

redakční kolektiv

UŽIVATELSKÝ POPIS

počítačů

ATARI[®]

600 XL / 800 XL

knihovnicka klubu mikroelektroniky ssm

UŽIVATELSKÝ POPIS
počítačů

A T A R I
600XL/800XL

překlad

Ing. Věroslav SZPUK

sestavil

Oldřich Burser

a kolektiv

Úvod

Doplněné vydání překladu Uživatelského popisu počítačů ATARI 600XL/800XL vychází po úplném rozebrání prvního vydání omezeného nakladu obsahově shodného titulu, "Uživatelská příručka HC ATARI 600XL/800XL", která vyšla koncem roku 1985. V té době bylo jen málo těch, kteří by aktivitu autorů brožury považovali za racionální využití tvůrčí energie, protože v celé ČSSR byly ve spojení pouhé dvě desítky uživatelů počítačů této značky.

Publikování dalších monotematicky orientovaných titulů připravených již dříve pro knižní vydání bylo podmíněno jednak nezbytným nárůstem základny uživatelů HC ATARI, ale i organizačním začleněním klubu Ataristů do některé složky NF. Stoupající počet majitelů počítačů ATARI, spíše však podepsání smlouvy obchodní organizace TUZEX s rakouskou společností WIENA KOMERZ (o dovozu počítačů ATARI a téměř celé existující škály periferních zařízení do ČSSR), se staly rozhodujícími momenty v úvahách o uspořádání odborného semináře pro nové majitele HC ATARI ještě v prvním pololetí roku 1986. Jeho zorganizování, redice Uživatelské příručky, jakož i vydání dalších titulů zaměřených na tento systém, to je výsledek společné práce svazků a Svazarmovců severomoravského a západoslovenského kraje. Předložená publikace Uživatelského popisu počítačů ATARI 600XL,800XL vychází v účelové edici klubu mikroelektroniky SSM, který byl ustaven při ZO SSM SOUŽ Jimramov. Tento svazácký klub se stává odrazovým můstkem pro vznik organizované základny uživatelů počítačů ATARI, pro jejichž vřelou potřebu je také tato brožura znovu vydávána. Překlad cizojazyčného manuálu není ilustrován všemi v původním titulu použitými obrázky a důsledně nedodrжуje ani skladbu této publikace. V případě nesrovnalostí lze proto doporučit porovnání textu s originálem.

Kromě aktuálních informací o komerčně dostupném programovém vybavení počítačů ATARI v USA, viz str.57 až 62, uvádíme na str.53 až 55 neúplný přehled evidovaných programů dostupných v ČSSR. Případným návštěvníkům Jugoslávie je možno doporučit cenově velmi výhodnou nabídku programů Jugoslávského Atari klubu, jehož rozsahý seznam je uveden na str.63 až 67. Programy uvedené v tuzemském seznamu byly sesbírány, eventuelně vytvořeny, společným úsilím většiny stávajících uživatelů počítačů ATARI z celé republiky. S ohledem na rychle rostoucí počet majitelů HC ATARI, je a bude naším cílem urychleně vytvořit organizační zázemí pro nekomerční rozšiřování programů.

V zájmu nových majitelů počítačů, k nimž se pravděpodobně dostane tato brožura i nepřímou cestou (tak řečeno "z druhé ruky"), uvádíme na tomto nezvyklém místě adresu Atari klubu, kde je možno přímo nebo prostřednictvím klubového Zpravodaje načerpat první informace do trnitých začátků...

redaktor AZ

A T A R I klub
pošt.schr. 137

O L O M O U C 2
772 11

Počítačový systém ATARI XL

Počítačový systém ATARI XL vznikl dalším vývojem v celém světě úspěšné typové řady ATARI 400/800 se zvláštním zřetelem na důslednou kompatibilitu systému; to má pro uživatele tu výhodu, že i v novém počítači ATARI XL mohou být využity periferní zařízení ATARI a výpočetní programy, které jsou z dřívějších k dispozici.

Možnosti nasazení počítače ATARI jsou různorodé a jsou vlastně omezeny jen vlastní fantazií a danými finančními možnostmi. Vedle programů her pro zábavu nabízí se celá řada dalších programů z oboru výuky, dalšího vzdělávání, pro domácnost a povolání jakož i systémových programů pro programovou podporu. Navíc jsou k dispozici kromě programů, které jsou již v počítači pevně zabudovány, a to v programovacím jazyku ATARI BASIC, další jazyky s různými nositeli dat. Nabídka programů ATARI je průběžně rozšiřována a aktualizována.

Počítač ATARI XL pomáhá např. mládeži při výuce matematiky, němčiny a cizích jazyků, ulehčuje práci při zpracování dat a ulehčuje práci při výpočtech a plánování dle tabulek, které je jinak pracné. V neposlední řadě pomáhá při výuce informatiky, zavedené na mnohých školách. Technologický vývoj posledních let umožnil výrobu stejně výkonných počítačů ve velikosti psacího stroje se stejně jednoduchou obsluhou. Programy mohou být zadávány prostřednictvím klávesnice nebo mohou být používány hotové programy ve formě zástrčných modulů nebo jiných nosičů dat jako kazety nebo diskety.

Přehled systému

Základem systému je počítač ATARI XL, a to buď model 600 XL nebo 800 XL. Vedle mnoha shodných vlastností podstatným rozdílem je kapacita pracovní paměti (RAM). Je měřítkem pro najednou zpracovatelný rozsah dat a měrnou jednotkou je Kilobyte - zkráceně K. Pro uložení do paměti jednoho znaku je zapotřebí jednoho Bytu. Model ATARI 800 XL má k dispozici 64K a model ATARI 600 XL 16K, tyto však mohou být pomocí modulu pro rozšíření paměti ATARI zvýšeny na 64K. To odpovídá asi 30 plně popsaným stránkám DIN A4.

Výkon systému XL může být však plně využit teprve pomocí dalších periferních zařízení jako zapisovač programů nebo stanice disket. Slouží trvalému ukládání programů, které by jinak byly po vypnutí počítače smazány a jsou také předpokladem pro využití rozsáhlé nabídky hotových výpočetních programů s kazetami a disketami jako nosiči dat/programů.

Kromě pracovní paměti pro zpracování dat a programů je v počítači ještě další paměťové pole, a to provozní systém (ROM). Ten řídí, počítá a porovnává. Abychom měli do obou oblastí přístup a mohli tak s počítačem "mluvit", potřebujeme programovací jazyk.

Jednoduché a snadno naučitelné programovací jazyky, jako je např. v počítači ATARI zabudovaný BASIC, mají ten nedostatek, že počítač nemůže zpracovávat data jak možno nejrychleji, protože je musí ještě překládat do svého vlastního (strojního) jazyka a k tomu potřebuje poněkud více času.

Rychlé programovací jazyky, jako je např. ATARI Assembler vyžadují poněkud vyšší nároky na učení, umožňují však rychlý průběh programu zobrazením, odpovídajícím struktuře strojního vybavení počítače.

Klávesnice

Klávesnice odpovídá klávesnici psacího stroje s malými a velkými písmeny a dále s klávesami pro grafiku. Přesněji vzato má klávesnice tři sady kláves: normální klávesy, klávesy pro grafiku a pro přehlásky.

Normální klávesnice

Klávesou CAPS lze přepínat mezi velkými a malými písmeny.

Klávesnice pro grafiku

Stlačením klávesy CONTROL současně s další klávesou zobrazí se příslušný grafický znak.

Klávesnice a přehláskami

Na tuto vnitřní sadu znaků lze přepnout zadáním POKE 756,204 (RETURN). Žádaný znak (přehláska) zobrazí se na stínítku dodatečným stlačením klávesy CONTROL. Vypsání POKE 756,224 vrátíme se zpět do normálního modu.

Funkční klávesy

Zvlášt uspořádané klávesy RESET, OPTION, START, SELECT a HELP mají zvláštní funkce:

- RESET nechá počítač právě prováděnou úlohu přerušit a vrátit se do původního stavu (výchozího). Této klávesy by se proto nemělo používat k zamýšlenému přerušení programu, jelikož v určitých případech dojde ke smazání celého obsahu paměti.
- HELP vyvolává v některých programech vysvětlivky a pokyny pro obsluhu. Následující tři klávesy pro zvláštní funkce jsou v zásadě rovnocenné a volně programovatelné, pro snadnější práci jsou většinou používány následovně:

OPTION volí proměnné uvnitř jednoho programu.

SELECT umožňuje volbu použití uvnitř jednoho programu.

START začíná hru nebo jeden díl programu.

BREAK slouží k přerušení uživatelem psaného programu.

ESC (ESCAPE) je ve funkci záviela na daném programu.

Pomocí CONTROL INSERT mohou být vsunuty prázdné znaky nebo volný mezíprostor.

DELETE BACK SPACE posunuje kurzor doleva a zhasí příslušný znak.

CONTROL a DELETE BACK SPACE zhasí znak pod kurzorem. Znaky vpravo se přisunou.

SHIFT vyvolává druhé osazení kláves jako např. velká písmena a zvláštní znaky.

CAPS slouží k přepínání mezi velkým a malým písmem. Uvolňuje modus CONTROL LOCK.

CONTROL a LOCK ovlivňuje modus CONTROL LOCK tak, že CONTROL je "zablokovan".

SHIFT a CAPS blokuje počítač v modu velkých písmen. Pro zvláštní znaky číselných kláves musí být navíc stlačeno SHIFT.

CONTROL s I přerušuje zobrazení na obrazovce. Opětovné stlačení obnovuje zobrazení.

CONTROL s 2 nechá zaznít bzučák.

CONTROL s 3 označí po zadání konec dat (EOF).

SHIFT s INSERT přidá nad kurzorem prázdný řádek.

SHIFT s DELETE BACK SPACE zhasne řádek ve kterém se nachází kurzor.

CONTROL s klávesnicí se šipkou pohybuje kurzorem, aniž by měnil program nebo zobrazení.

RETURN přesune kurzor na levý okraj. Informuje počítač o konci psaní nebo edice jedné programové řádky.

INVERSE VIDEO přepíná na negativní zobrazení a naopak.

V některých programech je tato klávesa nazvána ATARI-klávesa .

AUTO REPEAT způsobí, že delším přidržením jedné klávesy je příslušný znak opakován.

Periferní zařízení

Paměť dat

Pro ukládání dat do paměti a pro zavádění uživatelem psaných nebo předem zhotovených softwarových programů se používá dvou rozdílných systémů:

Zapisovač magnetofonový jako např. model ATARI 1010 poskytuje při mírných pořizovacích nákladech rozsáhlé možnosti pro ukládání do paměti a zavádění programů a dat. Jako nosič dat doporučuje se magnetofonová kazeta (Kompakt) ve Ferro-kvalitě. Chromdioxidové kazety jsou z technických důvodů méně vhodné. Paměťová kapacita obnáší dle délky pásku až do 100 KBytů, což odpovídá přibližně 60 na stroji psaným stránkám. Zapisovač je méně vhodný, jedná-li se o rychlé, volitelné vyhledávání údajů.

Disketová stanice jako např. model ATARI 1050 poskytuje při rozsahu paměti 127 KBytů rychlý přístup k údajům. Při sdruženém propojení více stanic lze celkový uzavřeně zvládnutelný rozsah dat mnohonásobně zvýšit. Všechny náročné programy pro zpracování dat předpokládají použití disketových stanic.

Pro výkonnost disketové stanice je směrodatný disketový provozní systém (DOS). Náleží do rozsahu dodávky každé disketové stanice ATARI a je přirozeně v německé řeči (návod je přeložen i do češtiny a slovenštiny).

DOS je samostatný program, který umožňuje výměnu programů a dat mezi počítačem a stanicí včetně řízení a dozoru. Bez DOS nemůže být program ani zaváděn ani ukládán do paměti. Verze DOS 3, která je nyní k dispozici, jsou obsaženy ještě další zlepšení. Zvýšená kapacita paměti, vysvětlivky k programu vyvolatelné klávesou HELP, zabudování uživatelem vypsanych zvláštních funkcí je rovněž možné.

Tiskárny

Barevným tiskárnám dává se přednost pro tisk vícebarevných grafů a obrázků. Model ATARI 1020 např. pracuje se čtyřmi různými barvami na 115 mm širokém rolovacím papíru.

Matricová tiskárna kreslí obrázky pomocí bodové matrice a používá různých šířek písma. Proto se s oblibou používá k tisku tabulek a výsledků výpočtů. ATARI nabízí v typech 1025 a 1029 dva zvláště výkonné přístroje.

Pro soukromou korespondenci se hodí ideálně dopisní tiskárna ATARI 1027. Poskytuje písmo jako psací stroj a píše vpřed/vzad.

Příslušenství

Ovládače slouží k přidavnému individuálnímu vkládání dat. Používá se jich především pro rychlé hry s otočnými regulátory a ovládacími páčkami. Dalším vývojovým stupněm těchto regulátorů je Track-Ball. Má místo řídicí páky ve všech směrech pohyblivou kouli a umožňuje zvláště rychlou reakci.

Zvláštní číselné klávesnice

Zvláštní číselné klávesnice umožňují rychlé zadávání čísel. Podstatná a účelná pomoc pro mnohé programy (tabulkové výpočty). Vysoce zajímavá je možnost bezpracných uživatelem psaných programů kreslit obrázky přímo na obrazovku. ATARI kreslící tabule a ATARI světelné pero (Lightpen) využívá úplnou barevnou paletu počítače ATARI. Obrázky je možno ukládat na kazetu nebo disketu a lze je zabudovat do uživatelem psaných programů v BASIC.

Uvedení do provozu

1. Zasuňte kulatou zástrčku síťového zdroje do zásuvky označené POWER na zadní stěně počítače. Napojte pak síťový zdroj na domácí síť.
2. Anténní kabel propojí se s TV zdířkou počítače a anténním vstupem TV přijímače. Kdyby váš televizor neměl normalizovaný vstup opatřte si u odborného obchodníka vhodný adaptor. Při použití monitoru použijte monitorovou zdířku počítače a vhodného spojovacího kabelu.
3. Televizor nalaďte na kanál 4 (VHF) nebo 36 (UHF)*. Zapněte počítač. Bzučivým tónem objeví se na obrazovce hlášení READY. Dle potřeby doladí se na přístroji ostrost. Když zůstane údaj delší čas bez změny, mění se jeho barva v pravidelných intervalech. To je normální a slouží to k ochraně obrazovky. Po vypnutí by se mělo počkat před novým zapnutím 3-4 vteřiny, aby elektronika nebyla přetížena. (*Zelaštké při zavádění strojových komerčních programů*).

Zkouška systému

Po každém zapnutí zkontroluje počítač sám sebe a zobrazí za krátký okamžik na obrazovce startovací obrazec. Pokud v počítači něco nefunguje bez závad, objeví se na obrazovce hlášení MEMORY TEST.

Tento test můžete vyvolat také sami,
zapnutím počítače při stlačené klávese OPTION

nebo

po hlášení READY vypsáním slova BYE a stlačením klávesy RETURN. Objeví se SELF-TEST menu. Jako menu je označován předem daný program nebo volba funkce.

Zvolte klávesou SELECT žádanou zkoušku a stlačte tlačítko START. Stlačením tlačítka HELP vrátí se SELF-TEST menu. Tlačítkem RESET může být zkouška přerušena a umožněn návrat k ATARI BASIC.

Zkouška paměti

Počítač má dva typy paměti: pevnou paměť (ROM) a volně přístupnou pracovní paměť (RAM), do které mohou být ukládány programy, data a zpracovávány.

Zkouška ROM: objeví se dva pruhy, které se musí měnit od bílé do světle zelené barvy (příp. modré). Při zbarvení do červené nebo růžové je přítomna závada.

Zkouška RAM: pro každý 1 KByte objeví se jeden čtverec. U 600XL tedy 16, u 800XL až 64 čtverců, které se barevně mění při bezvadné funkci tak, jako výše uvedené.

Důležité:

800 XL znázorní při zkoušce pomocí tlačítka OPTION 48 čtverců, jelikož zbývající rozsah do 64 K je zabrán provozním systémem a zabudovaným programovacím jazykem ATARI BASIC.

Celkový rozsah 64 K může však být také znázorněn, když při zapnutém počítači po hlášení READY vložíme tento příkaz:

POKE 740,0 : BYE (RETURN)

Zbývajících 24 čtverců se pak musí objevit červeně. Kdyby se zkouška po hlášení READY odstartovala vložením BYE, objeví se s použitím BASIC-u jen 40 čtverců. Jak je možno využít celého rozsahu 64K RAM vyřazením provozního systému, nebude v tomto návodu popsáno. Je užitečné jen pro zkušené programátory.

Zkouška tón/grafika

Počítač ATARI XL má 4 na sobě nezávislé tónové kanály a grafiku s vysokým rozlišením. Nad číslem právě zkoušeného kanálu objeví se notové linky s houslovým klíčem. Při každém ze 4 kanálů zobrazí se a zahráje 6 tónů. Chybní-li tón, přezkoušejte regulátor hlasitosti přijímače. Během průběhu zkoušky musí barvy zůstat nezměněny.

Zkouška klávesnice

Na obrazovce se objeví klávesnice. Stlačené klávesy blikají. U SHIFT a CONTROL je třeba současně stlačit i jinou klávesu. BREAK je bez funkce, HELP nebo RESET ukončují zkoušku.

Poznámka: Z technických důvodů neodpovídá horní řada uspořádání klávesnice vašeho počítače. Zobrazená čísla 1 - 4 jsou rezervována pro pozdější rozšíření funkcí.

BASIC

Programovací jazyk BASIC byl vypracován již v 60 letech v USA a je dnes nejrozšířenější programovací jazykem světa. Stálý další vývoj pro další možnosti použití vede k dialektům specifickým pro dané výrobky. Žde uvedený ATARI BASIC* je jazyk, přizpůsobený výkonosti vašeho ATARI počítače. Jeho struktura, snadná k naučení, ulehčuje pochopení dalších vyšších programovacích jazyků. Následující kurs ATARI BASIC pojednává o nejdůležitějších pojmech a zvláště se zabývá fascinujícími barevnými, grafickými a tónovými možnostmi. Je zaměřen především pro začátečníky. Pro další informace odkazuje na přehled literatury v dodatku. Cvičné programy, následující po tomto kursu, mají upevnit nabyté vědomosti a dočkat odvahy samostatnému sestavování jednodušších programů.

ATARI BASIC

Propojili jste dle pokynů počítačový systém ATARI s televizorem/monitorem a uvedli do provozu zabudovaný systém ATARI BASIC je aktivován a hlásí se:

READY

Malý bílý čtverec po READY je kurzor. Udává na stínítku polohu, kde se bude nalézat příští znak.

Stlačíte-li při zapnutí počítače také tlačítko OPTION, objeví se SELF-TEST menu. Pokud je zapnuta také stanice disket, může se při použití špatné diskety anebo nesprávné obsluhy objevit také BOOT-ERROR.

Ještě několik důležitých pokynů:

- Všechny zápisy musí být prováděny v normálním modu, t.j. výroky a návěští velkými písmeny. Malá písmena budou akceptována jen uvnitř uvozovek.
- Jeden řádek může být dlouhý do 114 znaků. Jelikož obrazovka zobrazuje jen 38 znaků na řádek, přeskočí počítač automaticky o řádek níže. Konec této "logické řádky" o 118 znacích oznámí počítač akustickým signálem.

* Vyšel již český překlad




- Všechny zápisy je třeba uzavřít na konci řádky tlačítkem RETURN.
- Dbejte na rozdíl mezi "0" (nula) a "O". Nula je umístěna na horní řadě klávesnice. Pro lepší rozlišení je při vypisování přes tiskárnu šikmo příčně přeškrknuta. Písmeno O je o řádek níže.
- Po každém řádku přezkouší počítač automaticky, je-li zápis logicky správný a při chybě hlásí ERROR
- Opakovaně zprostředkované informace jako stlačení klávesy RETURN na konci řádku nebudou již pro lepší srozumitelnost v průběhu tohoto kursu BASIC u různých příkladů programů opakovány.

Výrok/Pokyn	Str.:
PRINT	7
Přímý režim a programovací režim	7
LIST	7
NEW	8
END	8
Hlášení chyby	8
Řazení pokynů za sebou	8
Rozšíření pokynu PRINT	9
Zacházení s čísly	9
Proměnné	10
INPUT	11
REM	11
FOR/NEXT	11
GOTO	12
IF/THEN	13
READ/DATA/RESTORE	13
Indexované proměnné	
GOSUB/RETURN	16
ON/GOTO	16
ON/GOSUB	17
GRAPHICS 0-2	18-19
GRAPHICS 3-8	19
PEEK/POKE	23
GRAPHICS 9-11	24
Tabulka grafických stupňů	25
SOUND	
Volání řízení	

PRINT

Pokynem PRINT s libovolným doplněním zobrazí počítač něco na obrazovce. Výrokem PRINT dají se m.j. provádět číselné operace. Místo výroku PRINT může být vložen i ? otazník. U otazníku ? jedná se o zkrácení výroku PRINT.

Základní druhy výpočtů:

Sečítání	:	+	tlačítkem	
Odečítání	:	-	"-"	+ 
Násobení	:	*	"*"	* 
Dělení	:	/	"-"	?/

Abychom dostali výsledek není zapotřebí znaku "=", Po vložení výroku PRINT stačí stlačení klávesy RETURN.

Při umocňování stlačí se klávesy SHIFT a *^{⊞^}
Pokyn PRINT hodí se však také k tomu, aby na obrazovce bylo vytištěno znaky nebo texty. Musíte jen text který má počítač vypsát dát do uvozovacích znamének.

Při tom dbejte na to, aby počítač nezobrazil na obrazovce také uvozovací znaménka.

LPRINT

BASIC výrok LPRINT (zkratka pro Line print - tisk do vedení) má stejnou funkci jako pokyn PRINT, počítač však vydává výpočty a texty nikoliv na obrazovku nýbrž do připojené tiskárny.
Poznámka: Kdyby počítač hlásil po použití LPRINT ERROR 138, znamená to většinou, že tiskárna není připojena nebo je třeba přezkoušet vedení.

Přímý modus a programovací modus

Přímý modus. Dosud prováděl počítač výroky ihned po stlačení tlačítka RETURN. Vkládání v přímém modu je užitečné tehdy, mají-li se výpočetní operace provést ihned a obdržet tak výsledky co možná nejrychleji.

Programovací modus. Pro řešení komplexních úkolů je zapotřebí předat počítači řadu pokynů, které provede až po vložení určitého výroku. Řada pokynů, které počítač ihned neprovádí, se nazývá programem. Program se startuje BASIC výrokem RUN. Po každém vložení RUN proběhne program znovu. Program se liší od výroků v přímém modu tím, že před každým výrokem je uvedeno ještě tzv. číslo řádku. Jako číslo řádku se hodí u počítače ATARI kterékoliv číslo od 0 do 32 767.

Odstartuje-li se program pomocí RUN, provede počítač programové řádky ve vzestupném pořadí.

Při programování používají se často v 10-ových odstupech, aby zbylo dostatek místa pro dodatečné vsunutí doplňkových řádků.

Choeme-li řádek změnit, je nejjednodušší vypsát nový řádek se stejným číslem řádku a pak stlačit RETURN. Starý řádek se nahradí novým i když je kratší. Řádek se dá úplně vymazat, vloží-li se číslo řádku, který se má smazat, a stlačí RETURN.

LIST

Výrok LIST slouží k tomu, aby byly na obrazovce zobrazeny řádky, které již byly vypsány. Po provedení výše uvedených příkladů vypadá obrazovka po vložení LIST takto:

Při delších programech je pro lepší přehled provést soupis jen jedné části programu. Je-li např. zapotřebí provést výpis jen jedné řádky,

postačí vložit LIST s připojeným číslem řádky. Vložení LIST 20 objeví se na obrazovce řádka 20 PRINT "WALTER". Je-li třeba provést výpis určité části programu, musí se k LIST doplnit ještě číslo první a poslední řádky segmentu programu, který má být vypsan. Vložení LIST 10,20 vypíše všechny řádky od 10 do 20 i ty, které se nacházejí mezi těmito dvěma. Při tom se nesmí zapomenout na čárky mezi oběma čísly. Má-li se vydat výpis programu do připojené tiskárny, použije se výroku LIST "P:" Stanovená část programu vydá se do tiskárny pokynem: LIST "P:",10,20 (RETURN)
Důležité jsou při tom čárky mezi pokynem LIST "P:" a prvním číslem. Druhá čárka s druhým číslem může být vypuštěna, jedná-li se o vytištění jen řádku 10.

NEW

Zadáním NEW smaže se celý program uložený v počítači. Po vložení LIST po výroku NEW způsobí, že počítač vypíše jen READY na obrazovku, jelikož v paměti nemá k vypsaní k dispozici žádný program.

END

Často je zapotřebí výrok, který program na určitém místě zastaví. Potřebný výrok BASIC k tomu účelu je END. Řádek 100 END způsobí, že počítač ukončí program v řádce 100.

Hlášení chyb

U počítače ATARI jsou dva různé druhy hlášení chyb. První druh se vyskytuje tehdy, dají-li se počítači příkazy, které nemůže provést, jelikož to nejsou výroky BASIC. Může se při tom jednat o špatně napsaný výrok BASIC, nebo se napíše středník místo čárky apod. Počítač odpovídá na tyto chyby takto:

```
10 PRINT "AHOJ" (RETURN)
10 ERROR-PRINT "AHOJ"
LIST 20;30 (RETURN)
ERROR-LIST 20;30
```

Při tom počítač označí místo, na kterém došlo k omylu tak, že příslušný znak vypíše inverzním písmem. Místo, které označí počítač jako chybu, nemusí vždy souhlasit se skutečnou pozicí chyby.

Druhý druh chyb se vyskytuje většinou při provádění programu. Počítač pak přeruší program a vypíše slovo ERROR s následným číslem, charakterizujícím druh chyby a číslem řádku, ve kterém došlo k chybě. V následujícím příkladu programu hlásí počítač při nepřipojení tiskárně chybu:

```
10 LPRINT "TEST TISKARNY"
RUN
ERROR 138 LINE 10
```

Z odstavce LPRINT víme, že se jedná o výrok, který vyžaduje připojení tiskárny. Význam dalších čísel chyb je uveden v dodatku ke cvičným příkladům.

Řazení pokynů za sebou

V BASIC je možné psát více výroků místo do různých řádků do jednoho, počítač je pak provede jeden po druhém. Jednotlivé výroky musí pak být odděleny dvojtečkou. Dva výroky PRINT mohou být sdruženy následovně: 10 PRINT "AHOJ";PRINT "PEPO"
Počítač provede oba výroky PRINT tak, jakoby stály ve dvou řádcích.

```
RUN
AHOJ
PEPO
READY
```

Tak se dají výroky, které spolu logicky souvisí, psát do jednoho řádku. Touto technikou se zvýší přehlednost programů. Za druhé se tím uspoří také místa v paměti, kterých může být při velmi dlouhých programech nedostatek.

Rozšíření pokynu PRINT

Možnosti použití výroku PRINT dají se jednoduše rozšířit pomocí dvou znaků a jednoho přídavného výroku BASIC.

Čárka. Pomocí čárky můžeme vytisknout pod sebou texty ve sloupcích širokých 10 znaků.

```
10 PRINT "JEDNA","DVA","TRI"  
20 PRINT "1","2","3"
```

RUN

```
JEDNA      DVA      TRI  
1          2          3
```

READY

Středník; Pomocí středníku za výrokem PRINT způsobíme, že text za dalším výrokem PRINT se vytiskne hned za jeho koncem.

Program

```
10 PRINT "AHOJ MILY";  
20 PRINT "PEPO"
```

dá na obrazovce následující výraz:

AHOJ MILY PEPO

POSITION

Pokyn POSITION slouží k tomu, aby byl text umístěn na určitém místě obrazovky. Výrok musí být při tom vložen před příslušným výrokem PRINT. Typický výrok POSITION vypadá např. takto:

```
POSITION 20,10
```

První číslo udává, o kolik prázdných znaků má být text vzdálen od levého okraje. Druhé číslo udává vzdálenost od horního okraje obrazovky. Obě čísla musí být oddělena čárkou. Obrazovka má u počítače ATARI vodorovně 40 znaků a svisle 24 znaků. Čísla platná pro výrok POSITION leží tedy v rozsahu 0 - 39 příp. 0 - 23.

Jelikož různé TV-přijímače posunují zcela nebo částečně levou stranu obrazu, zobrazuje se normálně jen 38 znaků. Zkušení programátoři jsou však schopni za pomoci pokynu POKE i 40 znaků na řádku obrazovky.

Zacházení s čísly

Tečka a čárka. Počítač nezná u decimálních čísel čárku. Číslo deset-čárka-tři se píše u počítače s tečkou.

Špatně: 10,3 Správně: 10.3

U počítače nesmí se také jednotlivé skupiny tisícovek oddělovat čárkou.

Špatně: 9,999,999 Správně: 9999999

Pořadí výpočetních operací. Počítač vypočítává matematický výraz od leva do prava. Při tom však platí následující pořadí matematických operací:

1. Napřed se provedou operace v závorkách.
2. Následuje umocňování. 2^3 např. je při tom rovno $2*2*2$
3. Násobení a dělení
4. Sečítání a odečítání

Příklady: $4*5+2=22$

$$4*(5+2)=28$$

$$2^3+1=9$$

$$2^(3+1)=16$$

Zobrazení velmi velkých nebo velmi malých čísel

Počítač používá zobrazení s pohyblivou čárkou, aby mohla být zobrazena velmi velká nebo malá čísla. Při výroku PRINT 500000000000 (pětset miliard) vytiskne počítač 5.0E+11
Znamená to 5 krát 10 na 11, nebo 5×10^{11}

Při tomto způsobu psaní nazývá se 5 mantissa a +11 exponent. Obě jsou odděleny písmenem E.

Při převodu z čísel s pohyblivou čárkou na normální způsob psaní je třeba rozlišovat dva případy.

• Kladný exponent

Mantissa se píše zvlášť. Pak se musí desetinná tečka posunout o tolik míst doprava, kolik udává exponent. Vzniklá prázdná místa se doplní nulami.

Příklad: 3.141592E+10 Mantissa: 3.141592

Posunutí desetinné tečky o 10 míst doprava 31415920000

• Záporné exponenty

Mantissa se píše rovněž zvlášť. Desetinná tečka musí být posunuta doleva o tolik míst, kolik udává exponent. Prázdná místa se také zde doplní nulami.

Příklad: 4.674E-8 Mantissa: 4.674

Posunutí desetinné tečky o 8 míst doleva: 0.00000004674

Proměnné

Proměnná je paměťové místo pro jedno číslo nebo sled znaků. Název proměnné sestává z jednoho nebo více znaků (při počítači ATARI až 128 znaků).

Při pojmenovávání proměnných je nutno dbát následujících bodů:

1. Uvnitř názvu proměnné jsou přípustná pouze písmena A - Z a číslice 0 - 9.

2. Uvnitř názvu proměnné nesmí se vyskytovat žádný prázdný znak.

3. Prvním znakem názvu proměnné musí být písmeno.

4. Jsou přípustná jen velká písmena.

* Basic výrokem LET můžeme proměnné přiřadit číslo nebo řadu znaků. Řádkem: LET CISLO=27 přiřadí se proměnné CISLO hodnota 27. Pomocí výroku PRINT může být hodnota proměnné vytisknuta. PRINT CISLO
27

Abychom obdrželi hodnotu proměnné CISLO, nesmí být slovo CISLO za výrokem PRINT psáno v uvozovkách. Proměnné, do kterých jsou ukládána čísla, se jmenují číselné proměnné. Kromě číselných proměnných jsou ještě řetězcové proměnné (stringy). Do těchto se ukládají sledy znaků. Řetězcové proměnné se vyznačují tím, že na konec názvu proměnné je připojen znak dolaru \$. Řetězcové proměnné TEXT\$ přijímají text STRINGVARIABLE:

```
LET TEXT$="STRINGVARIABLE"
```

Text, který přiřazujeme proměnné, musí být uveden v uvozovkách.

Obsah TEXT\$ můžeme zobrazit na obrazovce vložím:

```
PRINT TEXT$
```

```
STRINGVARIABLE
```

Dříve než můžeme v programu použít řetězcové proměnné, musíme počítači sdělit, kolik může být v tomto řetězci uloženo maximálně znaků. To se provede výrokem: DIM

Řádek 10, ve kterém je řetězcová proměnná dimenzována na 20 znaků, vyhlíží následovně: 10 DIM TEXT\$ (20)

Do řetězcové proměnné můžeme nyní vložit i méně než 20 znaků. Není však možné uložit více než 20 znaků, nadbytečné znaky budou ztraceny. Není možné řetězcové proměnné přidělit text, který nebyl předem

dimenzován. Při tomto pokusu ohlásí počítač chybu, jelikož mu je proměnná neznámá.

INPUT

Výrok INPUT umožňuje, aby byly v každém programu při jeho průběhu přiřazovány proměnnými jiné hodnoty. Při výroku INPUT čeká počítač na další údaj a pak přiřadí tento údaj proměnné. Následující program je jednoduchou ilustrací výroku INPUT.

```
10 DIM NAMEŠ (20)
20 PRINT "BITTE GEBEN SIE IHREN NAMEN EIN"
30 INPUT NAMEŠ
40 PRINT "SIE HEISSEN" , NAMEŠ
10 DIM JMENOŠ
20 PRINT "UDEJTE SVÉ JMENO PROSIM"
30 INPUT JMENOŠ
40 PRINT "JMENUJE SE" , JMENOŠ
```

Počítač čeká na údaj, což poznáme podle otazníku, který se objeví automaticky na obrazovce při každém pokynu INPUT, který počítač provádí. Program pokračuje teprve po tom, až je údaj uzavřen pomocí (RETURN).

REM

Při složitých programech je účelné doplňovat program komentářem. Komentáře ulehčují např. třetím osobám porozumět programu. V programu BASIC dají se komentáře vsunout pomocí pokynu REM (REMARK=poznámka). Vše, co následuje v jedné programové řádce za výrokem REM počítač při provádění programu ignoruje.

```
10 REM +++ PŘÍKLAD PROGRAMU ===
20 REM RADKY 10 A 20 BUDOU BEHEM PROVADENI PROGRAMU IGNOROVANY
30 PRINT "HALLO"
```

Po odstartování programu provede počítač pouze výrok PRINT v řádce 30:
RUN
HALLO
READY

Za výrokem REM nelze žádné další výroky provést, jelikož je počítač interpretuje jako komentář. Pro každý výrok REM by se měla tedy používat vždy vlastní řádka.

FOR/NEXT

Tak zvaná smyčka FOR/NEXT používá se v BASIC programu k tomu, aby byla určitá část programu opakována tolikrát, kolik se předem udá. Smyčka FOR/NEXT začíná výrokem FOR a končí výrokem NEXT. Mezi nimi leží část programu, která má být opakována. Přídavná informace za výrokem FOR udává počítači, kolikrát má smyčka proběhnout. Pro smyčku FOR/NEXT je nutná řídicí proměnná, která se při jednotlivých průbězích smyčky zvyšuje nebo snižuje. Následující příklad čítá proměnnou POCET od 1 do 4. Počítač vytiskne při každém průběhu smyčky skutečnou hodnotu proměnné POCET na obrazovku:

```
10 REM +++ SMYČKA FOR-NEXT +++
20 FOR POCET=1 TO 4
30 PRINT POCET
40 NEXT POCET
RUN
1
2
3
4
```

READY

Mezi počáteční a konečnou hodnotou řídicí proměnné musí být výrok TO. Výrok TO je součástí smyčky FOR/NEXT.

Pomocí výroku FOR/NEXT můžeme počítat nejen po jednotlivých krocích, výrokem STEP za pokynem FOR může být zadána určitá zvolená šířka kroku. Následující program vytiskne všech šest dělitelných čísel mezi 6 a 60:

```
10 FOR CISLO=6 TO 60 STEP 6
20 PRINT CISLO, "JE DELITELNE 6"
30 NEXT CISLO
```

Pomocí výroku STEP je možno čítat i zpět. Je nutno pouze zadat zápornou šířku kroku:

```
10 FOR CISLO=100 TO 1 STEP - 1
20 PRINT CISLO
30 NEXT CISLO
```

Uvnitř smyčky FOR/NEXT je možné použít jiné smyčky:

```
10 REM +++ ZASMYCKOVANE SMYCKY FOR/NEXT
20 FOR VNEJSI = 1 TO 4
30 PRINT "VNEJSI SMYCKA"
40 FOR VNITRNI = 1 TO 2
50 PRINT "VNITRNI SMYCKA"
60 NEXT VNITRNI
70 NEXT VNEJSI
```

Dle tohoto příkladu bude vnitřní smyčka provedena čtyřikrát. Počítač tedy vypíše čtyřikrát tento text:

```
VNEJSI SMYCKA
VNITRNI SMYCKA
VNITRNI SMYCKA
VNEJSI SMYCKA
atd.
```

GOTO

Až dosud prováděl počítač program tvrdohlavě od počátku do konce. Pro další průběh musel být program znovu odstartován pomocí RUN. Výrokem GOTO může být provádění programu na libovolném místě přerušeno a v jeho provádění pokračováno na jiném místě. Při výroku GOTO skočí počítač na číslo řádky uvedené za výrokem GOTO. Následující příklad programu umožňuje přepočítání kilometrů na míle. Pomocí výroku GOTO je možno provést neomezené množství výpočtů, aniž by musel být program znovu startován pomocí RUN.

```
10 PRINT "KILOMETRY"
20 INPUT KM
30 LET M=KM/1.609344
40 PRINT KM; "KILOMETRY ODPOVIDAJI",
50 PRINT M; "MILE"
60 GOTO GOTO 10
```

IF/THEN

Pomocí výroku IF/THEN (když-pak) může počítač provádět porovnávání. Další průběh programu závisí pak na výsledku porovnání. Výsledek může být "pravdivý" nebo "falešný". Bude-li výsledek "pravdivý", provede počítač výroky následující za výrokem THEN. Není-li podmínka splněna, přeskočí počítač pokyny za výrokem THEN a provádí program dle další řádky.

Existují tato logická porovnání:

IF A=B THEN	(Je-li A rovno B, pak ...)
IF A>B THEN	(Je-li A větší než B, pak ...)
IF A<B THEN	(Je-li A menší než B, pak ...)
IF A>=B THEN	(Je-li A větší nebo rovno B, pak ...)
IF A<=B THEN	(Je-li A menší nebo rovno B, pak ...)
IF A<>B THEN	(Když A není rovno B, pak ...)

V krátkém programu může počítač porovnávat dvě čísla:

```
10 PRINT "UDEJTE CISLA"
20 PRINT "A= ",:INPUT A
30 PRINT "B= ",:INPUT B
40 IF A=B THEN PRINT "A JE ROVNO B"
50 IF A>B THEN PRINT "A JE VETSI NEZ B"
60 IF A<B THEN PRINT "A JE MENSI NEZ B"
70 GOTO 10
```

Za výrokem THEN může být jakýkoliv jiný výrok BASIC. Pomocí GOTO může program pokračovat dle výsledku porovnání na jiném místě. Stejně jako na "normální" řádce může být za THEN více výroků, které pak musí být odděleny dvojtečkou.

```
40 IF A=B THEN PRINT "A JE ROVNO B" ; GOTO 10
```

Výrokem IF/THEN se dají přirozené porovnávat i řady znaků:

```
10 DIM C$(5)
20 PRINT "UDEJTE HOSLO"
30 INPUT C$
40 IF C$(">") "ATARI" THEN PRINT "HESLO NENI SPRAVNE": GOTO 20
50 PRINT "HESLO SEDI BEREME"
```

*ladat klavesnicí
ATARI (něco jiné)*

READ/DATA/RESTORE

Výrok READ je třetí možností, vedle LET a INPUT, jak můžeme proměnné přidělit určitou hodnotu. Data, která mají být proměnné přiděleny, musí být již v programu ve tvaru DATA-řádků k dispozici.

Řádek DATA vypadá následovně: 70 DATA 23,34,545,3,323,53.

Hodnoty za výrokem DATA musí být vždy odděleny čárkou. Před první a za poslední hodnotou DATA řádku nesmí být žádná čárka. V případě, že by se hodnoty nevešly do jednoho řádku může se jednoduše pokračovat na dalším řádku. Výrokem READ můžeme proměnné přiřazovat jednu po druhé hodnoty DATA-řádku. Výrok READ, který proměnné KM přiřazuje hodnoty vyhlíží takto:

```
10 PRINT "PREPOCET KILOMTRU NA MILE"
20 READ KM
30 LET M=KM/1.609344
40 PRINT KM; "KILOMTRY ODPOVIDAJI"
50 PRINTM; "MILE"
60 GOTO 20
70 DATA 23,24, 545,3,323,53,9
```

Program má však jeden nedostatek. Po přečtení poslední hodnoty, která je k dispozici, zkouší program najít další hodnotu. Jelikož tam žádná není, program se přeručí s hlášením chyby. Tomu se dá zabránit, zadáme-li jako poslední hodnotu DATA-řádku určité hodnoty. Výrokem IF / THEN dá se pak program při této hodnotě zastavit.

```
10 PRINT "PREPOCET KILOMETRU NA MILE"  
20 READ KM  
30 IF KM=999999 THEN END  
40 LET M=KM/1.609344  
50 PRINT KM,"KILOMETRY ODPOVIDAJI"  
60 PRINT M, "MILIM"  
70 GOTO 20  
80 DATA 23,34,545,3, 323,53,9,999999
```

Po přečtení všech hodnot je možnost nastavit čítač DATA opět na první hodnotu DATA-řádku. Pro to slouží výrok:

RESTORE

Následující program vydává hodnoty DATA-řádku do nekonečna:

```
10 READ Z  
20 IF Z=999 THEN RESTORE:GOTO 10  
30 PRINT Z  
40 GOTO 10  
50 DATA 6,12,18,24,30,36,42,48,54,60,999
```

Navíc k výroku RESTORE může být přidáno ještě číslo určité řádky. Počítač pak nasadí čítač na začátek DATA-řádky, která udává číslo za výrokem RESTORE. Např. RESTORE 120

Tato řádka nasadí čítač DATA na první hodnotu řádku 120.

Výrokem READ dají se však vkládat i řady znaků do řetězcové proměnné

```
10 DIM OVOCE$ (20)  
20 READ OVOCE$  
30 IF OVOCE$="END" THEN END  
40 PRINT OVOCE$  
50 GOTO 20  
60 DATA JABLKA,MERUNKY,BANANY,HRUSKY,END
```

V následujícím programu budou čteny současně z DATA-řádku dvě hodnoty pokynem READ. Toho se dosáhne tím, že zadáme za výrokem READ dvě proměnné. Může se použít i více než dvou proměnných. Při tom se musí každá proměnná oddělit od další čárkou.

```
10 DIM O$ (20), OVOCE$(20)  
20 PRINT "ZADEJTE DRUH OVOCE"  
30 INPUT O$  
40 RESTORE  
50 READ OVOCE$,CENA  
60 IF OVOCE$="END" THEN GOTO 100  
70 IF O$-OVOCES THEN PRINT "CENA OBNASI" ,CENA, "KCS": GOTO 20  
80 GOTO 50  
100 PRINT "SPATNY DRUH": GOTO 20  
120 DATA JABLKA, 1.34,MERUNKY,1.99,BANANY,0.78, HRUSKY,1.13  
130 DATA END,0
```

Nula na konci řádku 130 je důležitá. Počítač po pokynu READ čte vždy dvě hodnoty z DATA řádku. Chyběla-li by nula, nemohl by počítač přiřadit proměnné CENA žádnou hodnotu a program by se s hlášením chyby přerušil.

Indexované proměnné

Indexovaná proměnná je číselná proměnná, které může být přiřazeno více než jedna hodnota. Tak zvaný index umožňuje přístup jednoznačně na každou hodnotu. Jako index slouží každé celé číslo. Název indexované proměnné se stává ze dvou částí.

První část názvu tvoří se úplně stejně jako název číselné proměnné. K tomuto názvu připojí se ještě v závorkách index.

Proměnná A(5) je např. indexovaná proměnná. Každému místu v paměti A(0) A(1) A(2) A(3) A(4) A(5) může být zadána jedna hodnota. Než bude v programu použita indexovaná proměnná, musí být tato dimenzována výrokem DIM. Tento je znám již z řetězcové proměnné. Pokyn v programu 10 DIM A(5) říká, že dále může být proměnné A zadáno šest hodnot. Každé jednotlivé paměťové místo je charakterizováno indexy od 0 do 5. Následující program zadává šest čísel do indexované proměnné POLE a pak vytiskne na obrazovce číslo, příslušející ke každému indexu:

```
10 DIM POLE (5)
20 FOR MISTO=0 TO 5
30 INPUT X
40 LET POLE (MISTO)=X
50 NEXT MISTO
60 FOR MISTO=0 TO 5
70 PRINT "POLE(";MISTO;")=";POLE(MISTO)
80 NEXT MISTO
```

Počítač rozeznává mezi proměnnými POLE, POLE 5 a POLE(5).

Všechny mohou být v jednom programu vedle sebe použity. Indexované proměnné nazývají se anglicky array. Zatím měly proměnné jen jeden index. Označují se jako jednorozměrné arrays (soustavy) nebo jako vektor. Kromě toho se vyskytují ještě dvourozměrné arrays (soustavy), které nazýváme maticemi(matrix).

Matrice je indexovaná proměnná se dvěma indexy.

Při dimenzování a přístupu do matrice musí být udány dva indexy, které jsou odděleny čárkou. Dimenzování matrice vypadá např. takto:

```
DIM B(3,2)
```

Proměnná B se stává nyní ze 4 krát 3 míst v paměti (musí být brán v úvahu index 0), které mohou být osloveny.

B(0,0)	B(0,1)	B(0,2)
B(1,0)	B(1,1)	B(1,2)
B(2,0)	B(2,1)	B(2,2)
B(3,0)	B(3,1)	B(3,2)

Pomocí matrice se dá např. programovat jednoduchá skladová agenda.

Program vede v evidenci šest druhů s čísly druhů 0 až 5.

Pro každý druh se zaznamenává cena a skladované množství.

```
10 DIM DR(5,1)
20 PRINT "===NAZEV DRUHU==="
30 FOR DR=0 TO 5
40 PRINT :PRINT "DRUH CIS.";DR
50 PRINT "CENA ";:INPUT CE
60 DR (DR,0)=CE
70 PRINT "SKLADOVANE MNOZSTVI ";:INPUT SK
80 DR (DR,1)=SK
90 NEXT DR
100 PRINT :PRINT "+++PRODEJ+++"
110 PRINT :PRINT "DRUH CIS." ,:INPUT DR
120 IF DR<0 OR DR>5 THEN 110
130 PRINT "KUSU ";:INPUT KU
140 IF DR (DR,1)-KU<0 THEN PRINT "NA SKLADE NENI DOSTATECNE MNOZSTVI!";GOTO 110
150 LET CE=DR (DR,0)*KU
160 LET DR (DR,1)=DR(DR,1)-KU
170 PRINT "SOUCET: ";CE;" KCS"
180 PRINT "JESTE ";DR(DR,1);" KUSU NA SKLADE"
190 GOTO 110
```

GOSUB/RETURN

Pokyn GOSUB je blízce spřízněn s výrokem GOTO. Narazí-li počítač v programu na výrok GOSUB, bude průběh programu pokračovat na řádce, jejíž číslo je uvedeno za GOSUB. Na rozdíl od výroku GOTO pamatuje si počítač, na kterém místě došlo ke GOSUB. Následující pokyn RETURN způsobí, že v programu bude pokračováno za výzvou GOSUB. Pokynem GOSUB/RETURN může být vyvolána určitá část programu, která je častěji potřebována, z různých řádek. Část programu, která je uzavřena pokynem RETURN, se nazývá podprogram. V následujícím příkladu programu začíná podprogram za řádkou 100. Je vyvolán dvakrát.

```
10 PRINT "PRVNI VYZVA"
20 GOSUB 100
30 PRINT "DRUHA VYZVY"
40 GOSUB 100
50 PRINT "KONEC PROGRAMU"
60 END
100 FOR T=1 TO 3
110 PRINT "TOTO JE PODPROGRAM"
120 NEXT T
130 RETURN
```

Pokyn END v řádce 60 zabráňuje tomu, aby průběh programu po druhém vyvolání podprogramu v řádce 100 pokračoval. Jinak by počítač v řádce 130 narazil na RETURN, aniž by byl předtím podprogram vyvolán. To by mělo za následek hlášení chyby.

ON/GOTO

Pokynem ON/GOTO můžeme na základě hodnoty některé proměnné odbočiti do určitého čísla řádku. V programu se např. může vyskytnout řádek:

```
50 ON X GOTO 100,110,120,130
```

V tomto příkladě závisí na hodnotě proměnné X, ve kterém řádku bude v programu pokračováno. Bude-li X rovno jedné, odbočí program do řádku 100,

bude-li X rovno dvěma, skočí program do řádku 110, atd. Za výrokem ON/GOTO může stát tolik čísel řádků, kolik jich vejde do programové řádky. Maximální hodnota proměnných se řídí pak počtem řádků. U proměnné se musí jednat o číselnou proměnnou. Řetězcovou proměnnou počítač nebude akceptovat.

Příklad programu:

```
10 PRINT :PRINT "1: RADEK 100"  
20 PRINT "2: RADEK 200"  
30 PRINT "3: RADEK 300"  
40 PRINT "4: RADEK 400"  
50 PRINT "5: RADEK 500"  
60 PRINT "UDEJTE CISLO OD 1 DO 5"  
70 INPUT X  
80 IF X<1 OR X>5 THEN PRINT "NEPLATNE ZADANI":GOTO 10  
90 ON X GOTO 100,200,300,400,100  
100 PRINT "RADEK 100":GOTO 10  
200 PRINT "RADEK 200":GOTO 10  
300 PRINT "RADEK 300":GOTO 10  
400 PRINT "RADEK 400":GOTO 10  
500 PRINT "RADEK 500":GOTO 10
```

ON/GOSUB

ON/GOSUB se používá stejně jako výrok ON/GOTO. Počítač si však pamatuje jako při "normálním" GOSUB číslo řádku, ve kterém došlo k vyvolání podprogramu. Po výroku RETURN bude v programu pokračováno v řádku za pokynem ON/GOSUB. Program z kapitoly o ON/GOTO pokynu může být modifikován následovně. Řádky 100 až 200 stanou se podprogramy.

```
10 PRINT :PRINT "1: RADEK 100"  
20 PRINT "2: RADEK 200"  
30 PRINT "3: RADEK 300"  
40 PRINT "4: RADEK 400"  
50 PRINT "5: RADEK 500"  
60 PRINT "ZADEJTE CISLO OD 1 DO 5"  
70 INPUT X  
80 IF X<1 OR X>5 THEN PRINT "NEPLATNE ZADANI":GOTO 10  
90 ON X GOSUB 100,200,300,400,500  
95 GOTO 10  
100 PRINT "RADEK 100":RETURN  
200 PRINT "RADEK 200":RETURN  
300 PRINT "RADEK 300":RETURN  
400 PRINT "RADEK 400":RETURN  
500 PRINT "RADEK 500":RETURN
```

GRAFIKA

V následujících kapitolách bude vysvětleno, jak lze ovlivnit barvu a vzhled objektů na obrazovce. Dále bude objasněno, jak budeme moci obrázky malovat sami. Při použití grafiky je možno volit z 16 různých okruhů provedů (grafických stupňů). První tři stupně slouží k zobrazování textů, zbývajících třináct stupňů se používá ke kreslení obrázků v různých grafikách s jemným rozlišením. Tři grafické stupně 0 až 2 umožňují zobrazování textů v různých barvách a velikostech. První stupeň je možno dosáhnout následujícím pokynem:

GRAPHICS 0 nebo GR.0

Obrazovka vyhlíží nyní ještě stále tak, jak jsme zvyklí. Pomocí uvedeného pokynu se můžeme vrátit do základního módu z jiných grafických stupňů. V grafickém stupni 0 se vyvolají texty na obrazovku pomocí pokynu PRINT. Příklad: PRINT "HALLO"

Vyvolaný obraz je identický s obrazem dle kapitoly PRINT. Na obrazovce lze mimo jiné vytvořit také celou řadu grafických znaků. Toho dosáhneme pomocí klávesy CONTROL, kterou je nutno zmáčknout současně s klávesou, označenou grafickým znakem.

Příklad: Print "S". Při stlačení klávesy S musí být současně stlačena klávesa CONTROL. Na obrazovce se pak objeví: PRINT "+"

Po stlačení klávesy RETURN se na obrazovce objeví již jen grafický znak +

READY

Nyní je možné kreslit pomocí grafických znaků jednoduché grafické obrazce. Příklad:

```
10 PRINT "HJ"           10 PRINT "  "
20 PRINT "VB"           20 PRINT "  "
30 PRINT "MM"           30 PRINT "  "
```

Kdybychom si přáli experimentovat s grafickými znaky je nutno stisknout současně klávesy CONTROL a CAPS. Aretaci klávesy CONTROL můžeme ukončit současným stisknutím kláves SHIFT a CAPS.

Grafické stupně 1 a 2 lze vyvolat přesně stejným způsobem jako grafický stupeň 0, přičemž může být obrazovka rozdělena do dvou částí.

GRAPHICS 1

Větší část obrazovky je nyní černá, jen ve spodní části se nachází tzv. textové okénko. Zadáme-li nyní pokyn PRINT, objeví se text v textovém okénku. Abychom nyní dostali text na černé pozadí musí být pokyn PRINT rozšířen o znaky: #6;

Příklad: PRINT #6;"ICH KANN ES"

Text nyní musí být viditelný v oranžové barvě na černém pozadí. Vyzkoušejme nyní totéž v grafickém stupni 2.

GRAPHICS 2

GR.2

PRINT #6; "ICH KANN ES"

Text se nám zobrazí opět nahoře na obrazovce. Když nyní zadáme tento text dle pokynu PRINT po jednorázovém stlačení klávesy INVERSE, objeví se na obrazovce text v barvě modré. Text se může na obrazovce objevit i v barvě zelené, je-li zadán malým písmem.

Příklad: PRINT #6; "MODRÝ TEXT"
PRINT #6; "zelený text"

Kombinací klávesy INVERSE a malého písma lze na stínítko přivést ještě další barvu. V kapitolách COLOR, SETCOLOR, PEEK a POKE bude ještě objasněno, jak lze na obrazovce psát další texty v libovolných barvách.

Grafické stupně 1 a 2 mohou být vyvolány i bez textového okénka. To je však možné jen v programovém módu. Dosáhne se toho přičtením čísla 16 k danému grafickému stupni.

Příklad: 10 GRAPHICS 2+16
20 PRINT #6; "TOTO JE GRAFIKA 2"
30 PRINT #6; "bez textoveho okenka"
40 GOTO 40

Program se přeručí stlačením klávesy BREAK. Objeví se zase grafický stupeň 0.

Abychom nyní napsali text na určité místo stínítka, musí se použít již známého způsobu POSITION. K tomu je třeba vědět, kolik je v každém grafickém stupni volitelných poloh.

	Vodorovně	Svisle	Svisle (bez textového okénka)
GR.0	0-39	-	0-23
GR.1	0-19	0-19	0-23
GR.2	0-19	0-9	0-11

Příklad: 10 GRAPHICS 2+16
 20 FOR Y=0 TO 11
 30 POSITION 6,Y
 40 PRINT#6; POS.6,";Y
 50 FOR W=0 TO 100:NEXT W
 60 NEXT Y
 70 FOR W=0 TO 500:NEXT W
 80 END

Mějme však na zřeteli, abychom při způsobu POSITION nevolili příliš velká čísla. Příklad:

```
10 GRAPHICS 1
20 POSITION 20,12
RUN
ERROR - 141 AT LINE 20
READY
```

V tomto případě je číslo 20 příliš velké, jelikož grafický stupeň 1 připouští jen polohy od 0 do 19.

GRAPHICS 3-8

Grafické stupně 3 až 8 umožňují kreslení a malování na obrazovku. Rozdíly mezi jednotlivými stupni jsou v rozdílných možnostech volby barvy a rozlišení /kontrastu/ obrázků. Jednotlivých grafických stupňů lze dosáhnout stejným pokynem jako u GRAPHICS 0-2. Grafické stupně 3,5 a 7 dovolují současně použít čtyř rozdílných barev, stupně 4 a 6 naproti tomu jen 2 barvy a stupeň 8 jen jednu barvu odlišenou dvěma stupni jasu. Stupeň 8 zato má nejvyšší rozlišení. V tomto režimu mohou být na obrazovce zobrazeny téměř vždy jednotlivé body obrazu. Všeobecně můžeme říci, že každý grafický stupeň má své přednosti a při jeho volbě záleží na problému, který chceme řešit. Např. chceme-li na obrazovku vynést statistický blokový přehled, doporučuje se stupeň 3, jelikož není třeba určovat příliš mnoho jednotlivých bodů, jako při stupni 8. Kromě toho máme k dispozici více barev. Chceme-li ale na obrazovku kreslit, doporučuje se použít vyšší grafický stupeň, protože dosáhneme větší detaily obrazu.

Abychom vůbec mohli využít grafických možností počítače ATARI, potřebujeme znát následující pokyny BASIC:

PLOT

Pokyn PLOT (zkráceně PL.) se používá při zobrazování grafiky místo pokynu POSITION. Tento pokyn nechá počítač vysadit bod na místě obrazovky, které je určeno dvěma čísly, následujícími za pokynem.

Příklad: 10 GRAPHICS 3
 20 COLOR 1
 30 PLOT 19,11

V uvedeném příkladu se vykreslí jediný bod ve středu obrazovky (Pokyn COLOR bude vysvětlen později). Obdobně jako při pokynu POSITION dochází u pokynu PLOT často k chybě tím, že jsou zvolena příliš velká čísla.

Jaký je rozsah volitelných bodů v jednotlivých grafických stupních můžeme zjistit z tabulky č. 765, která je situovaná za kapitolou GRAPHICS 9-11.

DRAWTO

Pomocí pokynu DRAWTO je možno na stínítku kreslit linie. Linie začíná v bodě daném pokynem PLOT a končí v bodě daném pokynem DRAWTO. Výchozím bodem pro další pokyn DRAWTO může být konečný bod předchozí linie. Příklady:

```
10 GRAPHICS 3
20 COLOR 1
30 PLOT 19,1
40 DRAWTO 34,16
50 DRAWTO 4,16
60 DRAWTO 19,1
```

Tento program dá na obrazovce trojúhelník. Pokyn DRAWTO může být zkrácen na DR.

```
Příklad: 10 REM DOMEK
20 GRAPHICS 4+16
30 COLOR 1
40 PL. 39,15
50 DR. 54,25: DR. 24,25: DR. 39,15
60 PL. 54,25
70 DR. 54,45: DR. 24,45: DR. 24,25
80 GOTO 80
```

Dle řádku 80 probíhá program do nekonečna a obrázek setrvává na obrazovce. Program lze zastavit pomocí klávesy BREAK.

Nahradíme-li řádek 80 takto:

```
80 FOR I=1 TO 750:NEXT I
```

Zpozdí se konec programu. Tyto obě možnosti jsou důležité především tehdy, když vypneme textové okénko, neboť obraz po skončení programu zmizí.

Následující program umožňuje kreslit na obrazovku libovolné tvary. Vyjadřuje, že určité obrazce lze vypočítat ze vhodných vzorců. (Pokyn POKE bude vysvětlen na dalších stránkách).

```
10 REM KRUH V GR.8
20 GRAPHICS 8+16
30 POKE 709,14
40 COLOR 1
50 REM VYPOCET KRUHU
60 FOR KS=1 TO 400
70 LET I=I+0.05
80 LET X=SIN(I)* 50: LET Y=COS(I)* 45
90 PLOT X+160, Y+85
100 NEXT KS
110 GOTO 110
```

Dalším příkladem je výpočet a vykreslení elipsy:

```
10 REM ELIPSA V GR.8
20 GRAPHICS 8+16
30 POKE 709,14
40 COLOR 1
50 REM VYPOCET ELIPSY
60 FOR ES=1 TO 400
70 LET I=I+0.05
80 LET X=SIN(I)* 80: LET Y=COS(I)* 15
90 PLOT X+160, Y+85
100 NEXT ES
110 GOTO 110
```

SETCOLOR

Ve všech grafických stupních máme možnost malovat nebo psát v různých barvách. S touto možností jsme se seznámili již v kapitole GRAPHICS 0-2. V režimu SETCOLOR však můžeme ovlivnovat podstatně více stupňů sytosti barev.

V každém grafickém stupni má počítač k dispozici více barevných registrů, jejichž obsah je možno měnit. V každém stupni však nelze používat všech barevných registrů. Příklad:

GRAPHICS 0

Barevný registr	Výchozí barva	Použití
0	-	-
1	světle modrá	barva pro psaní
2	tmavě modrá	barva pro pozadí
3	-	-
4	černá	barva okraje

Chceme-li nyní změnit barvu okraje obrazovky (rámečku) vložíme např. SETCOLOR 4,2,10 Při tom platí:

1. číslo určuje barevný registr, který chceme změnit. V GR.0 je vždy barva písma stejná s barvou pozadí. Dá se měnit jen jejich kontrast.

2. číslo určuje barvu.

3. číslo určuje jas barvy.

Druhé a třetí číslo mohou nabývat hodnotu od 0 do 15.

Barva (Jas 10)	Číslo barvy
šedá	0
zlatá	1
oranžová	2
červeně-oranžová	3
růžová	4
nachová	5
nachově modrá	6
modrá	7
modrá	8
světle modrá	9
tyrkysová	10
modrozelená	11
zelená	12
žlutozelená	13
oranžově zelená	14
světleoranžová	15

U hodnot jasu (třetí číslo) je 0 nejtmaší a 15 nejjasnější stupeň. Dostáváme tak 16 barev v 16 stupních jasu, tedy 256 barev.

Příklad:

```
10 GR. 1+16
20 POS. 5,12
30 PRINT# 6; "LIGHTSHOW"
40 FOR F=0 TO 15
50 FOR H=2 TO 10 STEP 2
60 SETCOLOR 4,F,H
70 FOR W=1 TO 80:NEXT W
80 NEXT H
90 NEXT F
100 GOTO 40
```

Program pro současné zobrazení všech 256 barev je uveden v cvičných příkladech. Není-li udán žádný pokyn SETCOLOR, používá počítač v jednotlivých grafických stupních vždy stejných výchozích barev.

COLOR

Tento pokyn slouží k tomu, aby byl zvolen pro malování určitý barevný registr. Nacházíme-li se např. v grafickém stupni 3, lze volit ze čtyř barevných registrů.

GRAPHICS 3

<u>Barevný registr</u>	<u>Výchozí barva</u>	<u>Použití</u>
0	oranžová	grafické body
1	světlezelená	grafické body
2	tmavomodrá	grafické body
3	-	-
4	černá	pozadí, okraj

Budeme-li chtít přiřadit určitou barvu pokynu PLOT nebo DRAWTO, docílíme toho pomocí pokynu COLOR. Příklad:

```
10 GR.3
20 COLOR 1
30 PLOT 10,4: DRAWTO 27,4
40 COLOR 2
50 PLOT 10,8: DRAWTO 27,8
60 COLOR 3
70 PLOT 10,12: DRAWTO 27,12
```

V uvedeném příkladu vidíme, že barevné registry, zvolené pomocí pokynu COLOR, nejsou voleny stejnými čísly, jako u pokynu SETCOLOR. To může v počátku poněkud mást, ale když už víme, jak jsou sestaveny hodnoty pokynu COLOR, neměl by to být problém. COLOR 0 volí v každém grafickém stupni barvu pozadí. Ostatní registry jsou očíslovány v pořadí od 1 do 3.

GRAPHICS 3

<u>Barevný registr</u>	<u>Výchozí barva</u>	<u>Hodnota COLOR</u>
0	oranžová	COLOR 1
1	světlezelená	COLOR 2
2	tmavomodrá	COLOR 3
3	-	-
4	černá	COLOR 0

Pokynu COLOR však nelze používat v grafických stupních 0-2. Vynecháme-li jej v ostatních grafických stupních, nasadí počítač automaticky COLOR 0 (barva pozadí) jako barvu každého znaku na stínítku.

Příklad:

```
10 LET X=0: LET Y=0: LET M=20: LET N=12
20 FOR I=3 TO 7 STEP 2
30 GRAPHICS I+16
40 SETCOLOR 0,0,0
50 SETCOLOR 1,15,14
60 REM CTYRUHELNIK
70 COLOR 1
80 PLOT 8+X, 5+Y
90 DRAWTO 30+X, 5+Y: DRAWTO 30+X, 16+Y
100 DRAWTO 8+X, 16+Y: DRAWTO 8+X, 5+Y
110 REM TROJUHELNIK
```

```

120 COLOR 2
130 PLOT 19+X, 7+Y
140 DRAWTO 28+X, 14+Y: DRAWTO 10+X, 14+Y: DRAWTO 19+X, 7+Y
150 LET X=M: LET Y=N: LET M=60: LET N=36
160 REM BAREVNY PRUBEH
170 FOR F=0 TO 15
180 LET F1=15-F
190 FOR H=0 TO 14 STEP 2
200 LET H1=14-H
210 SETCOLOR 0,F,H
220 SETCOLOR 1,F1,H1
230 FOR WAIT=1 TO 80: NEXT WAIT      (WAIT = čekej)
240 NEXT H
250 NEXT F
260 NEXT I
270 GOTO 10

```

Tento program lze zastavit jen stlačením klávesy BREAK. Znáznornuje rozlišení tří grafických stupňů, ve kterých jsou k dispozici vždy čtyři barevné registry.

PEEK/POKE

Pokud jsme chtěli měnit barvy v barevných registrech, používali jsme dosud pokyn SETCOLOR. Existuje však ještě i jiná možnost, a to je pokyn POKE.

Příklad: POKE 710,22

Zjistíme, že pozadí má nyní zlatou barvu.

Při pokynu POKE platí první číslo pro barevný registr. Následující přehled ukazuje, která hodnota POKE nahraňuje který barevný registr.

GRAPHICS	SETCOLOR	Použití	Registr POKE
0	2	pozadí	710
	4	okraj	712
	1	znaky	709
1,2	2	textové okénko (pozadí)	710
	1	znaky v text. okénku	709
	4	pozadí	712
	0	znaky	708
3,5,7	4	pozadí	712
	2	textové okénko (pozadí)	710
	1	znaky v text. okénku	709
	0	grafické body	708
4,6	4	pozadí	712
	0	grafické body	708
	1	znaky v okénku	709
	2	textové okénko (pozadí)	710
8	2	pozadí	710
	1	grafický bod	709
	4	okraj	712

Druhé číslo pokynu POKE udává barvu a její hodnotu jasu. Existuje 16 barev v 16 různých stupních jasu, což dává celkem 256 barev.

Pro výpočet hodnoty POKE pro určitou barvu vynásobme požadované číslo barvy číslem 16 a k tomu připočteme hodnotu požadovaného stupně jasu. Příklad: Barva pozadí v bleděmodrém (číslo barvy 9) ve stupni jasu 11, $9 * 16 + 11 = 155$ POKE 710,155
Nejlépe bude, když si znázorníme jednotlivé barvy a jejich hodnoty jednorázově pomocí následujícího programu:

```
10 GRAPHICS 1
20 FOR X=0 TO 255
30 PRINT X
40 POKE 712,X
50 FOR W=1 TO 50: NEXT W
60 NEXT X
```

Tento program lze zastavit v každém místě držíme-li klávesu CONTROL trvale stlačenou a navíc zmáčkeme dodatečně klávesu 1. Chceme-li, aby program pokračoval, zmáčkeme opakovaně obě klávesy.

Pokyn PEEK je komplement pokynu POKE. Nemění žádný registr, nýbrž dotazuje se na skutečnou hodnotu registru.

V následujícím příkladu tážeme se na skutečnou hodnotu barvy registru pozadí v grafickém stupni 0 (PRINT je pro zobrazení na obrazovce).

```
PRINT PEEK (710)
```

```
148
```

```
READY
```

Existuje ještě další možnost využití pokynů POKE/PEEK. Existují určité registry používané např. za účelem změnění šířky okraje, nebo za účelem ovládní tiskárny programů (je-li připojena). Kromě toho mohou být jednotlivé klávesy testovány přes určité registry.

```
POKE 82,X   levý okraj, X udává šířku okraje
POKE 83,X   pravý okraj, X udává šířku okraje
POKE 755,4  převrací text a nechá kurzor zmizet
POKE 54018,52 startuje tiskárnu programů, je-li stlačená
klávesa PLAY
```

Další zajímavá paměťová místa jsou uvedena v dodatku.

GRAPHICS 9-11

Tyto tři grafické stupně umožňují volit obzvláště velký počet barevných variant obrazu při dobré rozlišovací schopnosti. Žádný z uvedených tří stupňů nemá textové okénka a jsou uvažovány jen pro čisté grafické aplikace.

GRAPHICS 9

Tento stupeň dovoluje vynést na obrazovku jednu barvu v 16 stupních jasu současně. Pokyn SETCOLOR vyhlíží následovně:

```
SETCOLOR 4 : < 0 - 15 > 0
pevná hodnota volitelně pevná hodnota
od 0 do 15
```

GRAPHICS 10

V tomto stupni může být umístěno na obrazovku devět barev v různém jasu. Je však poněkud obtížné naplnit barevné registry a pak je volit. K tomu nám může pomoci následující tabulka:

	Výstupní barva	Registr POKE
COLOR 0	černá	704
COLOR 1	černá	705
COLOR 2	černá	706
COLOR 3	černá	707

COLOR 4	oranžová	SETCOLOR 0,x,x,	708
COLOR 5	světлезelená	SETCOLOR 1,x,x,	709
COLOR 6	světлезelená	SETCOLOR 2,x,x,	710
COLOR 7	oranžová	SETCOLOR 3,x,x,	711
COLOR 8	černá	SETCOLOR 4,x,x,	712

x = libovolné hodnoty dle tabulky pro barvu a jas

GRAPHICS 11

V uvedeném stupni lze na obrazovku vynést současně 16 různých barev, avšak všechny v jednom stupni jasu. Barvy se volí pokynem COLOR, přičemž není volen žádný barevný registr nýbrž jen barva.

COLOR <0-15>

Hodnota jasu, která platí pro všechny barvy, se mění pokynem SETCOLOR.

SETCOLOR 4,0, <0-15>

Tabulka grafických stupňů, 765

GR.	použití	vodorovně	svisle	svisle (bez text. ok.)	počet barev
0	text	40	-	24	2
1	"-	20	20	24	5
2	"-	20	10	12	5
3	grafika	40	20	24	4
4	"-	80	40	48	2
5	"-	80	40	48	4
6	"-	160	80	96	2
7	"-	160	80	96	4
8	"-	320	160	192	1/2
9	"-	80	-	192	1/16
10	"-	80	-	192	9
11	"-	80	-	192	16
12	"-	40	20	24	5
13	"-	40	10	12	5
14	"-	160	160	192	2
15	"-	160	160	192	4

V tabulce uvedené výrazy 1/2 a 1/16 znamenají, že je použito jedné barvy ve dvou nebo 16-ti jasech.

SOUND

Kromě grafiky nabízí počítač ATARI množství zvukových a tónových možností. Jsou vyráběny pomocí čtyř na sobě nezávislých tónových kanálů/generátorů.

Abychom byrobili nějaký tón, je nutno použít pokynu SOUND se čtyřmi hodnotami. Příklad: SOUND 0, 29, 10, 10

Tento pokyn dává vysoké c. Při tom určuje první číslo tónový registr, t.j. který tónový generátor tón vyrábí. Může nabývat hodnot od 0 do 3: Příklad: SOUND 0, 50, 10, 6

SOUND 1, 100, 10, 6

SOUND 2, 150, 10, 6

SOUND 3, 200, 10, 6

Druhé číslo určuje TÓN. Jeho hodnota může ležet mezi 0 a 255.

Příklad: 10 FOR N=0 TO 255

20 PRINT N

30 SOUND 0, N, 10, 10

40 FOR W=1 TO 80:NEXT W

NEXT N

60 END

Třetí číslo určuje ZABARVENÍ. Hodnota může ležet mezi 0 a 14, přičemž 10 je čistý tón.

```
10 FOR T=0 TO 14 STEP 2
20 SOUND 0, 29, T, 10
30 FOR W=0 TO 80:NEXT W
40 NEXT T
50 GOTO 10
```

Čtvrté číslo určuje, dodatečně k regulátoru televizního přístroje, sílu zvuku. Může nabývat hodnot mezi 0 a 15.

```
10 FOR L=0 TO 15
20 SOUND 0, 29, 10, L
30 FOR W=0 TO 80:NEXT W
40 NEXT L
50 FOR L=15 TO 0 STEP-1
60 SOUND 0, 29, 10, L
70 FOR W=0 TO 80:NEXT W
80 NEXT L
90 GOTO 10
```

Dotazování řídicích zařízení

Jovstíck/Rídicí páka

Rídicí páka je jednoduše ovladatelné zařízení pro zadávání a proto je zvláště vhodné pro televizní hry. Aby bylo možno používat řídicí páky v programech, jsou v jazyku Basic ATARI pokyny pro hodnoty, dané stavem řídicí páky.

Páka - STICK

Na počítač ATARI lze napojit dvě řídicí páky. Lze se na ně dotazovat pomocí pokynu STICK.

Rídicí páka v levé zdířce je dotazována pomocí STICK (0) a páka v pravé zdířce pomocí STICK (1). Příklad: 10 PRINT STICK(0)

```
20 GOTO 10
```

Při použití tohoto příkladu vidíme, že registr řídicí páky může přijmout devět hodnot. Následující program ukazuje názorně, které hodnoty náleží ke které poloze. Příklad:

```
10 GRAPHICS 2+16
20 ST=STICK(0)
50 IF ST=14 THEN POSITION 9,1:GOTO 140
60 IF ST=13 THEN POSITION 9,9:GOTO 140
70 IF ST=11 THEN POSITION 4,5:GOTO 140
80 IF ST=7 THEN POSITION 14,5:GOTO 140
90 IF ST=10 THEN POSITION 4,1:GOTO 140
100 IF ST=9 THEN POSITION 4,9:GOTO 140
110 IF ST=5 THEN POSITION 14,9:GOTO 140
120 IF ST=6 THEN POSITION 14,1:GOTO 140
130 IF ST=15 THEN POSITION 9,5:GOTO 140
140 PRINT#6;ST
150 FOR W=1 TO 300:NEXT W
160 GOTO 10
```

STRIG

Tímto pokynem se lze dotázat na spoušť (trigger) řídicí páky. Stejně jako u pokynu STICK jsou knoflíky každé páky dotazovány pomocí STRIG(0) a STRIG(1). Při dotazu mohou být v registru hodnoty 0 a 1. Hodnota 0 je pro stisknutý stav a hodnota 1 pro klidový stav.

Příklad: 10 PRINT STRIG(0)

```
20 GOTO 10
```

Následující program ukazuje, jak lze na obrazovce pomocí řídicí páky kreslit. Při stlačení spoušti lze na obrazovku nakreslit čáru. Čára je jinak kreslena v bravě pozadí a tudíž ji

není vidět. Této funkce lze tedy také používat ke korekturám.

```
10 GRAPHICS 7+16:X=80:Y=40
20 ST=STICK(0)
30 IF ST=7 THEN X=X+1
40 IF ST=6 THEN X=X+1:Y=Y-1
50 IF ST=14 THEN Y=Y-1
60 IF ST=5 THEN X=X+1:Y=Y+1
70 IF ST=11 THEN X=X-1
80 IF ST=10 THEN X=X-1:Y=Y-1
90 IF ST=13 THEN Y=Y+1
100 IF ST=9 THEN X=X-1:Y=Y+1
110 IF X > 159 THEN X=159
120 IF Y > 95 THEN Y=95
130 IF X < 0 THEN X=0
140 IF Y < 0 THEN Y=0
150 COLOR 1
160 PLOT X,Y
170 FOR W=1 TO 20:NEXT W
180 COLOR 0
190 IF STRIG(0)=0 THEN COLOR 1
200 PLOT X,Y
201 GOTO 20
```

Páčkový otočný ovladač/otočný regulátor - PADDLE CONTROL/DREHREGLER

Toto zařízení je stejně jako řídicí páka zařízením pro zavádění hodnot. Má tu přednost, že jím lze zadat více než devět hodnot do počítače. Dle polohy páčky ovladače zadá se hodnota mezi 0 a 228. Příklad: 10 PRINT PADDLE(0)

```
20 GOTO 10
```

Z toho vychází čtyři možnosti programování, např. pro jednoduchý hudební program. 10 LET PA=PADDLE(0)

```
20 SOUND 0,PA,10,10
```

```
30 GOTO 10
```

Spoušť páčkového otočného ovladače může mít uloženy v registru, stejně jako řídicí páka, hodnoty 0 a 1. Na tyto hodnoty se dotazujeme pomocí pokynu PTRIG. Příklad: 10 SETCOLOR 2,PA1,2:COLOR 1

```
20 LET PA=PADDLE(0):LET
```

```
PA 1=PA/10
```

```
30 POKÉ 82,PA1+7:PRINT PA
```

```
40 SOUND 0,PA,10,10
```

```
50 IF PTRIG(0)=0 THEN SOUND
```

```
0,PA,6,PA1
```

```
60 GOTO 10
```

HLÁŠENÍ CHYB A VYSVĚTLIVKY

U počítače ATARI se na obrazovce objeví kódové číslo v případě, že se vyskytne chyba. V dalším odstavci je vysvětlen význam těchto chyb.

2. Není k dispozici dostatek místa v paměti

Není k dispozici v paměti dostatek místa k uložení programu nebo proměnných nebo vloženo jedna do druhé příliš mnoho vnitřních smyček FOR-NEXT nebo podprogramů.

3. Nesprávná hodnota

Číselná hodnota je příliš velká nebo příliš malá, nebo na místě kde by měla být kladná, je záporná.

4. Příliš mnoho proměnných

V jednom programu nesmí být použito více než 128 různých proměnných. Tato mez je za určitých okolností překračována i proto, že k celkovému počtu proměnných se připočítávají i dříve použité proměnné, kterých se však mezitím přestalo používat.

5. Překročení délky alfanumerické proměnné

Je oslovena část alfanumerické proměnné, pro kterou proměnná nebyla dimenzována.

6. Příliš málo hodnot dat

Došlo k pokusu číst pomocí výroku READ více hodnot, než bylo udáno výrokem DATA.

7. Číselná hodnota větší než 32767

Číselná hodnota je v místě, kde takovou nesmí být, větší než 32767 nebo menší než 0.

8. Chybný vstup

Při provádění výroku INPUT bylo zjištěno, že udaný druh proměnné a hodnota, která byla této proměnné přidělena, si neodpovídají. Číselná proměnná nesmí např. obsahovat žádná písmena, větné nebo grafické znaky.

9. Chybné dimenzování

Výrok DIM obsahuje pole nebo alfanumerickou proměnnou, které již byly dimenzovány nebo pole, které zaujímá více než 32767 bytů. Nebo došlo k pokusu přístupu do pole, které dosud nebylo dimenzováno nebo na alfanumerickou proměnnou, která nebyla dosud dimenzována.

10. Příliš komplexní výraz

Výraz obsahuje příliš mnoho závorek nebo příliš mnohonásobně složené funkce.

11. Číslo příliš velké

Došlo k pokusu dělit nulou nebo se vyskytne číslo, jehož absolutní hodnota je větší než 999999999×10^{97}

12. Nelze najít řádek

U výroku GOSUB, GOTO, IF-THEN, ON-GOSUB- nebo ON-GOTO je uvedeno číslo řádku, který se v programu nevyskytuje.

13. Sehází výrok FOR

Narazilo se na výrok NEXT, ke kterému chybí odpovídající výrok FOR. Je možné, že došlo ke zkřížení dvou smyček FOR-NEXT.

14. Řádek příliš dlouhý

Výrok je příliš komplexní, nebo přesahuje délku logické řádky.

15. Řádka s výrokem GOSUB- nebo FOR byla smazána

Při provádění výroku RETURN nebo NEXT nemohla být nalezena řádka, ve které se nachází odpovídající výrok GOSUB nebo FOR.

16. Chybí výrok GOSUB

Narazilo se na výrok RETURN, ačkoliv dosud nebyl prováděn žádný výrok GOSUB.

17. Výrok není proveditelný

Výrok byl chybou v paměti, výrokem POKE nebo strojním programem změněn tak, že nemůže být identifikován a tudíž proveden.

18. Nesprávný znak

Došlo k pokusu pomocí funkce VAL převést alfanumerickou proměnnou na číselnou hodnotu, ačkoliv v této proměnné jsou obsaženy znaky, které nemohou být převedeny.

19. Program příliš dlouhý

Došlo k pokusu zavést program, který zaujímá více místa, než je v počítači k dispozici.

20. Špatné číslo kanálu

Došlo k pokusu použít kanálu 0 nebo bylo udáno číslo kanálu větší než 7.

21. Zavádění pomocí LOAD není možné

Došlo k pokusu zavádět pomocí výroku LOAD data nebo program, uložený do paměti pomocí výroků CSAVE nebo LIST.

128. Přerušeni klávesou BREAK
Během V/V operace byla stlačena klávesa BREAK
129. Kanál již otevřen
Došlo k pokusu otevřít kanál, který již byl otevřen. K provedení grafických výroků byl např. otevřen kanál 6 a k provedení několika jiných výroků použít kanál 7. Při výskytu této chyby může dojít za jistých okolností k automatickému uzavření kanálu, který chybu způsobil.
130. Neznámé zařízení
Došlo k pokusu o přístup na neznámý přístroj.
131. Kanál otevřen jen pro výstup
Bylo použito výroku GET nebo INPUT v souvislosti s V/V kanálem, který byl otevřen jen pro výstup.
132. Výrok XIO chybný
Při provádění výroku XIO se vyskytla chyba.
133. Kanál není otevřen
Došlo k pokusu použít kanál V/V, který dosud nebyl otevřen.
134. Špatné číslo kanálu
Lze použít pouze V/V kanálů s čísly 1,2,3,4,5,6 a 7.
135. Kanál otevřen jen pro vstup
Bylo použito výroku PUT nebo PRINT v souvislosti s V/V kanálem, který je otevřen jen pro vstup.
136. Dosaženo konce programového bloku
Byl čten záznam o konci programového bloku anebo byl učiněn pokus o čtení sektoru diskety, který náleží do neotevřeného bloku.
137. Záznam příliš dlouhý
Počítač nemohl úplně převzít záznam, jelikož byl delší než 256 Bytů.
138. Přístroj není oslovitelný
Udaný přístroj nebylo možno oslovit. Přesvědčte se, je-li přístroj zapojen, zda všechna propojení jsou řádně provedena a zda všechny přepínače s polohou "ONLINE" a "LOCAL" jsou v poloze "ONLINE".
139. Přístroj nepracuje bez závad
Zařízení pro záznam programu nebo stanice disket nepracuje nezávadně nebo nemůže provést určitý výrok.

140. Chyba na sériové sběrnici
Pravděpodobně je vadná kazeta nebo disketa.
141. Kurzor mimo okraj obrazovky
142. Chyba formátu při přenosu dat po sériové sběrnici
143. Chyba parity při přenosu dat po sériové sběrnici
Při přenosu dat po sériové sběrnici vyskytla se chyba. Kazeta nebo disketa nemohla být popsána nebo čtena. Pravděpodobně kazeta nebo disketa vadná.
144. Chyba diskety
Disketa je chráněna proti zápisu nebo v directory se vyskytla chyba.
145. Chyba záznamu na disketě nebo chyba zobrazování na obrazovku
Byl zjištěn rozdíl mezi tím, co mělo být zapsáno na disketě a tím, co bylo skutečně zapsáno nebo byla zjištěna chyba v souvislosti se zobrazováním na obrazovku.
146. Funkce není proveditelná
Byl učiněn pokus provést neproveditelnou funkci, např. zavést data na klávesnici nebo číst data z tiskárny.
147. Rozsah paměti nestačí pro zvolený grafický způsob provozu
Pro různé grafické módy provozu zaplňují nestejnou kapacitu paměti.
150. Sériové místo styku je již obsazeno
Pro každé určité sériové místo styku je možno otevřít jen jeden V/V kanál.
151. Místo styku nebylo otevřeno pro současný vstup/výstup
Předtím než mohou být data přes sériové místo styku vysílána nebo přijímána, musí být toto místo styku otevřeno pro současný vstup/výstup pomocí výroku XIO-40.
152. Chyba buffru při současném vstupu/výstupu
Údaje vztahující se na buffer nebyly konkrétní.
153. Aktivován současný vstup/výstup
Došlo k pokusu vkládat nebo vydávat data přes sériové místo styku v době, kdy bylo otevřeno jiné místo styku pro současný vstup/výstup.
154. Nebyl aktivován současný vstup/výstup
Byl učiněn pokus provést V/V operaci ve vztahu k sériovému místu styku, při níž musí být toto místo styku otevřeno pro současný vstup/výstup (což se nestalo).

160. Neznámá disketová stanice

Je-li oslovena stanice disket, mohou se k tomu použít jen názvy přístroje D: D1: D2: D3: nebo D4:, datování, označení návěští, pojmenování (název), místo pro data.

161. Otevřeno příliš mnoho stránek

162. Na disketě nezbyvá v paměti žádné místo

Na disketě nezbyvá v paměti žádné místo, všechny sektory jsou obsazeny.

163. Chyba systému

Během vkládání nebo vydávání dat byla zjištěna chyba, která nebyla odstraněna a jejíž původ nebyl zjištěn.

164. Chyba ve stránce/sektoru

Pointer, čítač stránek byl pomocí výroku POINT nasazen na sektor, který nenáleží k otevřené stránce nebo spojení mezi jednotlivými sektory jsou porušena.

165. Chybný název stránky

Název stránky začíná malým písmenem nebo obsahuje nepřípustné znaky. Je také možné, že znak pro označení volné karty nebyl správně použit.

166. Výrok POINT chybný

Byte udané výrokiem POINT neexistuje.

167. Stránka je chráněna proti zápisu

Stránky jsou chráněné proti zápisu nemohou být popsány ani smazány, ani nemůže být změněn název těchto stránek.

168. Neznámý výrok XIO

Došlo k pokusu provést výrok XIO, který neexistuje.

169. V directory není místa

V directory může být zaneseno 64 názvů stránek v závislosti na tom, kolik je ještě místa na disketě.

170. Stránka nebyla nalezena

Na disketě uložení v udané stanici disket, nebyla nalezena žádná stránka s udaným názvem.

171. Výrok POINT neproveditelný

Došlo k pokusu vstoupit do sektoru, který nenáleží k otevřené stránce.

ČASTO POUŽÍVANÉ ADRESY PAMĚTI A JEJICH FUNKCE

Mnohé paměťové buňky plní zcela určité úkoly. V tomto odstavci jsou buňky paměti, které jsou pro programátora v BASIC zvláště zajímavé. Obsah buňky paměti lze číst pomocí výroku PEEK a pomocí výroku POKE lze obsah buňky paměti měnit.

V BASICu udávají se adresy buněk a jejich obsah decimálně. Buňky paměti jsou číslovány od 0 do 65535. Hodnota obsahu může ležet mezi 0 a 255. Mají-li se ukládat větší hodnoty, potřebujeme dvě za sebou ležící buňky paměti. Celkový souhrnný obsah těchto dvou buněk paměti se vypočítá ze součtu obsahu první buňky paměti a 256 násobku obsahu druhé buňky paměti. Aby se určil sloupec, ve kterém se momentálně nachází kurzor, musí být např. vypočítán $PEEK(85) + 256 * PEEK(86)$. Naopak, za účelem změny polohy kurzoru ve sloupcích musí být provedeny $POKE 85, SPAL - INT(SPAL/256) * 256$ a $POKE 86, INT(SPAL/256) * 256$. Kurzor se poté nachází ve sloupci SPAL.

17. Stlačena klávesa BREAK (BREAK Key/klávesa/ Flag/vlajka/ = BRKKEY)

Nachází-li se v této buňce 0, byla stlačena klávesa BREAK.

77. Změna barvy vyp/zap (Mód attract zap/vyp = ATTRACT)

Je-li v této buňce paměti uložena 0, je odpojeno řízení barvy na obrazovce. K tomu dochází vždy tehdy, když je stlačena nějaká klávesa klávesnice. Je-li však v této buňce paměti uloženo 254, změna barvy se zapojí. K tomu dochází automaticky, není-li stlačena po 9 minut žádná klávesa.

82. Levý okraj (ohraničené) obrazovky. Levý okraj plochy pro text = LMARGN)

Tato paměťová buňka obsahuje číslo sloupce, ve kterém se nachází v grafickém způsobu provozu 0 levé ohraničení obrazovky. PEEK (82) leží vždy mezi 0 a 39. 0 odpovídá levému kraji obrazovky. Za normálních poměrů je v této paměťové buňce uloženo 2.

83. Pravé ohraničení obrazovky (Pravý okraj plochy pro text = RMARGN)

Tato paměťová buňka obsahuje číslo sloupce, ve kterém se nachází v grafickém způsobu provozu pravé ohraničení obrazovky. PEEK (83) leží vždy mezi 0 a 39. 39 odpovídá pravému okraji obrazovky. Normálně je v této buňce uloženo 39.

84. Okamžitá poloha kurzoru (řádek) průběžná řádková poloha cursoru = ROWCRS)

Obsah této paměťové buňky odpovídá řádku obrazovky, od kterého (ve kterém) bude psán nebo čten následující znak. Minimální hodnota PEEK (84) je 0. Maximální hodnota závisí na tom, jaký grafický způsob provozu je aktivován.

85., 86. Okamžitá poloha kurzoru (sloupec) průběžná sloupcová poloha kurzoru = COLCRS)

Obsah této paměťové buňky odpovídá sloupci obrazovky, ve kterém nebo od kterého bude psán nebo čten následující znak. Minimální hodnota PEEK (85) je 0. Jak bude velká maximální hodnota závisí na tom, který

grafický způsob provozu je aktivován. V grafickém způsobu provozu 0 až 7 je PEEK (86) vždy 0.

88.,89. Adresa paměti stínítka (adr. pam. stín = SAVMSC)

Tato buňka paměti obsahuje nejspodnější (nejnižší) adresu paměti stínítka. Obsah této adresy je zobrazen v levém horním rohu stínítka.

94.,95. Adresa kurzoru (adresa paměti kurzoru = OLDADR)

V této paměťové buňce je uložena adresa kurzoru.

106. Horní konec paměti RAM (Byte nejvyšší hodnoty)

Vrcholová adresa RAM = RAMTOP

Obsah této adresy odpovídá 16-ti násobnému počtu 4K bloků RAM, na které je počítač zbudován. Počet 1K bloků může být vypočítán pomocí PEEK (740) /4.

186.,187. Číslo řádku, ve kterém byl program přerušeno.

(Číslo stop čáry = STOPLN)

Tyto paměťové buňky obsahují čísla řádku programu BASIC, ve kterém byl program přerušeno stlačením klávesy BREAK nebo výrokem STOP nebo ve kterém vyskytla se chyba.

195. Kódové číslo vyskytnuvší se chyby (číslo chyby = ERRSAV)

Vyskytne-li se chyba, uloží se kódové číslo chyby do paměťové buňky.

212.,213. Funkční hodnota funkceUSR. (Funkční hodnotaUSR=FRO)

V těchto paměťových buňkách může být uložena číselná hodnota pomocí funkceUSR, vyvolaného strojního programu, která bude při zpětném skoku do programu BASIC předána jako funkční hodnota.

+ 251. Přepnutí (-inání) stupně/oblouková míra. (Radiány nebo stupně = RADFLG nebo DEGFLG)

Je-li tam vložena \emptyset , počítá se v obloukové míře a je-li tam vložena 1 , počítá se v stupních.

656. Poloha kurzoru v textovém okénku (řádek)

(Řádková poloha textového kurzoru = TXTROW)

Obsah této paměťové buňky odpovídá řádce textového okénka, ve které resp. od které bude psán nebo čten příští znak. PEEK (656) leží vždy mezi \emptyset až 3. \emptyset odpovídá první řádce textového okénka.

694. Inverzní zobrazení znaků (inverzní video znaky = INVFLG)

Nachází-li se v této paměťové buňce 0, pak po stlačení klávesy klávesnice vytvoří se ATASCII kód normálně zobrazeného znaku. Je-li tam

však uložena jiná hodnota než 0, utvoří se kód ATASCII inverzně zobrazeného znaku.

702. Přepnutí malých/velkých písmen nebo ovládacích znaků (posun/zámek ovládání = SHFLOK)

Obsahy, mající účelný smysl těchto paměť. buněk jsou: 0 (normální malé/velké písmo), 64 (velké písmo) nebo 128 (ovládací znaky).

741., 742 Horní konec volné paměti (nejvyšší adresa volné paměti = MEMTOP)

Nejvyšší adresa volné paměti může být vypočítána pomocí PEEK (741) + 256 PEEK (742) - 1. Obsah těchto paměťových buněk se dá změnit stlačením - li se klávesa SYSTEM RESET, nebo se otevře V/V kanál pro obrazovku.

743., 744 Dolní konec volné paměti (nejnižší adresa volné paměti = MEMLO)

Tyto paměťové buňky obsahují nejnižší adresu volné paměti. Obsah těchto paměť. buněk lze změnit stlačením klávesy SYSTEM RESET.

752. Potlačení kurzoru (Inhibice kurzoru = CRSINH)

Je-li v této paměťové buňce uložena 1 je kurzor viditelný. Každá jiná hodnota než 0 činí kurzor neviditelným.

755. Určení zobrazení kurzoru a znaků (Ovládání kurzoru a písmen = CHACT)

V této paměťové buňce je normálně uložena 2. Změnou obsahu této paměťové buňky lze učinit kurzor neprůhledným nebo neviditelným nebo invertovat zobrazení znaků.

756. Volba sady znaků (Charakter adresové základny = CHBAS)

Pomocí této paměťové buňky volíme, která sada znaků má být použita v grafických způsobech provozu 1 a 2. Je-li tam uložena hodnota 224, je použito té sady znaků, která obsahuje velká písmena a číslice. Je-li však v této paměťové buňce uloženo 226, je použito té sady znaků, která obsahuje malá písmena a grafické znaky.

764. Naposledy zmáčkнутá klávesa. (Charakter klávesnice = CH)

V této paměťové buňce je uloženo kódové číslo té klávesy, která byla stlačena jako poslední. Nebyla-li stlačena žádná klávesa, je tam uloženo 255.

765. Volba barvy znaků pro pokyn XIO (F III DATA = FILDAT)

Obsah této paměťové buňky určuje barvu, kterou má být vyplněna nějaká plocha na stínítku pomocí pokynu XIO.

766. Indikování ovládacích znaků (Písmena ovládání na displeji = DSPFLG)

Je-li v této paměťové buňce uložena 0, pak, když je na obrazovce přenášen znak s kódovým číslem ATASCII 27 ÷ 31, 123 ÷ 127, 187 ÷ 191 nebo 251 ÷ 255, je provedena odpovídající ovládací funkce.

Je-li v ní uložena jiná hodnota než \emptyset , pak místo toho dojde k zobrazení znaku na obrazovce.

767. Výstup na obrazovku blokován (start/stop na obrazovku = SSFLAG)

Je-li v této paměťové bunce uloženo 255, je výstup na obrazovku blokován. Je-li v ní uložena 0, výstup na obrazovku není ovlivněn. Při stlačení CTRL - 1 mění se obsah této paměťové bunky.

53279. Ovládání funkčních kláves (Post přepínače CONSOLE = CONSOL)

Pomocí pokynu PEEK (53279) lze zjistit, zda došlo k stlačení jedné ze tří funkčních kláves (OPTION, SELECT a START) a pakliže ano, pak které. Před pokynem PEEK (53279) měl by vždy předcházet pokyn POKE (53279,8). Jen tehdy je úplně jisté, že čtená hodnota je správná.

Obsah bunky 53279 (decimálně)	Stlačená funkční klávesa
0	OPTION, SELECT a START
1	OPTION a SELECT
2	OPTION a START
3	OPTION
4	SELECT a START
5	SELECT
6	START
7	žádná

DODATEK

Rozdělení paměti systému XL

Následující tabulka znázorňuje, jak procesor 6502 rozděluje rozsah adres. Maximální rozsah adres, který může 8-bitový procesor 6502 obsáhnout, leží v rozsahu \$ 0000 až \$ FFFF. Tento rozsah adres se hardwarovým zapojením člení následovně:

ROZDĚLENÍ PAMĚTI

HEX.ADR	OBSAZENA ČÍM	Pozn.
FFFF - D800	OS - ROM nebo RAM, když je ROM odpojena	①
D7FF - D000	Přístupem na tyto paměti PAGE se provádí aktivní výběr čipů u periférií	

Prostorové rozdělení V/V (paměťově mapované)

D000 - D0FF GTIA

D200 - D2FF POKEY

D300 - D3FF PIA

D400 - D4FF ANTIC

D500 - D5FF Každá oslovená adresa v tomto rozsahu aktivuje kontrolní zapojení CCNTL jednotky modulu styku (jako u předchozí řady)

D100 - D1FF Rezervováno pro budoucí obsazení
D600 - D6FF
D700 - D7FF
OS (operační systém) - ROM je fyzicky přítomen, nelze jej však oslovit (2)
CFFF - C000 OS - ROM nebo RAM, je-li ROM odpojen
BFFF - A000 RAM nebo modul jednotky styku, když je vedení RD5 zasunutím nabuzeno na + 5V
9FFF - 8000 RAM nebo modul jednotky styku, je-li vedení RD4 zasunutím vybuzeno na + 5 V
7FFF - 5800 RAM
57FF - 5000 RAM, pokud není v módu vlastní kontroly
4FFF - 0000 RAM (2)

Poznámka: ATARI 800 XL používá stejně jako rozšířený ATARI 600 XL přístupného rozsahu paměti 64K RAM. Tento rozsah, který je normálně celý přístupný uživateli, je řízením paměti (memory manager) rozdělen tak, že pevné paměti ROM, zásuvné moduly a periférie mohou obsadit část paměti.

- 1 Přístup na OS - ROM (oper. pam.) může být přerušen vypsáním hodnoty 0 na part. B PIA, bit PBO. Přístup se umožní normálně vypsáním 1 do tohoto bitu. Při změně tohoto bitu nesmí být ovlivněny ostatní bity registru.
- 2 Kód ROM - u pro vlastní kontrolu je v OS - ROM-u (oper. syst.) fyzicky k dispozici na adresách \$ D000 - \$ D7FF. Tento rozsah je však zapotřebí pro přístup na paměťově mapovaná zařízení V/V. Vybudí-li se vlastní zkouška (test), odpojí se RAM v rozsahu míst (buněk) paměti \$ 5000 - \$ 57FF. Řízení paměti (mem. manager) definuje přístup do paměti tak, že mohou být osloveny adresy \$ D000 - \$ 57FF. Používá portu B z PIA, bit PB7 za účelem zjištění, zda v rozsahu \$ 5000 - \$ 57FF má být aktivována RAM nebo ROM. Je-li PB7 pod napětím, je vybuzena RAM, jinak se přejde na OS - ROM. Při změně tohoto bitu by se neměly ovlivnit ostatní bity registru.

Cvičné příklady

Následující příklady (výpisy programů) Vám pomohou prohloubit znalosti, získané v předchozím kursu BASIC. Doporučujeme Vám, programy přesně opsat a pak obměňovat. Berte přitom ohled na uvedené poznámky. Všechny příklady jsou otištěny tak, jak se jeví na stínítku obrazovky. Za každou programovou řádkou je třeba stlačit klávesu RETURN. Zkouška syntaxe (hlášení ERROR) Vám ihned ukáže, zda nedošlo k nedovolenému zadání. V tom případě proveďte příslušnou opravu.

Start programu: Vložením RUN(RETURN)

Přerušeni: Stlačením klávesy BREAK. Opakovaný start pomocí RUN
Nový výpis vložením LIST

I při stlačení klávesy RESET zůstane program v paměti RAM a může být znovu odstartován po přerušeni programu nebo znovu proveden výpis. Pomocí programové tiskárny ATARI nebo disketové stanice ATARI můžete všechny programy pro další cvičení ukládat a dle potřeby znovu zavádět do paměti počítače. Máte-li k dispozici tyto paměti dat, bude uložený program při novém zavedení přepsán. Vložte přitom pro jistotu NEW(RETURN), aby v závislosti na délce programu nebyly současně přibrány do nového programu staré programové řádky a rušily průběh.

Beispiel 1

```

10 REM *** SIRENE 1 ***
20 GRAPHICS 18
30 POSITION 4,6:PRINT #6;"** SIRENE **"
"
40 FOR I=80 TO 180
50 SOUND 0,I,10,10:SETCOLOR 0,INT(I/10),8
60 NEXT I:SETCOLOR 2,0,0
70 FOR I=180 TO 80 STEP -1
80 SOUND 0,I,10,10:SETCOLOR 0,INT(I/10),8
90 NEXT I:SETCOLOR 2,10,10
100 GOTO 40

```

Beispiel 2

```

10 REM *** SIRENE 2 ***
20 GRAPHICS 0:? "LAENGE";:INPUT AA:GRAPHICS 18
30 POSITION 4,6:PRINT #6;"** SIRENE **"
"
40 FOR I=1 TO AA:NEXT I
50 SOUND 0,80,10,10:SETCOLOR 0,8,8
60 SOUND 1,60,10,10:SETCOLOR 0,8,8
70 SETCOLOR 2,0,0
80 FOR I=1 TO AA:NEXT I
90 SOUND 0,120,10,10:SETCOLOR 0,2,8
100 SOUND 1,100,10,10:SETCOLOR 0,2,8
110 SETCOLOR 2,10,10
120 GOTO 40

```

Beispiel 3

```

10 REM *** TUERKLINGEL ***
20 GRAPHICS 18
30 POSITION 1,6:PRINT #6;"** TUERKLINGEL **"
"
40 FOR I=1000 TO 200 STEP -4
50 IF I<=1000 AND I>700 THEN SOUND 0,150,10,INT((I-700)/20)
60 IF I<=800 AND I>500 THEN SOUND 1,110,10,INT((I-500)/20)
70 IF I<=600 AND I>300 THEN SOUND 2,90,10,INT((I-300)/20)
80 IF I<=1000 AND I>800 THEN SETCOLOR 2,4,8
90 IF I<=800 AND I>600 THEN SETCOLOR 2,8,8
100 IF I<=600 AND I>400 THEN SETCOLOR 2,10,8
110 IF I<=400 AND I>200 THEN SETCOLOR 2,0,0
120 SETCOLOR 0,INT(I/35),8
130 NEXT I
140 GOTO 40

```

S I R É N A

Tyto hodnoty (0 - 255) určují stoupající výšku tónu

Tyto hodnoty (0 - 255) určují klesající výšku tónu.

S I R É N A

Před provedením budete vyzváni ke vložení hodnoty 0 - 255

D O M O V N Í Z V O N E K

Tyto hodnoty určují rychlost změny tónů.

Tyto hodnoty určují barvu hvězdičky.

Beispiel 4

```

10 REM *** SPIELEN EINER NOTENFOLGE **
*
20 DIM A(10)
30 GRAPHICS 0:PRINT "BITTE GEBEN SIE Z
EHN ZAHLEN EIN":? :?
40 FOR I=1 TO 10:PRINT "ZAHL ";I;" ";
:INPUT B:A(I)=B:NEXT I
50 GRAPHICS 0:POSITION 15,12:? "ICH SP
IELE"
60 FOR I=1 TO 10:SOUND 0,A(I),10,15
70 FOR J=1 TO 200:NEXT J
80 NEXT I
90 SOUND 0,0,0,0:GOTO 30

```

Beispiel 5

```

10 REM *** MUSIK PER TASTATUR ***
20 GRAPHICS 18
30 POSITION 7,5:? #6;"MUSIK"
40 OPEN #1,4,0,"K:"
50 FOR I=1 TO 10000
60 GET #1,A:A=A-65:A=A*10+5
70 SOUND 1,A,10,15:POKE 708,A:NEXT I
80 CLOSE #1

```

Beispiel 6

```

10 REM *** GRAPHIK-FAECHER ***
20 GRAPHICS 11:WW=0
30 FOR Z=1 TO 2:FOR ZE=0 TO 15
40 COLOR ZE:WW=WW+5
50 PLOT 0,0:DRAWTO 79,WW
60 NEXT ZE
70 NEXT Z:WW=0
80 FOR Z=1 TO 2:FOR ZE=0 TO 15
90 COLOR ZE:WW=WW+5
100 PLOT 79,0:DRAWTO 0,WW
110 NEXT ZE
120 NEXT Z
130 GOTO 170
140 FOR H=0 TO 14:SETCOLOR 4,0,H
150 FOR W=1 TO 100:NEXT W:NEXT H
160 GOTO 140
170 WW=191:FOR Z=1 TO 2:FOR ZE=0 TO 15
180 COLOR ZE:WW=WW-5
190 PLOT 79,191:DRAWTO 0,WW
200 NEXT ZE
210 NEXT Z
220 WW=191:FOR Z=1 TO 2:FOR ZE=0 TO 15
230 COLOR ZE:WW=WW-5
240 PLOT 0,191:DRAWTO 79,WW
250 NEXT ZE
260 NEXT Z
270 FOR H=0 TO 14 STEP 2:SETCOLOR 4,0,
H:FOR SS=1 TO 10:NEXT SS:NEXT H
280 FOR H=14 TO 0 STEP -2:SETCOLOR 4,0
,H:FOR SS=1 TO 10:NEXT SS:NEXT H
290 GOTO 270

```

62

H R A S E K V E N C E T O N U

Zde můžete měnit počet tónů

Před provedením programu budete
vyzváni k uložení deseti hodnot

H U D B A P Ř E S K L Á V E S N I C I

Výška tónu a barva pozadí se mění podle
toho, která klávesa písmen od A do Z byla
stlačena. Při stisku jiné klávesy se
program přeruší.

G R A F I C K Á P A L E T A

Grafický stupeň 10 nebo 11

změny od 1 do - 5 ve dvojici

změny od 1 do - 5 ve dvojici

Beispiel 7

```

10 REM *** ADRESSEN-MASKE ***
20 OPEN #1,4,0,"K:":OPEN #2,12,0,"S:"
30 DIM A$(24)
40 GRAPHICS 0:POKE 710,0
50 POSITION 2,2:? "NAME :-----"
   "
60 POSITION 2,6:? "STRASSE :-----"
   "
70 POSITION 2,10:? "WOHNORT :-----"
   "
80 POSITION 2,14:? "ALTER :__"
90 POSITION 15,14:? "BERUF :-----"
   "
100 POSITION 10,2:? " :";
110 A$(1)=" ":A$(24)=" ":A$(2)=A$
120 B=2
130 FOR I=1 TO 3
140 FOR K=1 TO 24
150 GET #1,A:? #2;CHR$(A);
160 IF A=155 THEN POSITION 11,B:? A$;:
POP :GOTO 190
170 A$(K,K)=CHR$(A)
180 NEXT K
190 A$(1)=" ":A$(24)=" ":A$(2)=A$
200 IF I=1 THEN B=6:POSITION 10,B:? " :";
210 IF I=2 THEN B=10:POSITION 10,B:? " :";
220 NEXT I
230 POSITION 10,14:? " :";
240 A$="":A$=" "
250 FOR K=1 TO 2
260 GET #1,A:? #2;CHR$(A);
270 IF A=155 THEN POSITION 11,14:? A$;:
POP :GOTO 300
280 A$(K,K)=CHR$(A)
290 NEXT K
300 POSITION 21,14:? " :";
310 A$="":A$(1)=" ":A$(13)=" ":A$(2)=A$
320 FOR K=1 TO 13
330 GET #1,A:? #2;CHR$(A);
340 IF A=155 THEN POSITION 22,14:? A$;:
POP :GOTO 370
350 A$(K,K)=CHR$(A)
360 NEXT K
370 POKE 752,1
380 " :? :? :?
400 OPEN #3,8,0,"P:":GOTO 420
410 FOR I=1 TO 40:GET #2,A:? #3;CHR$(A)
:;NEXT I:? #3," ":RETURN
420 POSITION 0,2:GOSUB 410
430 POSITION 0,6:GOSUB 410
440 POSITION 0,10:GOSUB 410
450 POSITION 0,14:GOSUB 410
600 CLOSE #1:CLOSE #2:CLOSE #3
610 POKE 752,0

```

F O R M U L Á Ř A D R E S

Tento program není opatřen soupiskou adres. K realizaci tohoto požadavku by bylo zapotřebí rozsáhlých úprav programu a zkušeností v programování.

V případě, že není k dispozici tiskárna, musí být program ukončen již na řádce 360, kde je třeba provést změnu, např. 360 NEXT K: END.

Výroky OPEN na řádce 20 a 400, jakož i CLOSE v řádce 600, jsou použity pro výměnu dat s periferním zařízením. Podrobnější popis programu by překonal rámec obsahových možností. Proto odkazujeme na doporučenou literaturu.

Beispiel 8

```
10 REM ** HOCHAUFLOESENDE GRAPHIK **
20 GRAPHICS 24:POKE 710,0:COLOR 1
30 POSITION 40,40
40 FOR I=0 TO 1 STEP 0.01
50 ZEI=6.283*I
60 PLOT 160+45*SIN(ZEI),96+45*COS(ZEI)
:DRAWTO 160+90*SIN(ZEI+0.01*6.283),96+
90*COS(ZEI+0.01*6.283)
70 NEXT I
80 FOR I=0 TO 1 STEP 0.01
90 ZEI=6.283*I
100 PLOT 160+90*SIN(ZEI),96+90*COS(ZEI)
):DRAWTO 160+90*SIN(ZEI+0.01*6.283),96
+90*COS(ZEI+0.01*6.283)
110 NEXT I
120 FOR I=0 TO 1 STEP 0.01
130 ZEI=6.283*I
140 PLOT 160,96:DRAWTO 160+90*SIN(ZEI)
,96+90*COS(ZEI)
150 NEXT I
160 GOTO 80
```

VYSOCE JEMNÁ GRAFIKA

Hodnota může být zadána v rozmezí od 0 do 255. Je možné přizpůsobení na barvu obrazovky. Hodnoty mohou stoupat od 0,001 až 0,25. Musí však být stejné uvnitř každé skupiny (řádek 40/60 a 80/100 a 120)

Beispiel 9


```

10 REM ** BALLSPIEL/PING-PONG **
20 GRAPHICS 5:SETCOLOR 2,0,0
30 COLOR 1:PLOT 0,0:DRAWTO 79,0:DRAWTO
 79,39:DRAWTO 0,39:DRAWTO 0,0
40 POKE 752,1:?"SCORE ";P;"          Z
EIT ";S;
50 POKE 559,46
60 POKE 704,188
70 I=PEEK(106)-8
80 POKE 54279,I
90 POKE 53277,3
100 POKE 53256,1
110 PMBASE=I*256
120 FOR I=PMBASE+512 TO PMBASE+640:POK
E I,0:NEXT I
130 POKE PMBASE+512+85,60:POKE PMBASE+
513+85,60
140 FOR I=PMBASE+384 TO PMBASE+512:POK
E I,0:NEXT I
150 POKE PMBASE+384,3:S=999
160 YV=INT(1*RND(0)+2):XV=INT(-4*RND(0
)+4)
170 YM=19:XM=120
180 ST=STICK(0):X=X+4*((ST=7)+2*(X<195
)):X=X-4*((ST=11)+2*(X>45))
190 POKE 53248,X
200 POKE 53278,0
210 POKE PMBASE+384+YM1,0:A=A+1:S=S-1
220 IF S<10 THEN ?"";
230 IF S<100 THEN ?"";
240 POKE 752,1:?" ";S;"";
250 XM=XM+XV:YM1=YM:YM=YM+YV:IF XM<50
OR XM>205 THEN XV=-XV:SOUND 0,20,10,15
:SOUND 0,0,0,0:GOTO 210
260 IF S=0 THEN GOTO 340
270 IF YM<19 OR YM>92 THEN YV=-YV:SOUN
D 0,20,10,15:SOUND 0,0,0,0:GOTO 210
280 POKE PMBASE+384+YM,3:POKE 53252,XM
290 DF=0
300 IF PEEK(53256)<>0 AND A>3 THEN POK
E 752,2:P=P+1:?"SCORE ";P;"          Z
EIT ";:YV=-YV:XV=-XV:DF=1
310 IF DF=1 THEN POKE 53278,0:A=0:GOTO
 330
320 GOTO 180
330 XV=INT(-4*RND(0)+2):GOTO 210
340 POKE 53277,0:POKE 53265,0:POKE 532
61,0
350 GRAPHICS 18
360 POSITION 2,3:?" #6;"punkte : ";P
370 POSITION 2,9:?" #6;"START =neues sp
iel"
380 POSITION 2,11:?" #6;"SELECT=ende"
390 IF PEEK(53279)=6 THEN CLR :GOTO 20
400 IF PEEK(53279)=5 THEN GRAPHICS 0:E
ND
410 GOTO 390

```

H R A P I N G - P O N G

Program v technice Hráč/Raketa.
K pohybu pálek je zapotřebí páky ("knipl").
K přerušení je třeba stlačit nikoli klávesu BREIK ale RESET, neboť jinak by vyrůstal na levém okraji obrazovky viditelný průběžný zelený pás.

Pro znak  stlačena ESC a dále současně CONTROL a CLEAR.

Beispiel 10

```

10 REM *** GRAPHIK-KREISE I ***
20 GRAPHICS 31:COLOR 1
30 POKE 708,91:POKE 709,181:POKE 710,1
31
40 A=20:B=33
50 FOR K=1 TO 5
60 FOR I=0 TO 1 STEP 0.03
70 ZEI=6.283*I
80 PLOT A,B:DRAWTO A+15*SIN(ZEI+0.01*6
.283),B+30*COS(ZEI+0.01*6.283)
90 NEXT I
100 IF K=1 THEN A=50:B=33:COLOR 2
110 IF K=2 THEN A=80:B=33:COLOR 3
120 IF K=3 THEN A=110:B=33:COLOR 1
130 IF K=4 THEN A=140:B=33:COLOR 2
140 NEXT K
150 A=20:B=129:COLOR 3
160 FOR K=1 TO 5
170 FOR I=0 TO 1 STEP 0.03
180 ZEI=6.283*I
190 PLOT A,B:DRAWTO A+15*SIN(ZEI+0.01*
6.283),B+30*COS(ZEI+0.01*6.283)
200 NEXT I
210 IF K=1 THEN A=50:B=129:COLOR 1
220 IF K=2 THEN A=80:B=129:COLOR 2
230 IF K=3 THEN A=110:B=129:COLOR 3
240 IF K=4 THEN A=140:B=129:COLOR 1
250 NEXT K
260 A=35:B=81:COLOR 2
270 FOR K=1 TO 4
280 FOR I=0 TO 1 STEP 0.03
290 ZEI=6.283*I
300 PLOT A,B:DRAWTO A+15*SIN(ZEI+0.01*
6.283),B+30*COS(ZEI+0.01*6.283)
310 NEXT I
320 IF K=1 THEN A=65:B=81:COLOR 3
330 IF K=2 THEN A=95:B=81:COLOR 1
340 IF K=3 THEN A=125:B=81:COLOR 2
350 NEXT K
360 IF A=1 THEN POKE 708,INT(RND(0)*25
5)
370 IF A=2 THEN POKE 709,INT(RND(0)*25
5)
380 IF A=3 THEN POKE 710,INT(RND(0)*25
5)
390 FOR U=1 TO 20:NEXT U
400 A=INT(RND(0)*3+1)
410 GOTO 360

```

G R A F I K A - K R U H Y 1

Zde se dosáhne změna barvy pomocí výroku POKE.

Hodnotami od 1 do 1000 se ovlivňuje rychlost změny barev.

Beispiel 11

```

10 REM *** GRAPHIK-KREISE II ***
20 GRAPHICS 31:COLOR 1
30 SETCOLOR 0,3,10:SETCOLOR 1,8,10:SET
COLOR 2,11,10
40 A=20:B=33
50 FOR K=1 TO 5
60 FOR I=0 TO 1 STEP 0.03
70 ZEI=6.283*I
80 PLOT A,B:DRAWTO A+15*SIN(ZEI+0.01*6
.283),B+30*COS(ZEI+0.01*6.283)
90 NEXT I
100 IF K=1 THEN A=50:B=33:COLOR 2
110 IF K=2 THEN A=80:B=33:COLOR 3
120 IF K=3 THEN A=110:B=33:COLOR 1
130 IF K=4 THEN A=140:B=33:COLOR 2
140 NEXT K
150 A=20:B=129:COLOR 3
160 FOR K=1 TO 5
170 FOR I=0 TO 1 STEP 0.03
180 ZEI=6.283*I
190 PLOT A,B:DRAWTO A+15*SIN(ZEI+0.01*
6.283),B+30*COS(ZEI+0.01*6.283)
200 NEXT I
210 IF K=1 THEN A=50:B=129:COLOR 1
220 IF K=2 THEN A=80:B=129:COLOR 2
230 IF K=3 THEN A=110:B=129:COLOR 3
240 IF K=4 THEN A=140:B=129:COLOR 1
250 NEXT K
260 A=35:B=81:COLOR 2
270 FOR K=1 TO 4
280 FOR I=0 TO 1 STEP 0.03
290 ZEI=6.283*I
300 PLOT A,B:DRAWTO A+15*SIN(ZEI+0.01*
6.283),B+30*COS(ZEI+0.01*6.283)
310 NEXT I
320 IF K=1 THEN A=65:B=81:COLOR 3
330 IF K=2 THEN A=95:B=81:COLOR 1
340 IF K=3 THEN A=125:B=81:COLOR 2
350 NEXT K
360 IF A=1 THEN SETCOLOR 0,3,INT(RND(0
)*15)
370 IF A=2 THEN SETCOLOR 1,3,INT(RND(0
)*15)
380 IF A=3 THEN SETCOLOR 2,3,INT(RND(0
)*15)
390 FOR U=1 TO 20:NEXT U
400 A=INT(RND(0)*3+1)
410 GOTO 360

```

G R A F I K A - K R U H Y 2

Podobně jako u příkladu 10, s tím rozdílem, že změny barvy se dosáhne nikoli pokynem POKE, ale SETCOLOR

Jsou možné libovolné hodnoty od 1 do 3

Jsou možné libovolné hodnoty od 1 do 15

Spiel 12

```

10 REM ** STREICHHOLZ-SPIEL **
20 DIM A$(38):GRAPHICS 7
30 SETCOLOR 0,1,5:SETCOLOR 1,3,1:SETCO
LOR 2,0,6:SETCOLOR 4,0,6
40 A=157:B=10:C=30:D=33:RESTORE
50 X=1:E=A:F=B:G=C:ANDE=21:ANFANG=1:GO
SUB 360
60 X=2:E=A:F=C:G=D:ANDE=21:ANFANG=1:GO
SUB 360
70 FOR I=1 TO 6
80 A$="":READ A$:LANG=LEN(A$)
90 FOR J=1 TO LANG:FOR K=1 TO 50:NEXT
K
100 ? A$(J,J);:NEXT J:?:?
110 NEXT I:N=21
120 ? :? "WIEVIELE STREICHHOELZER ";
130 INPUT STREICH
140 IF STREICH<>INT(STREICH) THEN ? :?
"NUR GANZE ZAHLEN, SIE ....":?:GOTO
120
150 IF STREICH<1 OR STREICH>4 THEN ? :
? "OHNE MICH, SO GEHT ES JA NICHT":?:
GOTO 120
160 N=N-STREICH
170 X=3:E=A:F=B:G=D:ANDE=21-N:ANFANG=1
:GOSUB 360
180 STREICH=5-STREICH
190 FOR I=1 TO 300:NEXT I
200 ? :? "ICH NEHME ";STREICH
210 N=N-STREICH
220 X=3:E=A:F=B:G=D:ANDE=21-N:ANFANG=1
:GOSUB 360
230 IF N>1 THEN 120
240 ? "  *** ICH HABE GEWONNEN ***"
250 ? :? "NOCH EIN SPIEL (J,N) ";
260 A$="":INPUT A$
270 IF A$="J" THEN ? "ü":GOTO 30
280 GRAPHICS 0:END
290 ? :? " HERZLICHEN GLUECKWUNSCH"
300 DATA DAS SPIEL BEGINNT !!!!!!!
310 DATA WIR HABEN 21 STREICHHOELZER
320 DATA WER DAS LETZTE HOLZ NIMMT
330 DATA HAT VERLOREN
340 DATA WÄHLEN SIE ZWISCHEN 1 UND 4
HOELZERN
350 DATA SIE DÜRFEN BEGINNEN
360 COLOR X
370 FOR I=ANFANG TO ANDE
380 PLOT E-7*I,F:DRAWTO E-7*I,G
390 NEXT I
400 RETURN

```

H R A S E Z Á P A L K A M I

```

50 ZAČÁTEK
60
120 KOLIK ZÁPALEK?
140 JENOM CELÁ ČÍSLA, VY...
150 BEZE MNE, TAK TO PŘECE NEJDE
200 BERU
240 ZÍSKAL JSEM
250 JEŠTĚ JEDNU HRU
290 SRDEČNĚ BLAHOPŘEJI
300 HRA ZAČÍNÁ
310 MÁME 21 ZÁPALEK
320 KDO BERE POSLEDNÍ ZÁPAJKU
350 MUŽETE ZAČÍT, VOLTE MEZI 1 AŽ 4
ZÁPAJKAMI

```

Beispiel 13

```

10 REM *** ATARI MISCHFARBEN ***
20 GRAPHICS 11
30 FOR X=0 TO 79 STEP 5
40 FOR Y=0 TO 191 STEP 12
50 FOR I=0 TO 9 STEP 2
60 COLOR X/5
70 PLOT X,Y+I:DRAWTO X+3,Y+I
80 COLOR Y/12
90 PLOT X,Y+I+1:DRAWTO X+3,Y+I+1
100 NEXT I
110 NEXT Y:NEXT X
120 GOTO 120

```

S M Í Š E N Ě B A R V Y

Beispiel 14

```

10 REM ** ATARI BRINGT 256 FARBEN **
20 GRAPHICS 9
30 FOR A=0 TO 79:COLOR INT(A/5)
40 PLOT A,9:DRAWTO A,191:NEXT A
50 FOR A=1536 TO 1562:READ R:POKE A,B:
NEXT A:D=PEEK(560)+256*PEEK(561)
60 FOR A=0 TO 14:READ R:POKE D+B,143:N
EXT A
70 POKE 1616,0:POKE 512,0:POKE 513,6:P
OKE 54286,192
80 GOTO 80
90 DATA 72,173,80,6,24,105,16,141,80,6
,141,10,212,141,26,208,201,240,208,5,1
69,0,141,80,6,104,64
100 DATA 17,29,41,53,65,77,89,104,116,
128,140,152,164,176,188
110 GOTO 110

```

A T A R I P R E Z E N T U J E

2 5 6 B A R E V

Zde předvádění ATARI paletu 256 barev

Beispiel 15

```

10 REM ** ATARI REGENBOGEN-FARBEN **
20 DIM C$(24)
30 SETCOLOR 2,0,0:POKE 752,1:?" "
40 FOR I=1 TO 24
50 READ D
60 C$(I,1)=CHR$(D)
70 NEXT I
80 D=USR(ADR(C$))
90 END
100 DATA 162,0,173,11,212,201,32,208,2
49,141
110 DATA 10,212,142,24,208,232,232,208
,246,142
120 DATA 24,208,240,232

```

A T A R I V B A R V Á C H D U H Y

Program může být přerušen pouze klávesou
RESET

SEZNAM VÝROKŮ A FUNKCÍ POUŽÍVANÝCH V ATARI BASIC

Vyhrazený výrok (slovo)	Zkráceně	Funkce
ABS		Vyčísluje absolutní hodnotu čísla
ADR		Vyčísluje adresu, od které je v paměti proměnná
AND		Logický vztah AND mezi dvěmi hodnotami, nikoliv bitově, nýbrž výsledek je 0, je-li jedna z hodnot rovna 0.
ASC		Vyčísluje ATASCII kód znaku.
ATN		Vyčísluje arkustangens čísla
BYE	B.	Přechod z BASIC-u do módu testování sama sebe (vlastní zkoušky)
CLOAD	CLOA.	Převádí data nebo programy záznamového zařízení programů do paměti RAM.
CHR §		Převádí hodnotu, příslušející ATASCII (D - 255) na znak.
CLOG		Vyčísluje desítkový logaritmus čísla.
CLOSE	CL.	Uzavírá V/V kanál.
CLR		Maže všechny proměnné.
COLOR	C.	Určuje barevný registr v grafickém způsobu provozu.
COM		Vyhrazuje místo v paměti.
CONT	CON.	Způsobuje, že se pokračuje v provádění programu v následující řádce použití klávesy BREAK nebo zadání STOP.
COS		Vyčísluje cosinus úhlu.
CSAVE	CS.	Převádí(ukládá) na kazetu program, který se nachází v RAM
DATA	D.	Část kombinace READ/DATA. Slouží k ukládání do paměti hodnot, které mohou být výrokem READ přiděleny proměnným
DEG	DE.	Přepíná z obloukové míry na stupně.

DIM	DI.	Vyhrazuje místo v paměti pro pole nebo alfanumerické proměnné.
DOS	DO.	Nabíjení DOS - menu
DRAWTO	DR.	Kreslí linii od posledního nakresleného bodu k udanému konečnému bodu.
END		Zakončuje probíhající program.
ENTER	E.	Slouží k nabití programu BASIC, který byl uložen pomocí LIST z kazety nebo diskety.
EXP		Označuje hodnotu exponenciální funkce.
FOR, TO, NEXT	F.	Série výroků, které mají být prováděny tak dlouho, až průběžná (proměnlivá) proměnná dosáhne určité hodnoty.
FRE (0)		Vyčísluje zbývající volné místo v paměti v Bytech po odečtení OS, DOS, seznamu (Display-list) displeje, paměti obrazovky a BASIC s daty.
GET	GE.	Nabíjí za použití jednoho V/V kanálu jediný Byte z osloveného přístroje.
GOSUB	GOS.	Slouží k vyvolání podprogramu.
GOTO	G.	Provádění programu bude pokračovat od uvedeného čísla řádku.
GRAPHICS	GR.	Aktivuje jeden z grafických způsobů provozu.
IF ...THEN		Je-li splněna určená podmínka, budou provedeny výroky uveden za THEN
INPUT	I.	Pomocí INPUT mohou být přidělovány (přiřazovány) hodnoty proměnným.
INT		Zaokrouhluje číslo na celou hodnotu.
LIST	L.	Zobrazuje program, který se nachází v RAM. Pokud je uložen pomocí LIST na kazetu nebo disketu, může být znovu nabit jen pomocí ENTER.
LEN		Vyčísluje délku alfanumerické proměnné.
LET	LE.	Přiřazuje proměnné hodnotu.
LOAD	LO.	Přenáší předem uložený program do počítače.
LOCATE	LOC.	Nabíjí kód znaku nebo grafického bodu, který se nachází v udané poloze.

LOG		Vyčísluje přirozený logaritmus čísla
NEW		Maže program, který se právě nachází v RAM a všechny proměnné
LPRINT	LP.	Slouží k přenosu znaku do tiskárny.
NOT		Výsledkem je hodnota 1, je-li výraz nepravdivý je-li však pravdivý, je výsledkem hodnota 0.
NOTE	NO.	Stanoví, na kterém místě souboru (stránky) dat na disketě se nachází ukazatel (pointer) tohoto souboru dat.
ON		Používá se ve spojení s GOTO nebo GOSUB k provádění programu od určeného řádku nebo od podprogramu.
OPEN	O.	Přiděluje určitému V/V kanálu určitý přístroj
OR		Logické porovnání dvou výrazů. Pokud je jen jeden pravdivý, je výsledkem hodnota 0, jsou-li oba nepravdivé je výsledkem 1.
PADDLE		Určuje okamžitou polohu udaného otočného regulátoru (ovládače) hodnotou mezi 1 a 228.
PEEK		Slouží ke čtení jednoho paměťového místa (buněk) v RAM nebo ROM.
PLOT	PL.	Nakreslí na obrazovce v udaném místě bod.
POINT	P.	Nasadí (nastaví) ukazatele (pointer) souboru dat (stránky) diskety do určité polohy.
POKE	POK.	Slouží k ukládání určité hodnoty do určené paměťové buněk. První hodnota nesmí se týkat paměti ROM, druhá musí ležet mezi 0 a 255.
POP		Maže v zásobníkové paměti adresu návratu posledního provedeného výroku FOR - GOSUB - nebo ON GOSUB-.
POSITION	POS.	Nasadí kurzor do určitého místa obrazovky.
PRINT	PR. nebo ?	Slouží k tisku či zápisu na obrazovku.
PTRIG		Slouží k zjištění, zda je stlačena spoušť (stlačen knoflík) otočného regulátoru.
PUT	PU.	Slouží k výdeji čísla přes udaný V/V kanál.
RAD		Přepíná ze stupňů na obloukovou míru.
READ	REA.	Přiděluje hodnoty udané u výroku DATA, udané proměnné.

REM	R.	Slouží k doplnění vysvětlivek do programu.
RESTORE	RES.	Slouží k návratu ukazatele (pointer), příslušného (příslušejícího) k výrokům DATA.
RETURN	RET.	Slouží ke zpětnému skoku k výroku, který následuje bezprostředně po naposled provedeném GOSUB nebo ON GOSUB.
RND (ϕ)		Vyčísluje náhodné číslo.
RUN		Slouží ke startu programu, který se nachází v RAM, případně nabíjí z pojmenovaného přístroje a ihned startuje.
SAVE	S.	Slouží k přenosu programu BASIC, který se nachází v RAM do určeného ukládacího zařízení. Může být znovu nabit pomocí LOAD nebo RUN.
SETCOLOR	SE.	Určuje barvu a jas při zobrazení určitého (určeného) barevného registru.
SGN		Slouží ke zjištění, zda je číslo kladné, záporné nebo rovno nule.
SIN		Vyčísluje sinus úhlu.
SOUND	SO.	Zapíná nebo vypíná tónový kanál a určuje výšku tónu, čistotu tónu a hlasitost.
SQR		Vyčísluje druhou odmocninu čísla.
STATUS	ST.	Slouží ke zjištění, s jakým výsledkem byla uzavřena naposledy V/V operace ve vztahu k uvedenému (určenému) kanálu.
STEP		Používá se ve spojení se smyčkami FOR/NEXT, určuje zvyšování po skocích běžící proměnné (proměnných).
STICK		Stanoví okamžitou polohu udané řídicí páky.
STRIG		Slouží ke zjištění, zda spoušťový knoflík řídicí páky byl stlačen.
STOP	STO.	Slouží k přerušení programu BASIC.
STR \$		Převádí numerickou hodnotu na alfanumerickou.
TRAP	T.	Slouží k záchytu a zpracování hlášení chyby.
USR		Slouží k vyvolání strojního programu.
VAL		Převádí alfanumerickou hodnotu na numerickou.
XIO	X.	Slouží k práci s V/V kanály.

```

1000 REM HRA BYSTROST
1010 ? ")" : POSITION 5,5 : ? "PROVERIM TVOU BYSTROST A PAMET."
1020 ? "#6;" : "PO ZADANI OBTIZNOSTI VYTISKNU SADU"
1030 ? "#6;" : "CISEL.PO VTERINE JI UTAJIM A TVYM UKOLEM";
1040 ? "#6;" : "BUDE SPRAVNY ZAPIS CELE SADY." : ? : ? " START ?"
1050 OPEN #1,6;0,"K:" : DIM A(10)
1060 IF PEEK(53279)<>6 THEN 1050
1070 P=0
1080 GRAPHICS 18
1090 ? "#6;" : "OBTIZNOST(4-9):" : GET #1,C
1100 C=C-48 : ? "#6;" : IF C<4 OR C>9 THEN 1080
1105 FOR SER=1 TO 10
1115 POSITION 1,5 : ? "#6;" : SER ; " POZOR"
1120 FOR W=0 TO 600 : NEXT W : POSITION 6,5 : ? "#6;" "
1130 D=INT((20-C)/2) : POSITION D,5
1140 FOR DES=1 TO C
1150 A(DES)=INT(RND(0)*10) : ? "#6;" : A(DES) ;
1160 NEXT DES
1170 FOR W=0 TO 650 : NEXT W
1180 POSITION D,5 : ? "#6;" " : POSITION D,5
1190 FOR DES=1 TO C
1200 GET #1,CC : CC=CC-48 : IF CC<0 OR CC>9 THEN 1200
1205 IF CC=A(DES) THEN 1240
1210 SOUND 0,25,10,10 : FOR W=0 TO 100 : NEXT W : SOUND 0,0,0,0
1220 P=P+1 : IF P>2 THEN 1360
1230 GOTO 1270
1240 ? "#6;" : CC ;
1250 NEXT DES
1260 FOR W=0 TO 100 : NEXT W : POSITION D,5 : ? "#6;" "
1270 NEXT SER
1280 GRAPHICS 18 : POSITION 0,5
1290 IF C=4 THEN ? "#6;" : "SLABE ZVYS OBTIZNOST"
1300 IF C=5 THEN ? "#6;" : "DOBRE, JESTE SE UC"
1310 IF C=6 THEN ? "#6;" : "NECO Z TEBE BUDE"
1320 IF C=7 THEN ? "#6;" : "MAS SKVELOU PAMET"
1330 IF C=8 THEN ? "#6;" : "JSI GENIUS NA CISLA"
1340 IF C=9 THEN ? "#6;" : "PODVADIS MNE"
1350 ? "#6;" : "CHES HRAT ZNOVU?"
1353 ? "#6;" : "POKUD ANO, START."
1355 GOTO 1060
1360 POSITION 0,5 : ? "#6;" : "NESTACIS " : GOTO 1350
2000 REM 2000 BYTE
11230 GOTO 1270

```

```
10 POKE 106,PEEK(106)-8:GOSUB 400:GOSUB 1500:POKE 708,13
20 SCR=PEEK(88)+PEEK(89)*256:DIM AM$(10)
30 POSITION 2,2: ? #6;")ENTER:" :POSITION 6,4: ? #6;"1 FOR NOVICE"
40 POSITION 6,6: ? #6;"2 FOR PRO":POSITION 6,8: ? #6;"3 FOR EXPERT"
50 POSITION 6,10: ? #6;"4 FOR PERFECT"
60 J=PEEK(764):IF J<>24 AND J<>26 AND J<>29 AND J<>30 AND J<>31 THEN 60
70 IF J=29 THEN POKE 764,255:END
75 IF A=35 THEN 100
80 POSITION 1,10: ? #6;") USE C AND M KEYS TO MOVE LEFT AND RIGHT RESPECTIVE
90 FOR I=1 TO 2000:NEXT I
100 ? #6;")":N=SCR+245
110 I=0.1:IF J=30 OR J=24 THEN I=0.2
120 AM$=""
130 FOR Q=1 TO 4
140 FOR C=7 TO 0 STEP -1
150 FOR R=1 TO 7.2 STEP 1
160 Y=COS(R)
165 F=F+1:S=RND(0)
166 IF F>23 THEN F=23:A=USR(1536)
170 POSITION 4*Y+6,F: ? #6;"$";AM$(1,C);"$":IF S>0.5 THEN 200
180 IF J=31 OR J=30 THEN 200
190 IF F=23 THEN L=INT(420*RND(1)+20):POKE N+L,69
200 IF F=23 THEN POKE N,131:FOR T=1 TO 20:NEXT T:POKE N,0
210 IF PEEK(N)=4 OR PEEK(N+1)=4 OR PEEK(N-1)=4 OR PEEK(N+20)=4 THEN 275
220 P=PEEK(764):IF P=18 THEN N=N-1:GOTO 240
230 IF P=37 THEN N=N+1
240 POKE 764,255
250 IF PEEK(N)=69 OR PEEK(N+1)=69 OR PEEK(N-1)=69 OR PEEK(N+20)=69 THEN 275
270 NEXT R:NEXT C:NEXT Q
275 FOR M=20 TO 29:SOUND 1,M,8,15:FOR V=1 TO 7:NEXT V
280 POKE N,198:POKE N-1,198:POKE N+1,198
290 POKE N+20,198:POKE N-20,198:POKE N-19,198:POKE N-21,198
300 POKE N+21,198:POKE N+19,198
310 NEXT M:SOUND 1,0,0,0
320 REM
330 F=0:POSITION 0,21
340 ? #6;"ROAD IS";" ";C;" "; "FEET WIDE"
350 ? #6;"YOUR SCORE IS";" ";INT(10000/C)
360 ? #6;"PLAY AGAIN (Y/N)? "
370 A=PEEK(764):IF A<>43 AND A<>35 THEN 370
380 IF A=43 THEN ? #6;")":N=SCR+245:GOTO 140
390 ? #6;")":GOTO 30
400 GRAPHICS 1+16:POSITION 0,12: ? #6;"JUST A MOMENT PLEASE"
405 ST=(PEEK(106)-8)*256
410 FOR K=0 TO 511:POKE ST+K,PEEK(57344+K):NEXT K:POKE 756,ST/256
415 READ Y:IF Y=-1 THEN RETURN
420 FOR X=Y TO Y+7:READ Z:POKE X+ST,Z:NEXT X:GOTO 415
430 DATA 24,153,255,189,60,60,189,255,153
440 DATA 32,255,255,255,60,60,255,255,255
450 DATA 40,24,60,128,255,255,255,255,255
460 DATA 48,154,82,0,27,216,0,74,137
470 DATA 56,255,255,255,255,255,255,255,255,-1
1500 FOR I=1536 TO 1595:READ A:CK=CK+A:POKE I,A:NEXT I
1510 IF CK<>7887 THEN PRINT "Error in DATA--check typing."
1520 RETURN
1536 DATA 24,165,88,133,203,105
1542 DATA 20,133,205,165,89,133
1548 DATA 204,105,0,133,206,162
1554 DATA 23,160,19,177,205,145
1560 DATA 203,136,16,249,24,165
1566 DATA 205,133,203,105,20,133
1572 DATA 205,165,206,133,204,105
1578 DATA 0,133,206,202,208,227
1584 DATA 160,19,169,0,145,203
1590 DATA 106,16,251,104,96,0
```

SEZNAM PROGRAMŮ: 1. Hry

01	MC	T00	1	AZTECK CHALLENGE
02	BA	TD0	1	SCHATZ SUCHE
03	BA	TD0	1	AFFEN JAGD
04	BA	TD0	1	FISCHE FISCHEN
05	BA	TD0	1	SUPPER GARTNER
06	BM	TD0	1	CAMBODIA
07	BA	TD0	1	SPLAT
08	BA	TD0	0	MOUNTAINS SHOOT
09	BM	TD0	1	JUMPER
10	BM	TD0	1	ATLANTIS
11	BA	TD0	1	JESKYNE
12	BA	TD0	1	KOSTKY
13	BA	TD0	1	BATS
14	BA	TD0	1	WERSCHLUNGENE PFADE
15	BA	TD0	1	SUPERLABIRINT
16	BM	TD0	1	THE CASTLE
17	BA	TD0	1	HELICOPTER PILOT
18	BA	TD0	1	DEEP SPACE
19	BA	TD0	1	VYHNANSTVI
20	BA	TD0	2	TRAPPER
21	BA	TD0	1	RALLYE
22	BA	TD0	0	OVECKA V OHNI
23	BA	TD0	0	PISKVORKY
24	BA	TD0	0	HRACI AUTOMAT
25	BA	TD0	1	SKYING
26	BA	TD0	1	MUSHROOM INVADERS
27	BA	TD0	1	SPACE RESCUE
28	BA	TD0	1	POSTMAN
29	BA	TD0	0	CLOVECE NEZLOB SE
30	BM	TD0	1	OLYMPIA
31	BM	TD0	1	DUNGEONS OF XOTHA
32	BA	TD0	1	DIBLICI
33	BA	TD0	0	HADEJ I NEBO 0
34	BA	TD0	0	POSTREH
35	BA	TD0	1	ROADER
36	BA	TD0	0	WURM
37	BA	TD0	1	CERV
38	BA	TD0	1	HOUSENKA
39	BA	TD0	1	PING-PONG
40	BA	TD0	0	PYRAMIDA
41	BA	TD0	1	ROBOTS
42	BA	TD0	1	MINERS
43	BA	TD0	X	HAMURABI
44	XX	TD0	X	SEEKRIEG
45	XX	TD0	X	SURROUND
46	XX	TD0	X	ANGRIFF AUS DEM WELTRAUM
47	XX	TD0	X	STAEDTE - VERTEIDIGUNG
48	XX	TD0	X	OLD MAC DONALD'S FARM
49	XX	TD0	X	MELODIE-COMPOSER

50	XX	TD0	X	HEXAPAWN
51	XX	TD0	X	MOTIE
52	XX	TD0	X	SUPERBARICADE
53	XX	TD0	X	OTHELLO
54	XX	TD0	X	SPEEDELLO
55	XX	TD0	X	DEPTHCHARGE
56	XX	TD0	X	RITTER FUER EINEN TAG
57	BA	TD0	1	VRTULNIK 2
58	BM	TD0	1	PARASUTISTA
59	BM	TD0	1	ESCAPE FROM EARTH
60	XX	TD0	1	GOLDCELL
61	XX	TD0	X	SLADACKA
62	XX	TD0	X	REVERZ
63	XX	TD0	X	SPACE FLEYING
64	BA	TD0	1	COSMIC INVASSION
65	BA	TD0	2	TANKOVA BITVA
66	BA	TD0	2	HONICKA
67	MC	T00	1	FORBIDEN FOREST
68	MC	T00	1	HYPER BALAST
69	MC	T00	1	DIAMOUND
70	MC	T00	1	JET BOOT JACK
71	MC	T00	1	JUMPMAN JUNIOR
72	MC	T00	1	H.E.R.O.
73	MC	T00	1	CAVERNS OF KHAFKA
74	MC	T00	1	UBOOT
75	MC	T00	1	BUGGY
76	MC	T00	1	POLY POSITION
77	MC	T00	1	BLUE MAX
78	BM	T00	2	TOP TENIS
79	BM	TD0	1	NASENJACK
80	BM	T00	1	AGENT 0/8/15
81	MC	T00	1	PITSTOP
82	MC	T00	1	MIG ALLE ACE
83	MC	T00	1	BASE FIGHEER
84	MC	T00	1	SOCCER
85	MC	T00	1	SNOKIE
86	MC	T00	1	OZZYS ORCHARD
87	BM	T00	1	ZAXXON
88	MC	T00	1	QUASIMODO
89	MC	T00	1	PITFAL II
90	MC	T00	1	MEGALAX
91	MC	T00	1	FROGGER
92	MC	T00	1	SANT's CHRISTMAS
93	MC	T00	1	ONE ON ONE-basketbal
94	MC	T00	2	BRUCE LEE-karate
95	MC	T00		CAPTAIN BEBLES
96	MC	T00		CAVERNS OF MARS
97	MC	T00		DRELBS-tajemné bludiste
98	MC	T00		GHOST ENTOUNTERS
99	mc	T00		RAUMSCHIFF-souboj s kosm.tělesy

SEZNAM PROGRAMŮ: 2. Škola

- 01 BA TD0 0 ZEMĚPIS:EVROPA <grafický>
- 02 BA TD0 0 ZEMĚPIS:SVET<textový>
- 03 BA TD0 0 MATEMATIKA:POČTY PRO ZŠ <přezkoušení s vyhodnocením výsledků>
- 04 BA TD0 1 MATEMATIKA:MATEMATIKA HROU<missile mathematic>
- 05 BA TD0 0 MATEMATIKA:RACIONÁLNÍ FUNKCE <výpočty kořenů rovnic>
- 06 BA TD0 0 JAZYKY:SLOVNÍČEK <univerzální slovníček,možnost přezkoušení>
- 06 BA TD0 0 OBECNÝ TESTER
- 07 BA TD0 0 STATISTIKA:PŘEHLED O VÝSLEDKÁCH ZKOUŠEK NA SŠ <pro okresní řídící orgány >
- 08 BA TD0 0 STATISTIKA:PŘEHLED O TŘÍDĚ <pro třídní učitele>

POZNÁMKA

Seznam programů je aktualizován ve Zpravodaji AK. Rozšiřování programů mezi členy klubu je zajištěno bezplatně softwarovou službou AK. Programy, které byly získány péčí AK, není dovoleno dále volně rozšiřovat. Veškeré informace, na něž se vztahuje zákon o autorských právech, jsou určeny výhradně pro vnitřní potřebu ATARI klubu!!! Porušování tohoto upozornění je trestné z hlediska mezinárodního práva! Programy, které nejsou uvedeny v seznamu, můžete nabídnout manažérovi softwareservisu na adresu: Lubomír KONTRA, Ondřejova 24/8, PŘIEVIDZA, 971 01

LEGENDA

MC- strojové programy, BA- Basic programy, BM- kombinované programy,
 TDC- kazeta, disketa, kartridž, Ø- není třeba JOY STICK, 1- min. jeden
 JOY STICK, XX- programy nebyly překontrolovány

Častější dotazy z činnosti ATARI klubu:

Dotaz: Některé programy na disketách pro ATARI 800 nejdou aplikovat na novém počítači XL.

Odpověď: Pomůže překladačová disketa ATARI, pomocí které lze starší provozní systém 400/800 převést na počítač ATARI 600XL/800XL.

Dotaz: Jak lze zobrazit ä, ö, ü?

Odpověď: Jsou obsaženy v interní sadě znaků. Zapnutí pomocí POKE 756,204, vypnutí pomocí POKE 756,224, zobrazení pomocí stlačení klávesy CTRL současně s další klávesou. Pro tisk tiskárnou ATARI zadáme: POKE 756,204 (RETURN) LP.
 "ESC ESC CTRL W...

Dotaz: Lze DATA SET (magnetofon) ATARI řídit z BASIC?

Odpověď: Pohon kazety zapínat pomocí POKE 54018,52 a vypínat POKE 54018,60

Dotaz: Jak je možné v programu využít kláves pro zvláštní funkce?

Odpověď: Malý program pro vysvětlenou: 10 A = PEEK (53279)
 20 IF A=N THEN GOTO 40
 30 GOTO 10
 40 PRINTA:END

N=3 při klávese OPTION

N=6 při klávese START

N=5 -"- SELECT

N=7 -"- HELP (stav se nemění, návrat N=0)

Dotaz: Jak se vytisknou vlastní programy BASIC?

Odpověď: Při tiskárně připravené k provozu jednoduše zadáme LIST "P:" a stlačíme RETURN. Jednotlivé řádky pomocí LIST "P:",10,20,atd.

Dotaz: Jak lze ukládat a zavádět programy BASIC?

Odpověď: Jsou různé možnosti. Progr. tiskárna: Ukládání: LOAD"C:" CLOAD ENTER"C:"

Zavádění: SAVE"C:" CSAVE LIST"C:"

Disketová stanice: Ukládání: SAVE"D:Jméno LIST"D:Jméno
 LOAD"D:Název ENTER"D:Název
 Název stránky

Zavádění: Zavedení a start může být sdruženo RUN"C:" nebo RUN"D:Název stránky

Dotaz: Lze napojit disketovou stanici i na ATARI 600 XL?

Odpověď: Zásadně je to možné. Provozní systém disket DOS zabírá ale místo v paměti, takže v 16kB RAM nezbyde dostatek místa pro vlastní program. Doporučuje se rozšíření paměti ATARI 1064.

C. ATASCII Zeichensatz

Decimal Kode	Hexadezimal Kode	ATASCII Zeichen	Tastenschlag	Europäisches Zeichen
0	0	☐	Control	à
1	1	☐	Control A	ú
2	2	☐	Control B	ñ
3	3	☐	Control C	é
4	4	☐	Control D	ç
5	5	☐	Control E	ó
6	6	☐	Control F	ó
7	7	☐	Control G	í
8	8	☐	Control H	£
9	9	☐	Control I	í
10	A	☐	Control J	ü
11	B	☐	Control K	ä
12	C	☐	Control L	ö
13	D	☐	Control M	ú
14	E	☐	Control N	ó
15	F	☐	Control O	ó
16	10	☐	Control P	Ü
17	11	☐	Control Q	ä

Anmerkungen:

1. ATASCII bedeutet ATAZI ASCII. Buchstaben und Zahlen entsprechen dem ASCII-Zeichensatz. Einige der Spezialzeichen haben jedoch eine andere Funktion.
2. Wenn nicht anders angegeben, stellen die Werte von 128 bis 255 die inversen Zeichen der Werte 1 bis 127 dar.
3. Ändern Sie 32 zum Wert der Großbuchstaben um den entsprechenden Kleinbuchstaben zu erhalten.
4. Um den ATASCII-Wert zu erhalten, geben Sie im Direkten Modus PRINT ASC (" ") ein. Setzen Sie zwischen die Anführungszeichen den Buchstaben oder das Graphiksymbol.
5. Die invertiert dargestellten Zeichen werden weiß auf schwarzem Grund, inverse Zeichen dagegen schwarz auf weißem Grund dargestellt.

Decimal Kode	Hexadezimal Kode	ATASCII Zeichen	Tastenschlag	Europäisches Zeichen
18	12	☐	Control R	ú
19	13	☐	Control S	í
20	14	☐	Control T	é
21	15	☐	Control U	é
22	16	☐	Control V	ñ
23	17	☐	Control W	é
24	18	☐	Control X	ä
25	19	☐	Control Y	ä
26	1A	☐	Control Z	À
27	1B	☐	Esc Esc	
28	1C	☐	Esc Control -	
29	1D	☐	Esc Control =	
30	1E	☐	Esc Control +	
31	1F	☐	Esc Control *	
32	20	☐	Space bar	
33	21	☐	Shift 1	
34	22	☐	Shift 2	
35	23	☐	Shift 3	
36	24	☐	Shift 4	
37	25	☐	Shift 5	
38	26	☐	Shift 6	
39	27	☐	Shift 7	
40	28	☐	Shift 9	
41	29	☐	Shift 0	
42	2A	☐	*	
43	2B	☐	+	
44	2C	☐	.	
45	2D	☐	.	
46	2E	☐	.	

Dezimal Kode	Hexadezimal Kode	ATASCII Zeichen	Tastenschlag	Europäisches Zeichen
76	4C	L	L	
77	4D	M	M	
78	4E	N	N	
79	4F	O	O	
80	50	P	P	
81	51	Q	Q	
82	52	R	R	
83	53	S	S	
84	54	T	T	
85	55	U	U	
86	56	V	V	
87	57	W	W	
88	58	X	X	
89	59	Y	Y	
90	5A	Z	Z	
91	5B	Shift	Shift	
92	5C	Shift	Shift	
93	5D	Shift	Shift	
94	5E	Shift	Shift	
95	5F	Shift	Shift	
96	60	Control	Control	
97	61	a	a	
98	62	b	b	
99	63	c	c	
100	64	d	d	
101	65	e	e	
102	66	f	f	
103	67	g	g	
104	68	h	h	
105	69	i	i	

Dezimal Kode	Hexadezimal Kode	ATASCII Zeichen	Tastenschlag	Europäisches Zeichen
47	2F	/	/	
48	30	0	0	
49	31	1	1	
50	32	2	2	
51	33	3	3	
52	34	4	4	
53	35	5	5	
54	36	6	6	
55	37	7	7	
56	38	8	8	
57	39	9	9	
58	3A	Shift	Shift	
59	3B	:	:	
60	3C	<	<	
61	3D	=	=	
62	3E	>	>	
63	3F	Shift	Shift	
64	40	Shift	Shift	
65	41	A	A	
66	42	B	B	
67	43	C	C	
68	44	D	D	
69	45	E	E	
70	46	F	F	
71	47	G	G	
72	48	H	H	
73	49	I	I	
74	4A	J	J	
75	4B	K	K	

Decimal Kode	Hexadezimal Kode	ATASCII Zeichen	Tastenschlag	Europäisches Zeichen
106	6A	J	j	
107	6B	K	k	
108	6C	L	l	
109	6D	M	m	
110	6E	N	n	
111	6F	O	o	
112	70	P	p	
113	71	Q	q	
114	72	R	r	
115	73	S	s	
116	74	T	t	
117	75	U	u	
118	76	V	v	
119	77	W	w	
120	78	X	x	
121	79	Y	y	
122	7A	Z	z	
123	7B	↵	Control	
124	7C		Shift =	
125	7D	⌞	Lsc Control < or Lsc Shift <	
126	7E	⌟	Esc Delete Bk Sp	
127	7F	⏪	Esc Tab	
128	80	⏩	Control .	
129	81	⏮	Control A	
130	82	⏭	Control B	
131	83	⏯	Control C	
132	84	⏰	Control D	
133	85	⏱	Control E	
134	86	⏲	Control F	

Decimal Kode	Hexadezimal Kode	ATASCII Zeichen	Tastenschlag	Europäisches Zeichen
135	87	⏳	Control G	
136	88	⏴	Control H	
137	89	⏵	Control I	
138	8A	⏶	Control J	
139	8B	⏷	Control K	
140	8C	⏸	Control L	
141	8D	⏹	Control M	
142	8E	⏺	Control N	
143	8F	⏻	Control O	
144	90	⏼	Control P	
145	91	⏽	Control Q	
146	92	⏾	Control R	
147	93	⏿	Control S	
148	94	␣	Control T	
149	95	␤	Control U	
150	96	␣	Control V	
151	97	␤	Control W	
152	98	␣	Control X	
153	99	␤	Control Y	
154	9A	␣	Control Z	
155	9B	␤	Return	
156	9C	⏪	Esc Shift Delete Bk Sp	
157	9D	⏩	Esc Shift >	
158	9E	⏮	Esc Control Tab	
159	9F	⏭	Esc Shift Tab	
160	AC	␣	Space bar	
161	A1	⏴	Shift 1	
162	A2	⏵	Shift 2	
163	A3	⏶	Shift 3	

Dezimal Kode	Hexadezimal Kode	ATASCII Zeichen	Tastenanschlag	Europäisches Zeichen
194	C2	B	B	
195	C3	C	C	
196	C4	D	D	
197	C5	E	E	
198	C6	F	F	
199	C7	G	G	
200	C8	H	H	
201	C9	I	I	
202	CA	J	J	
203	CB	K	K	
204	CC	L	L	
205	CD	M	M	
206	CE	N	N	
207	CF	O	O	
208	D0	P	P	
209	D1	Q	Q	
210	D2	R	R	
211	D3	S	S	
212	D4	T	T	
213	D5	U	U	
214	D6	V	V	
215	D7	W	W	
216	D8	X	X	
217	D9	Y	Y	
218	DA	Z	Z	
219	DB	[Shift	
220	DC	\	Shift +	
221	DD]	Shift	
222	DE	^	Shift *	
223	DF	_	Shift -	

Dezimal Kode	Hexadezimal Kode	ATASCII Zeichen	Tastenanschlag	Europäisches Zeichen
164	A4	↓	Shift 4	
165	A5	↘	Shift 5	
166	A6	↙	Shift 6	
167	A7	↖	Shift 7	
168	A8	(Shift 9	
169	A9)	Shift 0	
170	AA	*	.	
171	AB	+	+	
172	AC	,	,	
173	AD	-	-	
174	AE	.	.	
175	AF	/	/	
176	B0	0	0	
177	B1	1	1	
178	B2	2	2	
179	B3	3	3	
180	B4	4	4	
181	B5	5	5	
182	B6	6	6	
183	B7	7	7	
184	B8	8	8	
185	B9	9	9	
186	BA	:	Shift :	
187	BB	;	;	
188	BC	<	<	
189	BD	=	=	
190	BE	>	>	
191	BF	~	Shift /	
192	C0	␣	Shift 8	
193	C1	␣	A	

00000001

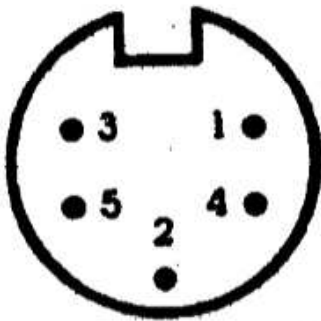
Dezimal Kode	Hexadezimal Kode	ATASCII Zeichen	Tastenanschlag	Europäisches Zeichen
254	FE	◀	Esc Control Delete Bk Sp	
255	FF	▶	Esc Control >	

Dezimal Kode	Hexadezimal Kode	ATASCII Zeichen	Tastenanschlag	Europäisches Zeichen
224	E0	◊	Control	
225	E1	␣	a	
226	E2	␣	b	
227	E3	␣	c	
228	E4	␣	d	
229	E5	␣	e	
230	E6	␣	f	
231	E7	␣	g	
232	E8	␣	h	
233	E9	␣	i	
234	EA	␣	j	
235	EB	␣	k	
236	EC	␣	l	
237	ED	␣	m	
238	EE	␣	n	
239	EF	␣	o	
240	F0	␣	p	
241	F1	␣	q	
242	F2	␣	r	
243	F3	␣	s	
244	F4	␣	t	
245	F5	␣	u	
246	F6	␣	v	
247	F7	␣	w	
248	F8	␣	x	
249	F9	␣	y	
250	FA	␣	z	
251	FB	␣	Control	
252	FC	␣	Shift	
253	FD	␣	Esc Control 2	

PIN-Belegung am Computer (von außen betrachtet)

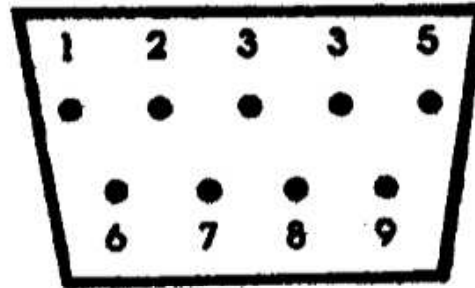
Monitor-Buchse

Mehrfarbmonitor über 2 und 4
Einfarbmonitor über 2 und 1



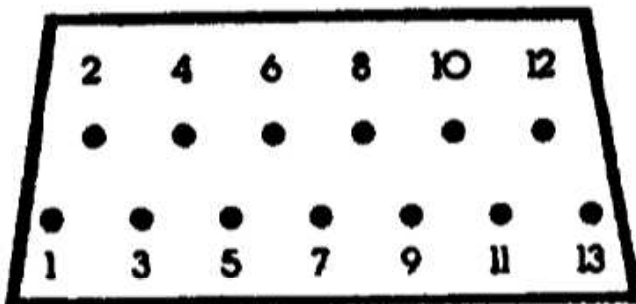
- 1 BAS (bei 600 XL nicht belegt)
- 2 GROUND (Masse)
- 3 Audio Output/NF
- 4 Composite Video (FBAS)
- 5 Composite Chroma (Color + Burst)

Buchse für Steuerungen Port 1 und Port 2



- 1. (Joystick) Forward Input
- 2. (Joystick) Back Input
- 3. (Joystick) Left Input
- 4. (Joystick) Right Input
- 5. B Potentiometer Input
- 6. Trigger Input
- 7. +5 volts
- 8. Ground
- 9. A Potentiometer Input

Serielle I/O Buchse



- 1. Clock Input
- 2. Clock Output
- 3. Data Input
- 4. Ground
- 5. Data Output
- 6. Ground - *propojeni' mead konuktory s' nach periferii*
- 7. Command
- 8. Motor Control
- 9. Proceed
- 10. +5/Ready
- 11. Audio Input
- 12. (nicht belegt)
- 13. Interrupt

YUGOSLAVIEN SOFTWARE ATARI CLUB

A:\ATARI_XL.DOC

Lacmanovic Dejan
 Sindjeliceva 31/A
 23000 Zrenjanin
 Tel. 023 66-879

A T A R I SOFT-CLUB
 Zrenjanin

Programi za 600XL/800XL i 130XE:

Naziv programa	usnimav.	format	vrsta	cena v dinarech
001 Acrobat	cload	C,D	igra	250
002 Action Biker	AB(start)	C	igra	300
003 Adressenverwal	cload	C,D	kor.	200
004 Adventure Creator	bic	C,D	igra	250
005 Alien Wars	cload	C,D	igra	60
006 Archon	bic	C,D	igra	300
007 Atari Basic	cload	C,D	kor.	200
008 Atari Bubbler	cload	C,D	igra	150
009 Atari Logo	bic	C,D	prj.	450
010 Atari Translator		D1	kor.	300
011 Atari Writer	xlt+bic	C,D	word	650
012 Atari Tick-Tock	cload	C,D	igra	200
013 Assembler	cload	C,D	prj.	300
014 Automatic Profreader	cload	C,D	kor.	150
015 Atlas 2 (Makroassembler)	bic	C,D	kor.	600
016 Bacanje kocaka	cload	C,D	igra	30
017 Baja Buggie	AB	C	igra	200
018 Balloonacy	cload	C,D	igra	150
019 Baseball	AB	C,D	igra	350
020 Basic Compiler		D1	kor.	700
021 Bc's Quest for Tires	bic	C,D	igra	250
022 Beach Head		D1	igra	300
023 Big Quest	cload	C,D	igra	150
024 Biorhythm	cload	C,D	kor.	200
025 Bioritmo 2	cload	C	kor.	300
026 Black Out	cload	C,D	igra	60
027 Bic	AB	C	kor.	150
028 Blue Max	bic	C,D	igra	300
029 Blue Max 2001	bic	C,D	igra	300
030 Black Box	cload	C,D	igra	200
031 Boccone	AB(start)	C	igra	150
032 Borba motora	cload	C,D	igra	100
033 Boulder Dash	bic	C,D	igra	350
034 Bruce Lee	AB	C	igra	300
035 Buck Rogers	bic	C,D	igra	300
036 Capture the Flag	bic	C,D	igra	200
037 Calendar Program	cload	C,D	kor.	150
038 Castle Combat	cload	C,D	igra	200
039 Castle Margue	cload	C,D	igra	300
040 Camelot	cload	C,D	igra	200
041 Caverns of Mars	AB	C,D	igra	300
042 Congo Bongo	bic	C,D	igra	250
043 Colossus Chess 3.0	AB	C	sh	500
044 Counter Action	cload	C,D	igra	200
045 Coda	cload	C,D	kor.	150
046 Colger	cload	C,D	kor.	30
047 Computer War	bic	C,D	igra	200

Pozadujete-li odpoved, prilozte do dopisu neorazitkovanou znamku v hodnotě 350 dinarů (lze koupit v prodejnách POFISu).

048	Circus	cload	C,D	igra	150
049	Crtanje	cload	C,D	kor.	40
050	Characters Editor	cload	C,D	kor.	200
051	Chess 7.0		D1	sah	500
052	Crystal Castles	blc	C,D	igra	250
053	Conan		D2	igra	450
054	CXL-Toolkit	AB	C	kor.	300
055	Dash	cload	C	igra	300
056	Datoteka	cload	C,D	kor.	150
057	Death Star	blc	C,D	igra	250
058	Deep Space	cload	C,D	igra	120
059	Desmond's Dungeon	AB	C	igra	300
060	Diamond Mine	cload	C,D	igra	200
061	Dig-Dug	blc	C,D	igra	200
062	Divlji zapad	cload	C,D	igra	100
063	Donkey Kong	blc	C,D	igra	300
064	Donkey Kong Jr.	blc	C,D	igra	300
065	Dos 3.0		D	kor.	300
066	Dos 3.5		D	kor.	500
067	Dreiba	cxt+blc	C,D	igra	250
068	Drop Zone	AB	C	igra	300
069	Apple Panic	blc	C,D	igra	300
070	Atari Chess	AB	C	sah	200
071	Captain Cosmo	blc	C,D	igra	250
072	Cohens Towers	blc	C,D	igra	300
073	Electracian	blc(p)	C	igra	300
074	F-Copy	blc	C,D	copy	1200
075	Feinscrolling	cload	C,D	kor.	30
076	Fische Fischen	cload	C,D	igra	100
077	Fighting	cload	C,D	igra	250
078	Flip and Flop	blc	C,D	igra	300
079	Formula 1	cload	C,D	igra	80
080	Forbidden Forest	AB	C	igra	300
081	Frecce Indiatrici	AB(start)	C	igra	150
082	Frogger	AB	C,D	igra	200
083	Frogger 3	blc	C,D	igra	300
084	Fruity Gambler	cload	C,D	igra	200
085	Galaxian	AB	C,D	igra	200
086	Gangsters	cload	C,D	igra	200
087	Ghostbusters	AB	C	igra	400
088	Gold Rausch	cload	C,D	igra	200
089	Golf	AB(start)	C	igra	200
090	Gradske odbrane	cload	C,D	igra	80
091	Java Jim	blc	C,D	igra	300
092	Journey to the Planets	blc	C,D	igra	350
093	Joust	blc	C,D	igra	300
094	Jungle Hunt	blc	C,D	igra	300
095	Jumper 2	cload	C,D	igra	250
096	Jumping Man	cload	C,D	igra	150
097	Kaboom	AB	C,D	igra	200
098	Kangaroo	blc	C,D	igra	300
099	Kanga	cload	C,D	igra	200
100	Kanopus Spiel	cload	C,D	igra	150
101	Kerzenheinz	cload	C,D	igra	100

102 King of the Castle	cload	C	igra	100
103 Krediti	cload	C,D	kor.	100
104 Gunfight	blc	C,D	igra	150
105 Landscape	blc	C,D	igra	250
106 Last Starfighter	blc	C,D	igra	300
107 Legionnaire	blc	C,D	igra	200
108 Le Travi Gemelle	AB(start)	C	igra	150
109 Lost in the Antarctica	cload	C,D	igra	50
110 Loto (7 od 39)	cload	C,D	kor.	30
111 Los Angeles 1964 Games	blc	C,D	igra	300
112 Manic Miner J	blc	C,D	igra	300
113 Mario Bros.	blc	C,D	igra	300
114 Memory	AB(start)	C	igra	150
115 Mikrobi	cload	C,D	igra	50
116 Miner 2049 er	blc	C,D	igra	300
117 Millipede	blc	C,D	igra	300
118 Mini golf	cload	C,D	igra	250
119 Microsoft Basic	blc(p)	C,D	prj.	450
120 Montezuma's Revenge	blc(p)	C,D	igra	350
121 Movie	cload	C,D	kor.	300
122 Mop	cload	C,D	igra	70
123 Monitor Programm	AB	C	kor.	300
124 Moon Patrol	blc	C,D	igra	200
125 Ms. Pacman	blc	C,D	igra	300
126 Mr. Cool	blc	C,D	igra	200
127 Mr. Do		D	igra	350
128 Music Construction	blc(p)	C,D	kor.	400
129 Mutation	cload	C,D	igra	100
130 MLX	cload	C,D	copy	900
131 Napad iz atmosfere	cload	C,D	igra	60
132 Necromancer	AB	C	igra	300
133 Neptune's Daughters	cxl+blc	C,D	igra	300
134 New York City	blc(p)	C,D	igra	250
135 Night Mission	blc	C,D	igra	350
136 Numbersort	cload	C,D	kor.	100
137 Paleta	cload	C,D	kor.	250
138 Pac-man	AB	C,D	igra	250
139 Pbs Chess	AB	C,D	igra	250
140 Pango	blc	C,D	igra	250
141 Pitfall	blc	C,D	igra	300
142 Pitstop	blc	C,D	igra	250
143 Pitstop 2	blc	C	igra	300
144 Pirat Adventure	blc	C,D	adv.	300
145 Pharaons Course	AB	C	igra	300
146 Planet of Doom	cload	C,D	igra	100
147 Players	cload	C,D	demo	150
148 Pole Position	blc	C,D	igra	300
149 Pooyan	blc	C,D	igra	250
150 Popeye	blc	C	igra	300
151 Preppie 2	blc	C,D	igra	300
152 Pool 400	cxl+blc	C,D	igra	250
153 Q-Bert	blc	C,D	igra	200
154 Qix	blc	C,D	igra	250
155 Quasimodo	blc	C,D	igra	300

156	Poker Sam	b1c	C,D	igra	300
157	Protector 2	cx1+b1c	C,D	igra	250
158	Race in Space	b1c	C,D	igra	200
159	Rainbow Walker	b1c	C,D	igra	250
160	Rat zvezda	cload	C	igra	150
161	Rasen Maeher	AB(start)	C	igra	200
162	Raster Blaster	b1c	C,D	igra	300
163	Rescue to Fractalus	b1c	C,D	igra	300
164	Renumber za Atari	cload	C,D	kor.	150
165	Rescue Mission	AB(start)	C	igra	250
166	Reflection	cload	C,D	igra	150
167	River Raid	AB	C	igra	300
168	Rotberg Synthesizer	b1c	C,D	demo	200
169	Roni, der Taschendieb	cload	C,D	igra	60
170	Robot	cload	C,D	igra	30
171	Rudnik	cload	C,D	igra	70
172	Sargon 2	AB	C	san	250
173	S-Copy 810		D1	copy	2000
174	Schatz Suche	AB(start)	C	igra	200
175	Sea Bandit	b1c	C,D	igra	200
176	Segelflugzeug	cload	C,D	igra	120
177	Senso	cload	C,D	igra	70
178	Shamus 2	b1c	C,D	igra	300
179	Snokie	b1c	C,D	igra	300
180	Soccer	b1c	C,D	igra	300
181	Solo Flight	AB	C	igra	450
182	Sound Demo	cload	C,D	demo	100
183	Sound Experimentier	cload	C,D	kor.	60
184	Sound Editor	cload	C,D	kor.	200
185	Space Eggs	b1c	C,D	igra	250
186	Space Invaders	AB	C	igra	150
187	Spelunker		D1	igra	450
188	Speed Script 3.0	AB	C	kor.	500
189	Slip	AB(start)	C	igra	200
190	Super Cubes (Rubic)	AB(start)	C	igra	200
191	Super Pacman	b1c	C,D	igra	300
192	Spy Vs. Spy		D1	igra	450
193	Squish'em	b1c	C,D	igra	200
194	Space Dodger	cload	C,D	igra	150
195	Suppergarten Sammeln...	cload	C,D	igra	50
196	Super Snupe	cload	C,D	igra	150
197	Stargate	b1c	C,D	igra	150
198	Star Raiders	b1c	C,D	igra	200
199	Strip Poker		D1	igra	400
200	Strip Poker 3		D1	igra	400
201	Straussenkampf	cload	C,D	igra	120
202	Hard Hat Mac	b1c	C,D	igra	300
203	Harvey	AB	C	igra	100
204	Hc Text Atari	cload	C,D	word	300
205	Hires Text	cload	C,D	word	200
206	Hockey	b1c	C,D	igra	250
207	One on One (T)		D	igra	400
208	Tape to Tape	cload	C,D	copy	850
209	Tape Duplicate	cload	C,D	copy	900

210	Target Practice	AB(start.)	C	igra	200
211	Tennis	bic	C,D	igra	250
212	Turbo Taps	AB(start.)	C	kor.	400
213	The Flight of the Swan	cload	C,D	igra	250
214	The Spy Strikes Back	bic	C,D	igra	200
215	Trull	cload	C,D	igra	150
216	Tutankhamun	bic	C,D	igra	200
217	Unknown	bic	C,D	igra	250
218	Viergewint	cload	C,D	igra	70
219	War Games	bic	C,D	igra	250
220	Willy Stronschlag	cload	C,D	igra	70
221	Wizard of Wor	bic	C,D	igra	300
222	Wortsuchspiel	cload	C,D	igra	60
223	Zauberwurm	cload	C,D	igra	120
224	Zaxxon	AB	C	igra	250
225	Zaxxon 2	bic	C,D	igra	300
226	Zeppelin	bic	C,D	igra	300
227	Supply Blaster	cload	C,D	igra	200
228	Star Texter		D1	word	1000
229	Scanalyzer		D1	kor.	1500
230	Sam Speech Syn.	bic*	C,D	kor.	400
231	Black Disk		D	copy	1900
232	S-Copy 2		D	copy	1900
233	Disk Manifold		D	copy	1900
234	N-Dos Converter		D	kor.	1000
235	Amiga Ball Demo	bic(p)	C,D	demo	300
236	Turbo Basic XL 1.5	bic(p)	C,D1	kor.	700
237	Nato Commander	cload	C	igra	300
238	Dos 2.5		D1	kor.	300
239	Mankala	AB	C	igra	150
240	Insects	AB	C	igra	200
241	Canyon	AB	C	igra	250
242	Equestrian	AB	C	igra	300
243	TT Racer	AB	C	igra	200
244	Deutsche D&D Adventure	AB	C	adv.	200
245	Hockey (Gamma)	AB	C	igra	200
246	Atari Basketball	AB	C	igra	300
247	Space Rescue	cload	C,D	igra	150
248	Ritter Fuer Einen Tag	cload	C,D	igra	200
249	Hexa Pawn	cload	C,D	igra	150
250	Lov na majmunove	cload	C,D	igra	150
251	Jumper	cload	C,D	igra	200
252	Fort. Apocalypse		C,D	igra	300
253	Super Bunny		C,D	igra	300
254	Super Zaxxon		X,D	igra	400
255	Caverns of Khafka		X,D	igra	350
256	Caverns of Mars 2		C,D	igra	350
257	Fast Eddie		X,D	igra	200
258	Whistler's Brothers		D1	igra	600
259	Trains		D1	igra	500
260	Zorro		D1	igra	500
261	The Great American Cross- Country Road Race		D1	igra	600
262	Wizard & Princess (T)		D2	adv.	500

UŽIVATELSKÝ POPIS POČÍTAČŮ (ATARI 600XL/800XL)

Překlad:
Vydal:
Tisk povolil:
Náklad:
Tech. redakce:

Ing. Věroslav SZPUK
Klub mikroelektroniky Zvazarmu, LEVICE
ONV Levice, odbor kultúry, č. j. 254/86
150 výtisků
O. Burser

Publikované zo súhlasom AK Tlmače.

Igi (2026)