko :

LINE (x1,y1)-(x2,y2),a,b

x1 in y1 sta koordinati začetne točke, x2 in y2 pa koordinati končne točke. a je lahko PSET, tedaj se črta izriše v osnovni barvi, ali PRESET, tedaj se črta izriše v barvi ozadja (brisanje !). b je razširitev ukaza, ki ni nujna in jo lahko izpustimo. b je lahko B (BOX) ali BF (BOX FILL). Z B dosežemo, da se izriše pravokotnik, ki ima diametralne vogale v točkah x1,y1 in x2,y2. Z BF dosežemo, da se izriše enak pravokotnik, le da je v seloti izpolnjen z poljem v osnovni barvi.

10 PMODE 4,1:SCREEN 1,1:PCLS 5:COLOR 0,5

20 FOR I=1 TO 1000

30 X=X+L*SIN(R):Y=Y+L*COS(R)

40 IF X<-128 OR X>128 THEN 90

50 IF Y<-96 OR Y>96 THEN 90

60 LINE -(X+128,Y+96),PSET

70 R1=R1+60:R=R1/57.29578:L=L+0.5

80 NEXT I

90 GOTO 90

DRAW

Z ukazom DRAW lahko narišemo na ekran eno ali več črt, ki so definirane v stringu. Ukaz ima obliko :

DRAW string

String je lahko znakovna konstanta ali spremenljivka.

Vsebuje lahko naslednje ukaze :

M x,y	move, pojdi na točko s koordinatama x in y
U n	pojdi n točk navzgor
D n	pojdi n točk navzdol
L n	pojdi n točk v levo
R n	pojdi n točk v desno
E n	pojdi n točk v desno in navzgor (45°)
F n	pojdi n točk v desno in navzdol (135°)
G n	pojdi n točk v levo in navzdol (225°)
H n	pojdi n točk v levo in navzgor (315°)
X	izvede se string in nadaljuje izvajanje v isti vrstici
C	color, podana je barva črte s svojo kodo
A k	zavrti naslenje črte za kot k. Kot je med 0 in 4 in sicer : pri $k=0$... 0° , pri $k=1$... 90° pri $k=2$... 180° , pri $k=3$... 270°
S k	nariše črto v merilu k. Merilo je med $1:4$ in $14.5:1$ in sicer za $k=1$ je merilo $1:4$ k=4 je merilo $1:1$ k=8 je merilo $2:1$
N	položaj risalnega peresa se ne spremeni
B	položaj risalnega peresa se spremeni, vendar se ne riše (dvignjeno pero)

Relativen premik lahko dosežemo z parametrom M v naslednji obliki :

M+x,+y

+x in +y so števila, s katerimi je podana velikost premika v x in y smeri. Predznaki so obvezni !

GET

Ukaz GET lahko uporabimo le v grafiki visoke ločljivosti. Z GET shranimo rizbo z definirane pravokotnika na ekranu v polje spremenljivke. Velikost polja mora biti predhodno dimenzionirana. Ukaz ima obliko :

GET(x1,y1)-(x2,y2),ime polja,G

x1,y1 in x2,y2 so koordinate točk, ki definirata gornji levi in spodnji desni vogal pravokotnika na ekranu.

ime polja je ime vnaprej definirane spremenljivke, dimenzija polja pa je enaka številu stolpcev in vrstic v pravokotniku na ekranu.

G je zahteva, da se slika shrani z vsemi grafičnimi detajli. Ta del ukaza lahko izpustimo.

Z ukazom PUT narišemo na ekran pravokotnik, katerega vsebino smo shranili z GET. Ukaz PUT moramo uporabiti v enakem grafičnem načinu, kot je tisti, s katerim smo definirali grafiko ob shranjevanju z GET. Ukaz ima obliko:

PUT(x1,y1)-(x2,y2),ime polja,ukaz

x1,y1 in x2,y2 so koordinate točk, ki definirajo gornji levi in spodnji desni vogal pravokotnika na ekranu.

ime polja je isto ime, pod katerim smo shranili risbo z ukazom GET in mora biti vnaprej dimenzionirano.

Ukaz moramo navesti, ako je bilo polje shranjeno z uporabo parametra G, sicer ne. Ukaz je lahko:

PSET ... nariše sliko, kot je bila shranjena z GET

PRESET... nariše negativ slike, kot je bila shranjena

AND... primerja sliko v spominu s sliko na ekranu, vse točke, ki so svetle v obeh slikah ostanejo svetle, ostale ugasnejo

OR ... primerja sliko enako kot AND, le da ostanejo svetle vse točke, ki so svetle vsaj v eni sliki

NOT ... ugasne vse svetle točke na ekranu v območju pravokotnika in prižge vse ugasnjene, neodvisno od vsebine shranjenega polja.

Izrisana slika mora ustrezati dimenziji polja spremenljivke, sicer se lahko na ekranu pojavijo poljubni znaki.

10 PCLEAR 4:PMODE:3,1:PCLS:SCREEN1,1:DIM W(30,30)

20 DRAW"BM10,12,S8;R1U3R1D2R2U2R1D3R1D2L1D2R1D1L2U3L4
D3L2U1R1U2L1U2R1U2BR1BD1D2R2U2NL2R2D2L2U2"

30 PAINT(11,13),6,5:GET(0,0)-(30,30),W,G

40 AS=INKEY\$:IF AS="" THEN 40

50 PCLS:FOR C=0 TO 100 STEP 20

60 FOR A=0 TO 200 STEP 20

70 PUT(A,C)-(30+A,30+C),W,PSET

80 PUT(A,30+C)-(30+A,60+C),W,PSET

90 PUT(A,C+60)-(30+A,90+C),W,PSET

100 PLAY"T255;ABFGBA::PCLS:NEXT A,C

PAINT

Ukaz PAINT nam omogoča, da narisani lik izpolnimo z izbrano barvo. Ukaz ima obliko :

PAINT (x,y),m,s

x,y sta koordinati točke, kjer naj se barvanje prične.

m je koda barve, s katero želimo lik izpolniti. Lahko je številka med 0 in 8 in biti mora v barvni skupini izbranega načina. Če m ne definiramo se vzame osnovna barva.

Barvanje prične pri x,y in se nadaljuje, dokler ne zadene na točko, ki imajo barvo kodne številke s. Če s ni definiran se privzame koda osnovne barve.

CIRCLE

Ukaz CIRCLE nam poenostavi risanje kroga, elipse in krožnega loka. Ukaz ima obliko :

CIRCLE (x,y),r,f,hb,začetek,konec

x je x-koordinata središča kroga (0 do 255)

y je y-koordinata središča Kroga (0 do 191)

r radij kroga v ekranskih enotah

f kodno število izbrane barve, če ni podan se privzame koda osnovne barve

hb je razmerje med višino in širino (ekscentričnost). Uporablja se za risanje elips. Če ni podan se privzame vrednost 1.

začetek je začetna točka krožnega loka (od 0 do 1). Risanje se prične na horizontalni osi (x) in poteka v smeri vrtenja urinih kazalcev.

konec je končna točka krožnega loka (od 0 do 1)

AUDIO

Ukaz AUDIO poveže tonski izhod kasetofona s tonskim vhodom televizorja. Povezavo vzpostavimo z ukazom AUDIO ON

In prekinemo z ukazom
AUDIO OFF

MOTOR

Z ukazom MOTOR vključujemo pogon kasetofona.

MOTOR ON vključi motor kasetofona,
MOTOR OFF pa ga izključi.

PLAY tipka kasetofona mora biti ves čas vključena.

PLAY

Z ukazom PLAY lahko zaigramo melodijo. Ukaz ima obliko :

PLAY glasba

glasba je sestavljena iz naslednjih elementov :

nota : od A ali število od 1 do 12

oktava : On, n je število med 1 in 5, osnovna nastavitev 02

dolžina note: Ln, n je med 1 in 255, osnovna nastavitev L4

tempo : Tn, n je med 1 in 255, osnovna nastavitev T2

glasnost : Vn, n je med 0 in 31, osnovna nastavitev V15

pauza : Pn, n je med 1 in 255

izvedba stringa :X ime znakovne spremenljivke;

z + ali # dosežemo povišanje za pol tona, z - znižanje

Dolžino note lahko modificiramo s tem, da za številko dodamo piko (L2.). Na-ta način proizvedemo punktirano noto.

Oktava, glasnost, tempo in dolžina note se lahko modificira

z naslednjimi znaki : + - < >

FUNKCIJE

Funkcija je za računalnik poseben podprogram, ki je shranjen skupaj z ostalimi ukazi programskega jezika BASIC.

Funkcija ima obliko :

IME FUNKCIJE (ARGUMENTI)

in se uporablja v formulah in stavkih enako kot matematični operatorji (+, -, *, ...)

1. skupina : matematične funkcije

ABS (X) absolutna vrednost

ATN (X) arkus tangens

COS (X) kosinus

EXP (X) e^x

FIX (X) izračuna celoštevilčno vrednost, vse številke za decimalno piko se brišejo

INT (X) enako kot FIX (X) za $X = 0$, za negativne x -e se izvede zaokroževanje navzdol $INT(12.01) = -13$!

JOYSTK (X) vrednost je trenulni horizontalni ali vertikalni položaj krmilnih palic

$X=0$ desna k.p. horizontalno

$X=1$ desna k.p. vertikalno

$X=2$ leva k.p. horizontalno

$X=3$ leva k.p. vertikalno

LOG (X) naravni logaritem, X mora biti večji od 0

PEEK (X) rezultat je vsebina spominske celice z naslovom X

POINT(X,Y)	testira, ali je točka v grafiki majhne ločljivosti osvetljena. X mora biti med 0 in 63, Y med 0 in 31. Vrednost funkcije je 0 če je točka temna in 1, če je točka na znaku ter 1 do 8, če je točka osvetljena. Vrednost ke koda barve.
POS(X)	da položaj pisalne glave na tiskalniku ali kurzorja na ekranu v horizontalni vrstici. X=0 ... kurzor X=-2 ... tiskalnik
PPOINT(X,Y)	ukaz ima enako funkcijo kot POINT, le da velja za grafiko velike ločljivosti. X mora biti med 0 in 255, Y med 0 in 191.
RND(X)	vrednost funkcije je naključno število med 1 in X
SGN(X)	vrednost funkcije je enaka predznaku argumenta. -1 za $X < 0$ 0 za $X = 0$ +1 za $X > 0$
SIN(X)	sinus (x), x je kot vadianih
SQR(X)	kvadratni koren argumenta. Če je argument negativen, se izračuna koren njegove absolutne vrednosti in mu da negativen predznak.
TAN(X)	tangens(X), X je kot vadianih

2. skupina : funkcije te skupine imajo numeričen argument, njihov rezultat pa je string

CHR\$(X)	izračuna znak, ki odgovarja znakovni kodi X. X mora biti med 0 in 255. Tabela kod znakov je v prilogi.
HEX\$(X)	izračuna heksadecimalno vrednost števila X
STR\$(X)	pretvori numerično vrednost v string

3. skupina : funkcije te skupine nam služijo za delo s stringi. Argument je največkrat string, lahko pa je tudi numeričen izraz. Rezultat je del stringa.

LEFT\$(X\$,N)	izračuna prvih N znakov stringa X\$
MID\$(X\$,M,N)	izračuna N znakov stringa X\$, začnši z M-tim znakom. M>0 !
RIGHT\$(X\$,N)	izračuna zadnjih N znakov stringa X\$
STRING\$(N,C)	izračuna string dolžine N. String sestavlja N znakov C. C je lahko kodno število v ASCII ali znak sam v narekovajih.

4. skupina : funkcije te skupine so podobne funkcijam 2. skupine. Njihov argument je string, izračunajo pa numerično vrednost.

ASC(X\$)	izračuna ASCII kodo prvega znaka v X\$-u
INSTR(P,S\$,Z\$)	išče string Z\$ v stringu S\$, pričnši od P-tega znaka. Če Z\$-a ne najde, dobi INSTR vrednost 0, če pa ga najde, dobi INSTR vrednost, ki odgovarja položaju prvega znaka Z\$ v S\$

LEN(X\$) izračuna število znakov v X\$. Preštejejo se vsi znaki, tudi prazna mesta med besedami. Prazen string da vrednost Ø.

VAL(X\$) pretvori vrednost v stringu zapisanega števila v njegovo numerično vrednost. še se X\$ prične z črko dobimo vrednost Ø.

5. skupina : sistemske funkcije brez argumentov

INKEY\$ preveri tastaturo in če je kaka tipka pritisnjena privzame znak s te tipke kot string.

MEM da število prostih spominskih mest

TIMER nam da vrednost računalnikove ure, ki je lahko med Ø in 65535. T ukazom
TIMER = Ø postavimo vrednost računalnikove ure na izhodišče.

PROGRAMI V STROJNEM JEZIKU

Prednost programov v strojnem jeziku je predvsem hitrejši potek, poleg tega pa potrebujejo manj spominskega prostora. Njihova slabost je predvsem v zahtevnosti njihovega sestavljanja, saj strojni jezik za razliko od BASIC-a ni orientiran k uporabniku, temveč k stroju.

Za programiranje v strojnem jeziku na računalniku GRAF 09 lahko uporabimo knjigo : Rodnay Zaks in William Labiak Programming the 6809, SYBEX Verlag GmbH, Heyerstrasse 22, 4000 Düsseldorf 12.

GRAF 09 ima že v osnovni verziji nekaj funkcij, ki nam omogočajo uporabo programov in programiranje v strojnem jeziku. to so :

USRn(argument)

n je številka od 0 do 9 in nam omogoča, da z ukazom USR kličemo enega od 10. podprogramov v strojnem jeziku.

argument je lahko string ali numerični izraz.

Ko računalnik v programu naleti na ukaz USRn, skoči na naslov, ki je definiran z ukazom DEF USRn

DEF USRn=naslov

naslov ima lahko vrednost med 0 in 65535 in mora označevati naslov spominske celice, kjer se prične program v strojnem jeziku.

CLEAR s,h

Z ukazom CLEAR rezerviramo prostor za programe v strojnem jeziku. s pove računalniku, koliko prostora je rezerviranega in h je najvišji naslov spominske celice, ki se lahko uporabi v BASIC-u. Od naslova h+1 dalje je pomnilnik rezerviran za programe v strojnem jeziku.

POKE naslov,vrednost

Ukaz POKE uporabimo, da bi v spominsko celico naslov vpisali vrednost. Vrednost je lahko med 0 in 255.

VARPTR (X)

Ukaz VARPTR nam da naslove spremenljivk v BASICU. Lhko jo uporabimo kot argument v USR funkciji. Na ta način lahko v strojnem programu spreminjamo vrednosti polja spremenljivke.

X je ime spremenljivke v BASIC-u.

Program v strojnem jeziku zapisujemo na kaseto z ukazom CSAVEM"ime",začetek, konec,začetni naslov

ime je ime programa

začetek je najnižji naslov spominske celice s programom

konec je najvišji naslov programske celice s programom

začetni naslov je naslov programske celice, kjer se program v strojnem jeziku prične izvajati.

Čitanje strojnega programa s kasete dosežemo z ukazom

CLOADM"ime", zamik,R

parameter zamik uporabimo, da se nalaganje prične od

začetni naslov + zamik dalje. Če ukaz zaključimo z R

se program takoj po končanem čitanju tudi izvede, ne da bi morali uporabiti ukaz EXEC.

EXEC naslov

je ukaz, s katerim startamo izvajanje strojnega programa.

Naslov označuje spominsko celico, kjer se program začne.

Če naslov izpustimo se uporabi podatek iz zadnjega CLOADM ukaza.

VRSTE ŠTEVIL

Vsa doslej navedena števila so destetiška števila (baza 10), GRAF 09 pa lahko uporablja tudi oktalna in heksadecimalna števila.

Pred oktalnimi števili mora stati znak &

49= & 61

Pred heksadecimalnimi pa &H

91= &H 5B

P R I L O G A A

ASCII kode

tipka	decimalna koda brez SHIFT s SHIFT		tipka	decimalna koda brez SHIFT s SHIFT	
(BREAK)	3	3	>	62	-
(CLEAR)	12	92	?	63	-
(ENTER)	13	13	@	64	19
(✓)	32	32	A	97	65
!	33	-	B	98	66
"	34	-	C	99	67
*	35	-	D	100	68
\$	36	-	E	101	69
%	37	-	F	102	70
&	38	-	G	103	71
,	39	-	H	104	72
(40	-	I	105	73
)	41	-	J	106	74
+	42	-	K	107	75
+	43	-	L	108	76
,	44	-	M	109	77
-	45	-	N	110	78
.	46	-	O	111	79
/	47	-	P	112	80
0	48	18	Q	113	81
1	49	-	R	114	82
2	50	-	S	115	83
3	51	-	T	116	84
4	52	-	U	117	85
5	53	-	V	118	86
6	54	-	W	119	87
7	55	-	X	120	88
8	56	-	Y	121	89
9	57	-	Z	122	90
:	58	-	↑	94	95
;	59	-	↓	10	91
<	60	-	←	8	21
=	61	-	→	9	93

[... CHR\$(123),]...CHR\$(125), znaki od 128 do 255 so grafični znaki.

P R O G A M
O Z N A K E N A P A K

oznaka	razlaga
/Ø	deljenje z Ø, nemogoča operacija
AO	(Attempt to open) poizkus odpiranja datoteke, ki je že odprta. Do nje pride najpogosteje, če uporabimo RESET, da bi preprečili dostop do datoteke. Izključi in ponovno vključi računalnik.
BS	(Bad Subskript) največkrat je indeks spremenljivke večji od ustreznega DIMENSION
CN	(Can't Continue) CONT smo uporabili, čeprav je program že dosegel END
DD	(Dimension Dimension) zahteva za spremembo velikosti polja ni sprejemljiva. Vsaka spremenljivka je v programu lahko le enkrat dimenzionirana.
DS	(Direct Statement) največkrat, če želimo čitati podatke z CLOAD.
FC	(Function Call) napačen klic funkcije. Največkrat ker je parameter izven dovoljenega obsega ali je napačnega tipa.
FD	(File Data) čitate numerične podatke kot string ali obratno
FM	(File Mode) čitaš z INPUT iz datoteke, ki je odprta z OPEN"O" ali pišeš z PRINT v datoteko, ki je odprta z OPEN"I".
ID	(Illegal Direct) ukaz se lahko uporablja le v programu
IE	(Input Error) čitanje preko konca datoteke. Uporabimo lahko IF EOF(-1).
IO	napaka na kaseti ali napačna nastavitev rekorderja
LS	(Long String) string ne sme imeti več kot 255 znakov
NF	(Next/For) v programu je ukaz NEXT, ne da bi se vezal na ustrezen ukaz FOR.
NO	(No open) čitanje datoteke ali pisanje v datoteko, ki ni odprta z OPEN.
OD	(Out of Data) READ brez ali z prekratkom DATA stavkom
OM	(Out of Memory) vse razpoložljive spominske celice so zasedene ali reserovane

- OD (Out of DATA) READ brez ali s prekratkim DATA stavkom se nahaja v programu.
- OM (Out of Memory) vse spominske celice so zasedene ali rezervirane.
- OS (Out of String Space) rezervirano območje za stringe (CLEAR) je zasedeno.
- OV (Overflow) vrednost leži izven območja (ABS(X) 1E38)
- RG v programu je RETURN, brez ustreznega GOSUB.
- SN (Syntax Error) v programu je pravopisna napaka
- ST (String Error) ukaz za delo s stringi je prekompliriran in se mora razgraditi v več ukazov.
- TM (Type Mismatch) numerični spremenljivki se prirejuje string ali obratno.
- UL (Undefined Line) predviden je skok v vrstico, ki je ni.