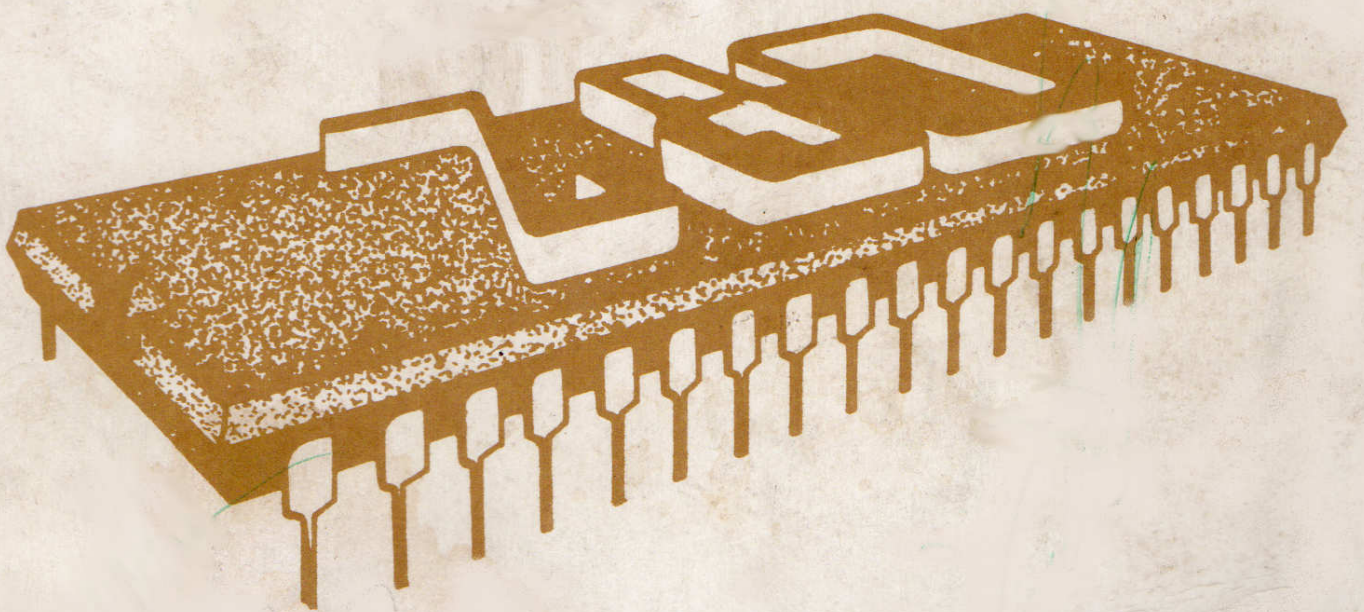
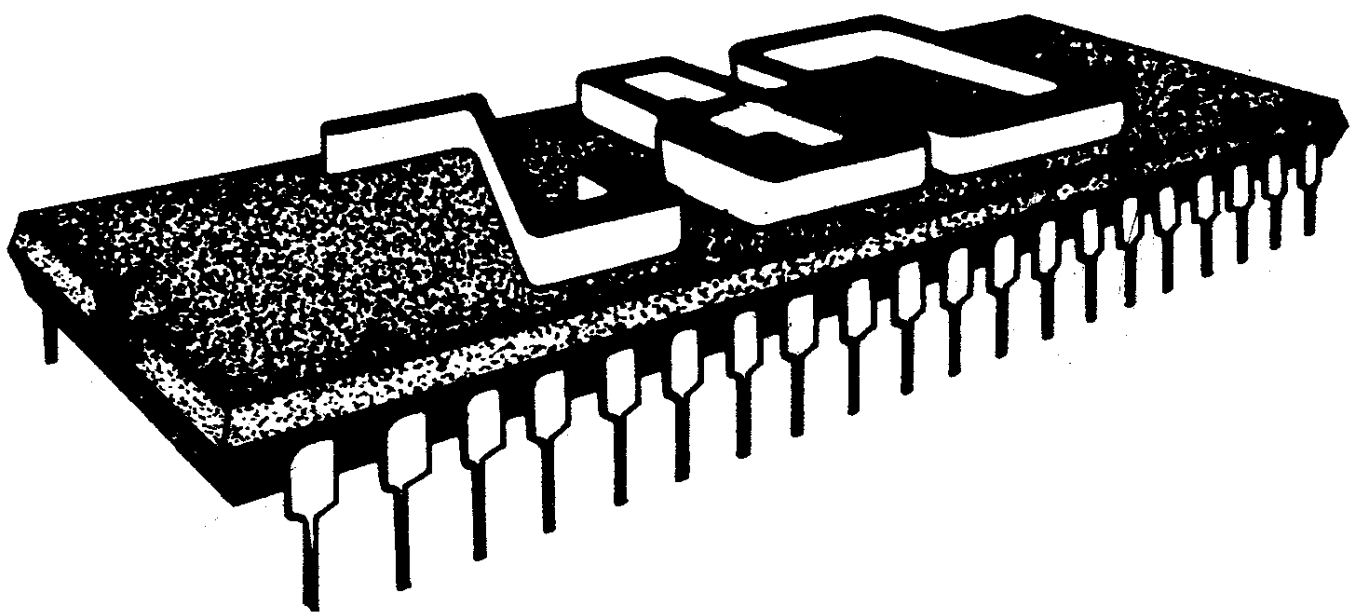


ZX SPECTRUM -BÁDATEL'



ZX SPECTRUM -BÁDATEL'



Táto publikácia je určená pre všetkých majiteľov niektorého z širokej rodiny počítačov vychádzajúcich zo štandardu SINCLAIR ZX SPECTRUM:

ZX Spectrum /+128K/+2/+2A/+3, Delta, Didaktik GAMA, Didaktik M.

Z X S P E C T R U M - B A D A T E I L

O B S A H

1.0	Programovanie v Basicu	007
1.1.1	Kameň, nožnice, papier	010
1.1.2	Kameň, nožnice, papier - vo farbe	012
1.2	Účitel ovládania klávesnice	013
2.0	Experimentovanie s farbami	016
2.1	Pretlač	020
2.2	ATTR kalkulačtor	021
2.3	Duha	021
2.4	128 farieb	022
2.5	Sieť	023
2.6	Ladová hviezda	023
2.7	Nočná obloha	024
2.8	Náhodné kruhy	024
2.9	Gulečniková guľa	025
2.10	Kužel	025
3.0	Experimentovanie so zvukom	026
3.1	Hudobník	027
3.2	Klavír	028
3.3	Skladateľ	028
3.4	Premena obňa	030
3.5	Prechádzka	031
3.6	Bomba	032
3.7	Tvarovanie frekvencie a času	032
3.8	Frekvenčné tvarovanie	032
3.9	Muzikograf	033
4.0	Použitie v obchode	033
4.1	Osobné konto	037
4.2	Diár	039
4.3	Telefónna ústredňa	041
4.4	Databáza	042
4.5	Finančný model odbytu	044
5.0	Použitie vo výchove	047
5.1	Test delenia	049
5.2	Aritmetický kvíz	050
5.3	Rovnice	055
5.4	Mačky a predmety	057
5.5	Anglický slovníček	058
5.6	Majster kvízu	060
5.7	Test čítania	062
5.8	Kreslička	063
6.0	Hráme hry	065
6.1	Nočný prípad	066

1.0 P R O G R A M O V A N I E V B A S I C U

BASIC je svetovo preslavený populárny programovací jazyk, pretože je ľahký na učenie. Počítačové jazyky majú rôzne úrovne komunikácie. Najvyšší jazyk je ten, ktorý je spojený s angličtinou (BASIC), jazyky najnižšej úrovne sú spojené s kuzelným drmlením jedničiek a nul, ktorým počítač rozumie.

BASIC je vysokourovňový jazyk. Ak nemáte skúsenosti v programovaní, prosíme vás, učte sa od tohto bodu každý bit BASICu.

Slova ako PRINT, STOP, AND sú tiež ako v angličtine. Taktiež aj slova IF, THEN a OR. Domnievam sa, že programovanie je jednoduchšie, než sa vám môže zdať.

Predovšetkým si uvedomte, že ak píšete program, zadávate vlastne Spectru sériu postupných inštrukcií. Spectrum, tak ako všetky počítače, bude dáta a má mimoriadne počtové a dosadzovacie schopnosti, ale absolútne a vôbec nie imaginárne. Ak niečo prikažem počítaču, tak to spraví. Ak vypustíme časť inštrukcií, bude to atentať na obsah základu informácie a realizuje sa čosi iného alebo zle.

Obrazne máte robota služobníka, ktorý príde do kúpeľne. GO TO BATH môže byť prvá inštrukcia v programe. IF vaňa je prázdna THEN začni napúšťať. IF vaňa je plná THEN ukonči napúšťanie vody. Robot šťastne svoju elektronicou rukou nájde kohútik a báda či je prázdno a či otvorí prítok. Otvorí prítok a čaká večne na rovnováhu hladiny, ktorú má detekovať, aby mohol potom zavrieť prívod. Ale to preto, že ste zabudol pripojiť inštrukciu: Napušťaj IF je zátka v otvore a IF zátka nie je v otvore THEN daj zátku do otvoru. Robot nebude myslieť za vás.

Takto presne Spectrum pracuje, vykonáva vaše inštrukcie. A nech sa to zdá, akokoľvek kľamlivé, Váš počítač môže bud sledovať inštrukcie dané vami, alebo ich môže zadávať napr. robot, pretože tieto inštrukcie (IF, THEN, GO TO) obsahujú aj slovná zásoba programovacieho jazyka BASIC.

Tu je jednoduchý program pre Spectrum, ktorému môžete bezpečne rozumieť bez tréningovania programovacieho jazyka.

```
LET A = 20
LET B = A + A
IF B = 40 THEN PRINT "B sa rovná 40"
```

Toto je nefalšovaná angličtina (slovenčina) a zároveň je to tiež BASIC. Prvý riadok programu (LET A = 20) je jednoduchý, tak isto i riadok druhý (LET B = A + A). Tretí riadok je pozoruhodne podobný predošlej robotovej inštrukcii ako IF vaňa je prázdna THEN otvor prívod. vid: IF B = 40 THEN PRINT B "B sa rovná 40". Mohol by ste tieto nápisy pomocou Spectra vypísať rovnako ako B rovná sa 40. K prevádzke programu potrebujú tieto tri riadky len jediná vec a to sú čísla pred každým riadkom. Nasleduje program, ktorý Spectrum uvedie do činnosti.

```
10 LET A = 20
20 LET B = A + A
30 IF B = 40 THEN PRINT "B sa rovná 40"
```

Čísla riadkov môžu byť také aké si vyberiete (od 1 do 9999). Počítač ich automaticky usporadúva. Radte čísla príkazov s krokom 10, tak vznikne priestor medzi riadkami a ak sa rozhodnete môžete doplniť (príčiť) riadky neskôr.

Tak, ako vidíte, máme už nejakú skúsenosť v programovaní BASIC slov,

LET, IF ... THEN a PRINT. Rovnítko a znamienko plus poznáte z riadnej matematiky. V mnohých prípadoch sa chovajú v BASICu ako pri normálnych súčtoch. Predtým sme aktualizovali zatku v akomsi programovaní. Chcem uviesť iné slovo, ktoré už všeobecne poznáte. Fakticky sú to dve slova GO a TO (chod na).

V BASICu sú vždy spolu ako GO TO a sú použiteľné jednu klávesou Spectra (klávesa G). Budeme ich volať tak, ako skôr náš robot, ktorý štartoval kúpeľňový program vyriešením GO TO BATH. Vo svete programovania hovorme GO TO a číslo riadku. Môžeme pridať konečný riadok, ktorý pod tromi riadkami čítame:

```
40 GO TO 30 (chod na riadok č. 30)
```

Výkonáva sa riadok 30 a tlačí B sa rovná 40, potom sa to pohybuje k riadku 40, kde nájde inštrukciu GO TO 30. Okamžite sa výkon programu presunie na príkazový riadok s číslom 40. Opäť sa vypíše text B sa rovná 40 a celý proces sa opakuje až kým sa nezaplní celá obrazovka.

KLÁVESNICA

Dve najvýznamnejšie súčasti klávesnice sú SHIFT kláče. V najspodnejšom rade je CAPS SHIFT, v rohu nalavo. SYMBOL SHIFT je druhý v spodnom rohu pri pravej ruke.

Tieto kláče - klávesy určujú čo sa stane, keď stlačíte inú klávesu. Pre začiatok budeme predpokladať, že je počítač zapnutý. Stlačte nejaké písmeno a budú sa písať klúčové slová (hľadajte napr. LOAD, LIST alebo PRINT). Jedna z číest pre Spectrum, využítie jeho pamäte a usporiadanie pre veľmi jednoduchý program je použitie klúčového systému. Jednoduche stlačenie kláves pre celé klúčové slová. Veľa iných počítačov má typ so slovom LIST, DIM alebo FOR, ale viac nie. Držte CAPS SHIFT a stlačte DELETE (celkom vpravo v hornom rohu) pri súčasnom stlačení budú znaky vymazané. Teraz stlačíme klúč P a objaví sa slovo PRINT. Potom stlačte akýkoľvek numerický taster, takže máme napr. PRINT 62735 na spodnom okraji obrazovky. Teraz stlačíme klúč označený ENTER, v pravej dolnej časti klávesnice. Obrazovka bude svetlá a žiadané číslo sa objaví v najhornejšej časti obrazovky. Počítač sa podriadiť vašim inštrukciám PRINT a číslam. Spectrum všeobecne čaká pokiaľ nestlačíte ENTER, v prípade, že nerobíte nič iné - nepokračuje.

Stretneme sa s jednoduchým programom, ktorým ukážeme, ako sa používa klúčový systém. Nasledujúci typ tvoríme držaním CAPS SHIFT a stlačením klávesy 2 (CAPS LOCK). Nasledujú veľké písmena v programe.

```
10 INPUT (teraz stlačte klávesu X a potom ENTER, ktorý ukončí programový riadok)
20 PRINT X (stlačte ENTER)
30 GO TO 10 (stlačte ENTER)
```

Pre PRINT je taster P a GO TO je taster G. Medzery v programe sú generované automaticky. Ak je program v počítači, môžete stlačiť R taster, ktorým vytvoríte slovo RUN a odosleť ho ENTERom. Svetielkujúci kurzor umiestnený na spodnom okraji obrazovky čaká na číslo. Vstup čísla potvrdíme stlačením ENTERu. Číslo budete vidieť celkom hore na obrazovke. Viete bez diskusie, že GO TO 10 prevedie beh programu na riadok číslo 10. Takže počítač bude nepretržite prevádzat v programe slučku stále až do

STOP. Stop programu môžeme previesť použitím písmena A, keď Spectrum čaká na číslo. Počítač sa zastaví a tlačí chybu určenia na spodnom okraji obrazovky.

Akonáhle stopnutý program vymažeme stlačením A tastru a zatlačením ENTER, obrazovka scernie, potom zasvieti (zjasnie) s copyrightovou notáciou na spodnom okraji obrazovky.

Príkaz (klúčové slovo) pre písmeno A je NEW, ktorý vyčistí všetko v počítačovej pamäti, takže ho musíte používať rozumne.

Takto sa pracuje s klúčovými slovami, ktoré dostanete stlačením tastrov po čísle riadku. Pozrieme sa na slová ako SYMBOL SHIFT klúčom. Podržte SYMBOL SHIFT a stlačte taster V. Uvidíte, ako sa objaví slovo AND. Teraz podržte SYMBOL SHIFT, stlačte G a objaví sa THEN. Čiže červenou zvýraznené slová na taastroch a symboly + a znak libry dostanete držaním SYMBOL SHIFT a potom stlačením žiadaných klúčov (tastrov).

Teraz prichádzajú na rad príkazové slová zobrazené v hornej časti kláves. Získame ich stlačením oboch presuvných kláves (CAPS a SYMBOL) súčasne a potom stlačením dotýčného tlačítka. Skuste to. Stlačte CAPS SHIFT a SYMBOL SHIFT súčasne a potom sa dotknite tastra A. Objaví sa slovo READ.

Trochu obťažnejšie je podávať slová v spodnej časti kláves. Stlačte oba kláče súčasne a podržte CAPS SHIFT až do okamihu, keď stlačíte klávesu s hľadaným príkazom. Napr. stlačte X taster. Objaví sa slovo INK. Je možné, že máte prax pre zabezpečenie vstupu slova INK (alebo BEEP, PAPER, FLASH alebo BRIGHT). Prečítajte si túto lekcii znovu.

Ak tomu nebudete bezpečne rozumieť, urobte si s každou verziou cvičenie. Netrápte sa, ak sa vám zdá, že to trvá dlho. Budete príjemne prekvapení, ako živo a rýchlo sa naučíte ovládať klávesnicový systém. Ku koncu tejto kapitoly máme program označený ako učenie pre vaše povzbudenie "tipovanie stlačením", ale najskôr vás potrebujeme uviesť do prvého reálneho programu v tejto knižke. Môže vás omrzieť opísanie, ale prosím, vytrvajte. Pokiaľ zadáte celý program do vášho počítača, stlačte RUN a hrajte hru opakovane so Spectrum. Program hrá ROCK, SCISSORS, PAPER - (ďalej kameň, nožnice, papier) v počítačovej verzii spoločenskej hry, v ktorej dvaja hráči skrývajú svoje ruky vzadu za svojím chrbtom a označujú rukou vytvorený symbol pre kameň (uzavrená ruka), pre nožnice (dva vztyčené prsty) alebo papier (otvorená ruka). Kameň poráža nožnice (pretože kameň tupí nožnice), nožnice bijú papier (pretože nožnice môžu strihať papier) a papier bije kameň (pretože papier môže zabaliť kameň).

Vstupujete preto do programu voľbou kameň, nožnice, papier označené číslami 1, 2, 3. Vaša voľba je hore na obrazovke. Pokiaľ zasahujete do programu, môžu vám pomôcť poznámky.

Jednoducho signujeme znak + (prvýkrát použité v riadku 30). Držte SYMBOL SHIFT a potom stlačte L.

Nezabudnite stlačiť ENTER po napísaní každého programového riadku. Ten (riadok) sa potom presuva zo spodku obrazovky do programu. Ak sa riadok odmietne presunúť, objaví sa na riadku bližajúci otáznik. Tým označuje počítač nejakú odhalenú chybu v riadku. Prehľadnite si pozorne miesto, kde bliká otáznik a hľadajte tam chybu.

Dvojbodka (:) po slove CLS (znamená Clear The Screen - mazanie obrazovky) v riadku 60 je shiftovaný taster (klúč) Z. Podrž SYMBOL SHIFT a stlač Z. Riadok 70 má jednoduchý (jediný) apostrof (') za slovom PRINT, predtým než sa otvorí úvodzovky ("). Apostrof nájdete na 7. tastru a získame ho podržaním SYMBOL SHIFT a stlačením tastru 7. Dvojité úvodzovky sa nachádzajú na tastru P.

Ak potrebujete výcvik a ak máte trochu záujem, hľadajte, či riadok

170 je v počítači presne (správne). Po 170 LET C =, potrebujete slovo INT, ktoré nie je písane písmeno po písmene, dostanete ho z R tastra. Stlačte oba kľúče SHIFT, uvoľnite ho skôr než stlačíte R a INT bude realizované. Otvorená zátvorka je na tastru 8 (podrž SYMBOL SHIFT pokiaľ stlačíte taster 8) a slovo RND je prístupné tasterom (klucem) T. Stlačte oba shiftové kľúče naraz, uvoľnite ich, potom stlačte T. Hviezdička, ktorou realizujeme násobenie v BASICu pochádza z tlačítka B, a dostaneme ju podržaním SYMBOL SHIFT a tlačítka B. Neskúšajte hlásťovať slová ako AND, OR, THEN a TO, ale nájdite ich v celku. Hľadáme ich ako kľúč. Tieto slova (klúče) obdržime podržaním SYMBOL SHIFT a stlačením príslušného tlačítka.

1.1.1 K A M E Ň, N O Ž N I C E, P A P I E R

```

10 REM Kameň, nožnice, papier
30 LET COMP = 0
40 LET HUM = 0
50 FOR A = 1 TO 10
60 CLS: PRINT "Číslo partie: "; A
70 PRINT " 1 - kameň"
80 PRINT " 2 - nožnice"
90 PRINT " 3 - papier"
100 PRINT "Zadaj 1,2 alebo 3"
110 INPUT B
120 PRINT "Zvolil ste: "
130 IF B = 1 THEN PRINT "kameň"
140 IF B = 2 THEN PRINT "nožnice"
150 IF B = 3 THEN PRINT "papier"
160 PAUZA 50
170 LET C = INT (RND * 3) + 1
180 PRINT "Ja, počítač, volím: "
190 IF C = 1 THEN PRINT "kameň"
200 IF C = 2 THEN PRINT "nožnice"
210 IF C = 3 THEN PRINT "papier"
220 LET D = 260
230 IF B = C THEN PRINT "Remíza!"; GO TO D
240 IF C = 1 AND B = 2 OR C = 2 AND B = 3 OR C = 3 AND
B = 1 THEN PRINT "Výhral som!"; LET COMP = COMP + 1: GO TO D
250 IF B = 1 AND C = 2 OR B = 2 AND C = 3 OR B = 3 AND
C = 1 THEN PRINT "Vý ste vyhrali!"; LET HUM = HUM + 1
260 PRINT " skóre "; " moje "; COMP; TAB 10; " vaše "; HUM
270 PAUZA 200
280 NEXT A
290 IF HUM = COMP THEN PRINT "REMÍZA !"
300 IF HUM < COMP THEN PRINT "výhral som celú hru."
310 IF COMP < HUM THEN PRINT "výhral ste celú hru. Gratulujem! "
```

Ak budeme programom riadok po riadku prechádzať, tak nám použité príkazov prinesie rôzne prekvapenia na každom riadku.

10 Toto je štart pomocou REM - (REMARK). Správa za REM je v programe len pre poznámky. Počítač text poznámky ignoruje, kedykoľvek sa objaví pri programovom listingu za príkazom REM
30 LET znamená aby, nech. Slovo COMP (skrátene pre počítač) je pre volanie premennej, ktorej meno je kombináciou písmen a čísel a

reprezentuje numerickú hodnotu. COMP uchováva skóre dosiahnuté počítačom a rovná sa nule pri štarte hry.

40 To isté sa týka volania premennej HUM pre skóre hráča.

50 FOR... To je štart, kde sa volá FOR-NEXT slučka. Počítač prechádza programom a sleduje číslo slúčky, posledné číslo v stave FOR. Pozrite sa na riadok 280 (NEXT A). Toto je koniec FOR-NEXT slúčky. V tomto programe program ruší zákon pre počítač vykonávať riadky postupne podľa čísel a usmerňuje chod príkazmi FOR-NEXT.

60 CLS - zmlenili sme sa už skôr. Znamená mazanie obrazovky. Dvojbodka (:) oznamuje ďalší príkaz v riadku (možnosť viacerých príkazov v riadku). PRINT tu píše poradie partie v cykle 10 hier.

70-90 Tieto tri riadky tlačia čísla 1 až 3 a prislúchajúce slová

100 Apostrof je použitý pre umiestnenie prázdneho riadku medzi "zadaj 1,2 alebo 3"

110 Príkaz INPUT čaká pokiaľ nie je použitý vstup čísla. V tomto prípade číslo vášho vstupu je označené ako premená B.

120 Tu sa tlačí "Zvolili ste: "

130-150 Tieto tri riadky interpretujú číslo, ktoré vstupuje (1,2 alebo 3 - označené ako B) a rozhoduje, ktoré slovo (kameň, nožnice alebo papier) tu bude stáť a bude vytlačené na obrazovku.

160 PAUSE tu spôsobíme krátku zdržanie v programe. Vytvorí sa slučka.

170 Toto je veľmi zaujímavý riadok, v ktorom počítač generuje náhodné čísla. Náhodné čísla sú veľmi užitočné v hrách. RND funkcia (oba kľúče SHIFT a potom stlačiť T) generuje náhodné čísla medzi nulou a jedna.

Uvidíte ukážku príkazom PRINT RND a potom stlačením ENTER. Teraz vychádzajú čísla medzi 0 a 1. Riadok 170 mení tieto čísla do rozsahu 1 až 3. Ak nahradíme trojku, ktorá nasleduje za hviezdičkou spolu so

zátvorkami, na 10, potom C bude rovnaké ako číslo volené generátorom náhodných funkcií medzi 1 a desať. Pokiaľ, ako je tomu v tomto prípade čaká počítač voľbu 1, 2 alebo 3, tak násobíme RND tromi.

180 Tu začína návstievie zobrazujúce, čo chce počítač voľiť.

190-210 Tieto riadky menia 1,2 alebo 3 získané z C, na požadované "papier", "nožnice", "kameň".

220 Premenná D je rovná 260.

230 kontrola, či nenastal stav, že počítač aj vy ste uskutočnili rovnaku voľbu.

240 kontrola, či vyhral počítač. Ak áno, potom sa zvýši premená COMP o jednotku. COMP uchováva informáciu o počte víťazstiev nad vami.

250 kontrola, či ste vyhral vy. Ak áno, potom sa zvýši premená HUM o jednotku. HUM uchováva informáciu o počte víťazstiev nad počítačom.

260 Tlačíte priebežný výsledok skóre.

270 PAUSE - pauza vám umožní prečítať stav hry na obrazovke.

280 NEXT A je koniec pre FOR-NEXT slučku, ktorá započala na riadku 50. T.j. zvýši sa A o jedna. Ak je A<10, tak beží program znova od prvého príkazu za FOR. Ak už A dosiahlo 10, potom program pokračuje na nasledujúcom riadku za NEXT A.

290 Ak hodnota označená ako HUM rovnaká ako premená označená COMP potom Spectrum vie, že hra skončila remízou. Vytlačí sa výsledok.

300 Ak HUM je menší než COMP, počítač oznámi, že ste prehral.

310 Ak COMP je menší ako HUM, počítač vie, že ste vyhral vy.

Osvedčený spôsob, ako si najrychlejšie osvojíte slová a prácu v BASICU, je metóda praktického rozoberania malých programov. Tak získate nové vedomosti a aj potrebnú prax. Na to slúži i uvedený komentár.

DODATOK O TROCHE FARIEB

Budeme oboznámení s dosiahnutím tej najväčšej kapacity farieb i zvuku u Spectra v BASICU, ale o trochu neskôr v tejto knižke. Ale teraz by sme vám radi uviedli a zoznámili vás s jednoduchšími príkazmi (povelmi). Pohráme sa s príkazmi na farbu textu. Nasleduje program kameň, nožnice, papier s pridanými príkazmi. Môžete jednoducho modifikovať program použitím jednoduchej editácie.

1.1.2 K A M E Ň, N O Ž N I C E, P A P I E R - V O F A R B E

```

10 REM Kameň, nožnice, papier
30 LET COMP = 0
40 LET HUM = 0
50 FOR A = 1 TO 10
60 CLS: PRINT INK 2; "Číslo partie "; INK 1; A
70 PRINT INK 1; "kameň"
80 PRINT INK 2; "nožnice"
90 PRINT INK 3; "papier"
100 PRINT INK 6; PAPER 2; "Zadaj 1,2 alebo 3"
110 INPUT B
120 PRINT INK B; "Zvolili ste: ";
130 IF B = 1 THEN PRINT "kameň"
140 IF B = 2 THEN PRINT "nožnice"
150 IF B = 3 THEN PRINT "papier"
160 PAUSE 50
170 LET C = INT (RND * 3) + 1
180 PRINT INK C; "Ja, počítač, volim: ";
190 IF C = 1 THEN PRINT "kameň"
200 IF C = 2 THEN PRINT "nožnice"
210 IF C = 3 THEN PRINT "papier"
220 LET D = 260
230 IF B = C THEN PRINT FLASH 1; "Remíza!": GO TO D
240 IF C = 1 AND B = 2 OR C = 2 AND B = 3 OR C = 3 AND B =
    THEN PRINT FLASH 1; BRIGHT 1; "Vyhral som!": LET COMP =
    COMP + 1; GO TO D
250 IF B = 1 AND C = 2 OR B = 2 AND C = 3 OR B = 3 AND C = 1
    THEN PRINT FLASH 1; "Vý ste víťaz": LET HUM = HUM+1
260 PRINT "skore "; ; moje "; COMP; TAB 10; " vaša "; HUM
270 PAUSE 200
280 NEXT A
285 INVERSE 1
290 IF HUM = COMP THEN PRINT "REMÍZA !"
300 IF HUM < COMP THEN PRINT "Vyhral som celú hru!"
310 IF COMP < HUM THEN PRINT "Vyhrali ste celú hru. Gratulujem!"

```

Vidite, že v riadku 60 boli doplnené slova INK 2 a 1. Aby mohli byť zavedené do riadku stlače LIST (klávesa K) potom 60. Ak máte LIST 60 potom stlače ENTER. Zobrazí sa výpis programu a pretože je program dlhší než počet riadkov obrazovky, tak sa dole zjavi otázka "scroll?" - ešte?. Odpovieme N prípadne BREAK, keď nechceme pokračovať v listovaní programom. Ak chceme vidieť aj nasledovné časti programu odpovieme ľubovoľnou klávesou, samozrejme okrem N a BREAK. V našom prípade listovanie prerušíme klávesou N, podržíme CAPS SHIFT stlačíme klávesu 1, ktorá má funkciu EDIT.

Tak dosiahneme, že sa nám starý 60. riadok okopíruje do spodnej, dialógovej časti obrazovky. Šipkami nastavíme kurzor na pozíciu, kde chceme previesť patričné zmeny. Ako vypísať kľúčový príkaz INK sme si už vysvetlili. Číze po doplnení riadku o príkazy INK 2,1 stlačíme ENTER. Ak sme napravili chyby, tak sa opravény 60. riadok zaradí na svoje miesto v programe. Pri jeho vykonávaní už budú použité aj nové príkazy na farbu. Ak spustíte program, uvidíte slova "Číslo partie: " v červenej farbe. Farbu môžeme meniť číslami 0 až 7 (ako bude podrobne vysvetlené neskôr). Napr. INK 1 spôsobí zmenu farby textu na modrú, INK 2 zmení farbu na červenú atď. (nepotrebuje farbu meniť pri každom príkaze PRINT, pokiaľ program pokračuje plynule ďalej).

AK SÚ FARBY NEJASNE OTÁČAJTE GOMBÍKOM FAREBNEJ KONTROLY NA FTV POKIAĽ NEUVIDÍTE SPRÁVNU FARBU MODRÚ A ČERVENÚ. ROZDIEL TVORÍ REPRODUKČIA TELEVIZNÝCH FARIEB, TAK NAPR. MÔŽETE HĽADAŤ JEDNU FARBU INTENZÍVNEJŠIE NEŽ DRUHÚ.

Samozrejme, že rôzne farby sa na ČB TV zobrazia ako odtiene šedi. AJ V TOMTO PRÍPADE JE DÔLEŽITÉ STAROSTLIVO SI NASTAVIŤ UROVEŇ JASU A KONTRASTU, ABY STE DOKAZALI ROZLIŠIŤ VŠETKY STUPNE FAREBNÉHO SPEKTRA POČÍTAČA. V OPAČNOM PRÍPADE BY STE MOHLI VNÍMAŤ NIEKTORE VIACFARBÉNE PLOCHY AKO JEDNU FARBU, ČO ZNAČNE SKRESUJE OBRÁZOVU INFORMÁCIU.

Môžete teraz pre celý program priradiť farby podľa vlastnej fantázie. Riadky ktoré potrebujete modifikovať sú 60, 70, 80, 90, 100, 120, 180, 230, 240, 250 a 285 (riadky, v ktorých musia nastať zmeny budú triedené samozrejme po poriadku a automaticky, môžete ich editovať bez ohľadu na ich poradové čísla v programe). Uvedte si sami program do činnosti povelom RUN a sledujte ako príkazy uvádzajú farby do života. Nezabudnite, že hodnoty pre farbu v poveloch INK a PAPER môžete vybrať z intervalu 0-Čierna až 7-biela.

1.2 U Č I T E Ľ O V L Á D A N I A K L Á V E S N I C E

Naš konečný program v tejto úvodnej kapitole našej knihy je určený ako pomôcka na vašej ceste po klávesnici. Ak zadáte RUN pre váš program, budú vstupovať čísla alebo písmená abecedy, volené alebo náhodné na obrazovku pravdepodobne v tom tvare ako na klávesnici. Máte limitovaný čas na stlačenie označenej klávesy. Ak na to idete zle, ospravedlniteľný počítač ide inou cestou. Budete prekvapení ako rýchle sa naučíte využívať usporiadanie klávesnice používajúc túto rutínu (podprogram).

AK SA VÁM ZDÁ ČASOVÝ INTERVAL KRÁTKY, TAK ZMENEŤTE HODNOTU 100 V RIADKU 180 NA VÄČŠIE ČÍSLO. ŠTART NA 300 A STUPŇOVITÁ GRADÁCIA POKIAĽ NIE JE ČAS PRÍBLIŽNE PRESNÝ, PRE VÁS DÁVA PRAVIDLO TYPICKÉHO VYSKÚŠANIA. PRI OPAĽNOSTI VYCHÁDZAJÚ SPRÁVNE ČÍSLA A DATA V DATA PRÍKAZE (RIADKY 80 A 110) A TÁTO KONTROLA KDE PREMENNÉ A PÍSMENÁ BUDÚ VYTĽAČENÉ. POTOM AŽ BUDEME MAŤ PROGRAM ROVNAKÝ VENUJTE TROCHU ČASU A VRÁŤTE SA SPÄŤ K TĚJTO KNÍŽKE A PRÉJDITE RIADOK ZA RIADKOM.

```

10 REM Učiteľ ovládania klávesnice
20 DIM A$ (36,5)
30 LET SKORE = 0
40 FOR A = 1 TO 36
50 READ B$
60 LET A$ (A) = B$
70 NEXT A
80 DATA "10602", "20605", "30608", "40611", "50614", "60617",

```

```

"70620", "80628", "90626", "00629"
90 DATA "A1202", "B1517", "C1511", "D1208", "E0908", "F1211",
"G1214", "H1217", "I0923", "J1220"
100 DATA "K1223", "L1226", "M1523", "N1520", "O0926", "P0929",
"Q0902", "R0911", "S1205", "T0914"
110 DATA "U0920", "V1514", "W0905", "X1508", "Y0917", "Z1505"
120 REM
130 REM Test
140 REM
150 FOR A = 1 TO 10: BEEP.02,-10:CLS
160 LET B = INT (RND * 36) + 1
170 PRINT FLASH 1; AT VAL (A$ (B,2 TO 3), VAL A$(B,4 TO 5)); A$ (B,1)
180 FOR C = 1 TO 100
190 LET C$ = INKEY $
200 IF C$ <> "" THEN GO TO 260
210 NEXT C
220 PRINT AT 0,0; INK 7; PAPER 4; FLASH 1; BRIGHT 1; "NESKORO !!!"
225 FOR D = 1 TO 70: NEXT D
230 NEXT A
240 GO TO 300
260 IF C$ = A$ (B,1) THEN LET SKORE = SKORE + 1; BEEP.25.
SKORE *4; PRINT AT 16,0; PAPER 6; INK 2; "Dobre, vaše skóre je
teraz "; SKORE
265 IF C$ <> A$ (B,1) THEN PRINT AT 0,0; INK 1; PAPER 6; FLASH 1;
"Je mi ľúto, mali ste smolu."; TAB 10; "Skuste znova.": GO TO 170
270 FOR D = 1 TO 100
280 NEXT D
290 NEXT A
300 CLS: PRINT AT 8,2; FLASH 1; BRIGHT 1; PAPER 6; INK 2;
"Vaše skóre bolo "; SKORE;
310 IF SKORE > 7 THEN PRINT AT 10,7; FLASH 1; INK 4; PAPER 7;
BRIGHT 1;"Gratulujeme !!!"

```

10 tu je REM (spomeníte si REMark je označenie komentára, poznámky) počítačom ignorovaný príkaz. Poznámkou môžete vyplniť priestor za REM, tak ako vám to vyhovuje.

20 DIM vyhradi priestor v pamäti pre pole hodnôt. V našom prípade ide o pole textových reťazcov z dát, ktoré budú neskôr konvertované na čísla. 30 ako výsledok vašeho snaženia pri práci s programom sa bude vypočítavať úspešnosť vášho tipovania na klávesnici. K uchovaniu tejto informácie o úspešnosti slúži premenná skóre. Tá musí byť na začiatku inicializovaná na nulovú hodnotu, pretože ste zatiaľ nemali ani jeden pokus.

40-70 toto je FOR-NEXT slučka ako v predchádzajúcom programe. Riadok 50 (READ B\$) je aktivovaný pomocou návestia DATA (riadky 80 - 110). Číta sa zakazdým keď sa táto slučka opakuje, a to jeden krát za jeden cyklus slučky. Prvý krok slučky sa číta z dát na DATA riadku 80 textový reťazec "10602" a zapisuje sa do B\$. Riadok 60 zapisuje do A\$ (1) hodnotu z B\$. Nasledujúci krok slučky, keď A = 2. Načíta sa B\$ opäť z dát na riadku 80, ale tentoraz to bude nasledovná hodnota "20605". Riadok 60 - bunka A\$(2) sa prepíše hodnotou z B\$. Keďže má cyklus 36 priebehov a program má v dátach 36 hodnôt, bude sa tento cyklický proces 36 krát opakovať, až do zaplnenia pola A\$ tridsiatimi siestimi hodnotami.

120-140 Tri REM návestia s komentárom k programu. 150 Tu začína druhá slučka, určujúca počet pokusov. FOR-NEXT slučky nemusia byť volané znakom A. Je dovolené aj ine písmeno abecedy, napr. FOR J = 1 TO 10 alebo FOR M = 1 TO 10. BEEP.02, -10 vytvára krátky zvuk (o BEEP sa diskutuje detailne v tejto knižke neskôr) a CLS vykona zmazanie obrazovky.

160 Premenná B sa vygeneruje ako náhodné číslo medzi 1 a 36. Toto číslo reprezentuje, či máte stlačiť číslo od 0 do 9 alebo písmeno abecedy. 170 Tu je jednoducho komplex príkazov, ktoré tlačia písmená alebo čísla z premennej A\$ (B,1) pola A\$ správnou lokalizáciou, čo sa deje pomocou PRINT AT druhé a tretie číslo v DATA návesti (ak druhé a tretie číslo sú 12, bude tlačeny 12. riadok zhora) a štvrté a piate číslo križom (x-ová os). Napríklad, ak štvrté a piate číslo v DATA návesti je 05 bude tlačeny 5 polôh naprieč (na x-ovej osi). Inými slovami každá hodnota z DATA návestia obsahuje číslo alebo písmeno ako prvý prvok a konečné štyri elementy sú lokalizácia na obrazovke. 2. a 3. určujú pozíciu na osi Y, 5. a 4. pozíciu naprieč na osi X.

180-210 Táto FOR-NEXT slučka čaká a sleduje čas pre stlačenie klávesy. Dĺžka zdržania začína číslom na konci riadku 180 a končí riadkom 180. Ak je klávesa stlačená, riadok 200 to zistí a prenáša akciu na riadok 260

220 Ak nestlačíte klávesu včas, počítač vytlačí "NESKORO !!!" 225 Použitím slučky FOR-NEXT vznikne krátke oneskorenie. Môžete použiť PAUSE alebo "prázdnu" slučku FOR-NEXT pre zdržanie.

230 Zasiela priebeh programu na riadok 150 do FOR-NEXT testu. 240 Nakoniec posielajú počítač na riadok 300 k vytlačeniu konečného skóre.

260 V prípade, že je stlačená rovnaká klávesa, aká je zobrazená na obrazovke, tak sa pričíta skóre a tlačí hlásenie "Dobre, vaše skóre je teraz 3", alebo akékoľvek aktuálne skóre.

265 Ak bude stlačená klávesa alebo to isté na obrazovke, tlačí sa "Je mi ľúto, máte smolu. Skuste znova.", a potom zasiela beh späť na riadok 170 pre opätovné skúšanie. Poznámka tu má jednoduchu čiariku a po uzavrených úvodzovkách nasledujú slová "Skuste znova". Vymažete čiariku a pozrite sa, ako sa zmení obrazovka.

270-280 Tu je cyklus na zdržanie dovtedy, než je vytlačené nasledujúce písmeno alebo číslo na obrazovke.

290 Zasiel späť chod programu na riadok 150. (Pokiaľ sa nevyčerpalo všetkých 10 pokusov.)

300 Zobrazí skóre. 310 Vypíše hlásenie gratulácie, ak bolo skóre väčšie ako 7.

Nájdete tu veľmi užitočný program, ktorý vám pomáha bližšie sa zoznámiť s klávesnicou.

Materiál v tejto kapitole je čiastočne zhustený, ale domnievam sa, že pri poptivom prečítaní a zavedení programov v pravú chvíľu vám dá nádejný úvod k programovaniu v BASICu Spectra. Tieto informácie sú kompletované v dvoch manuáloch, ktoré prišli s Vaším počítačom a pomôže vám zoznámiť sa s rýchlym a pekným programovaním. Zvyšok knihy bude slúžiť k osvojeniu zbehlosti informácií daných v tejto kapitole a k porozumeniu úvodného manuálu a mnoho iného materiálu, ktorý prisiel s počítačom.

2.0 E X P E R I M E N T O V A N I E S F A R B A M I

V mnohých materiáloch sa dočítate, že váš počítač dokáže zobrazit 3 rôznych farieb. Niektoré zdroje uvádzajú až 16 a iné 15 farieb. Vysvetlíme si, ktorá informácia je pravdivá.

Spectrum má 8 základných farieb: tmavomodrá, červená, magenta (červenofialová), zelená, kyanová (svetlomodrá) a žltá plus čierna a biela. Farby sú určené číslami medzi 0 a 7. Relácie medzi číslami a farbami sú:

- 0 čierna
- 1 tmavomodrá
- 2 červená
- 3 magenta (červenofialová)
- 4 zelená
- 5 kyanová (svetlomodrá)
- 6 žltá
- 7 biela

Ale pozor! Pre každú farbu môžeme vybrať jeden z jej dvoch odtieňov pomocou príkazu BRIGHT 1, prip. 0. Číže 16 farieb je teoreticky pravdivá informácia. Ale prakticky sa farby so zvýšeným jasom dajú odlišit od svojich vzorov tým okatejšie, čím sú svetlejšie. Preto je svetlá biela tak výrazne jasnejšia ako obyčajná biela, ktorá je niekedy považovaná aj za sivú. A naopak čierna je v oboch prípadoch čierna, a preto je správny aj údaj o palete 15 farieb.

Ak máme čiernobielu televíziu alebo priebeh farieb sledujeme na farebnej televízii, uvidíme tieto farby tlačené v nasledujúcom poradí alebo ako odtiene šedej a čiernej v ČB TV.

Skúste toto:

- 10 FOR X = 0 TO 7
- 20 PRINT INK X; "\$\$" (\$\$ - dva štvorce získané stlačením CAPS SHIFT kľúča a následne klávesou "9"(GRAPHICS mód) a potom klávesou "8")
- 30 NEXT X

Kód farieb teda odpovedá tomu, ako sú farby produkované televíziou. Farebný televízor používa farbu modrú, zelenú a svetločervenú. Všetky ostatné farby sú vyrobené miešaním týchto troch. Magenta je vyrobená zmiešaním modrej a červenej a jej kód je 3. Je to súčet dvoch farebných kódov.

Keď poprvý krát zapnete Váš počítač, vstupná obrazovka je biela a vy na ňu píšete čierne. Je jednoduché sčítaním meniť farby použitím príkazu BORDER (klúč B). Skúste toto:

- BORDER 1 (nasledované ENTER)

Tento príkaz mení lem obrazu na tmavomodrý. Môžete vybrať čísla nula až sedem po príkaze BORDER a po stlačení ENTER je okraj obrazu farebne menený. PAPER je farba vnútornej plochy a INK farba je farba textu alebo grafov, ktorých farebný charakter je daný ich typmi. Písanie príkazov PAPER a INK je trochu zložitejšie ako príkaz BORDER. Zmenu PAPER pre novú farbu prevedieme najskôr stlačením CAPS kľúča na ľavej strane klávesnice a

súčasne SYMBOL kľúča na pravej strane, nakoniec podržíme červený SYMBOL SHIFT a stlačíme C. Skúste si nasledujúce kombinácie.

- PAPER 3: CLS (CLS príkaz je "v" klávesa)
- PAPER 2: CLS
- PAPER 6: CLS

Vidíte, že na pravouholníkovej obrazovej stránke sa vymeni najskôr magenta potom červená a potom žltá. Ako asi pravdepodobne viete, môžete obdržať niektorú z ôsmich farieb prostredníctvom príkazu PAPER. Ak chcete dať farbu na celú stranu (obrazovku), musí nasledovať po príkaze PAPER, CLS príkaz (clear the screen), na čistenie obrazovky. Teraz sa tým hlbšie zaoberajte, pretože sa k tomu vrátíme neskôr.

Príkaz INK je práve tak jednoduchý. INK sa dostane obodným postupom ako pri PAPER, len miesto klávesy "C" zvolíme "X". Príkaz INK je použitý takým istým postupom ako PAPER. Pozrite sa na farebný efekt textu.

Skúste:

- 10 PAPER 7
- 20 INK 3
- 30 CLS
- 40 PRINT "testovanie"

Ak spustíte program, uvidíte, že počítač tlačí text v magente a biela je základom (pozadím). Môžete zadať akýkoľvek farebný text na akékoľvek farebné pozadie (PAPER). Môžete si teda voľit, ako vyberať farby INK a PAPER. PAPER farba je farba pozadia pracovnej plochy obrazovky a INK je farba grafiky a textu.

Je veľmi obtiažné čítať slová v istých farebných kombináciách. Červená a magenta sa napríklad nesmú miešať jedna s druhou.

Skúste nasledujúce:

- PAPER 7: CLS: PRINT INK 5; PAPER 1; "testovanie"

Ako môžete vidieť, práve priestor za slovami bol v svetlomodrej s textom v tmavomodrej.

Tu sú ďalšie štyri dôležité príkazy pre prácu s farbou a všetky nájdete dole na spodnej časti rady kľúčov na počítači. Sú to: FLASH, BRIGHT, OVER, INVERSE. Posledné dva nie sú určené k zmene farieb, ale majú potrebné použitie pre prácu s farbami.

BRIGHT

Príkaz BRIGHT je veľmi jednoduchý na porozumenie. Je použitý pre vytvorenie jasnejšieho alebo tmavšieho odtieňa farby toho alebo onoho textu (INK) alebo pozadia (PAPER).

Pozrite sa, ako pracuje tento podprogram:

- 10 FOR X = 0 TO 7
- 20 PRINT PAPER X; BRIGHT 0; " "; BRIGHT 1; " "

30 NEXT X

Táto tlač sa nachádza (mysli sa tým malý blok PAPER) v dvoch radoch. V prvom je normálna farba a v druhom je tá istá farba vytvorená jasnejšie. Ako vidíte, parameter príkazu BRIGHT je určený buď nulou alebo jedničkou.

Tento nasledujúci program vytvára tiež farebné kombinácie textu s rôznymi stupňami jasnosti pomocou slucky pre BRIGHT.

```
10 FOR X = 0 TO 7
20 FOR Y = 0 TO 1
30 PRINT INK X; BRIGHT Y; " ";
40 NEXT Y
50 NEXT X
```

FLASH

Ďalšie slovo je FLASH a je použité podobne ako BRIGHT. Je rovnako určené pomocou nuly a jedničky. FLASH totiž slúži na zvýraznenie textu alebo grafiky blikaním. FLASH 1 znamená zapnuté blikanie, FLASH 0 vypnuté. Ukážeme si kombináciu použitia FLASH a BRIGHT v jednom a tom istom programe.

```
10 FOR X = 0 TO 7
20 FOR Y = 0 TO 1
30 PRINT PAPER X; INK 9; BRIGHT Y; FLASH Y; " ";
40 NEXT Y; NEXT X
50 GO TO 10
```

Podprogram ukazuje, že FLASH spôsobuje blikanie medzi farbou INK a farbou PAPER. INK 9 je špeciálny parameter, vďaka ktorému Spectrum píše v ktoromkoľvek INK tak, aby bol v dobrom kontraste s PAPER farbou. Program práve ukazuje, že pre štyri tmavé farby PAPER 0 až 3 si Spectrum volí biely INK a pre svetlé farby PAPER 4 až 7 sa zvolí čierny INK.

To je dôvod prečo BRIGHTované tmavé farebné štvorce blikajú medzi farbou a bielou a svetlé farby blikajú medzi farbou a čiernou. Objasnime si zároveň aj funkciu INK 8, aj keď sa v programe nevyškytuje. Nie je totiž žiaden problém program jednoducho obmeniť a porovnať výsledok. INK 8 slúži na to, aby sme mohli písať na obrazovku tak, že nezmeníme pôvodnú farebnú informáciu. Vy si to sice vôbec neuvedomujete, ale pri písaní na obrazovku sa neprenáša do obrazu len tvarové vyjadrenie písmena ale súčasne sa zobrazí aj farebná informácia. Práve pri INK 8 to neplatí - prenáša sa iba tvar a farba zostáva pôvodná. Môžete použiť tieto dva parametre s príkazom PAPER a máte obdobný efekt. Takto sú teda príkazy PAPER 8 a PAPER 9 rovnocenné, ba možno ešte pádnejšie ako bežne zaužívané kombinácie. Môžete použiť číslo 8 po FLASH a BRIGHT ale nie 9. V každom prípade číslo 8 zaručí, že každý atribút stupnice bude podržaný a nový znak bude vytlačený bez novej farby.

Pre istotu si otestujte tento program a porovnajte ho s prípadom, keď vynecháte PAPER 8.

```
5 CLS
10 FOR F = 0 TO 7
```

```
20 PRINT PAPER F; " ";
30 NEXT F
40 PRINT AT 0,0; PAPER 8; "TEST NA FARBY";
```

Pre dokonalé pochopenie si vyskúšajte obmeniť program tak, že príkazy PAPER vymeníte za INK a pod.

INVERSE, OVER

Dve posledné slová, ktoré sa môžu použiť, keď je styk s farbou, sú INVERSE a OVER. INVERSE - zinvertuje, prehodí farby naopak, t.j. INK za PAPER a PAPER za INK. Toto má ten efekt, že biely zápis sa píše čierne a naopak. To isté platí pre všetky farby z INK na PAPER a týka sa i formy podobnej pre BRIGHT, a FLASH.

Ako v predošlých prípadoch, príkaz je charakterizovaný parametrom, ktorý nadobúda dve hodnoty: 1 - zinvertované farby, 0 - pôvodné farby. Taktiež možno použiť 8 na zachovanie pôvodnej hodnoty na mieste na obrazovke, kde sa bude písať.

```
10 FOR X = 0 TO 1
20 PRINT INVERSE X; INK 5; PAPER 1; "UKÁŽKA INVERSE"
30 PAUSE 50
40 NEXT X
```

OVER je veľmi užitočné slovo BASICu a referuje o spôsobe akým sa prekryje starý a nový nápis. Príkaz má 1 parameter, ktorý nadobúda hodnoty 0 a 1. Príkaz OVER 0 spôsobí, že ak čokoľvek budeme písať na obrazovku, tak sa tesne pred tým zmaže ten text alebo grafika, ktoré ležia bezprostredne na mieste kam má prísť nový text.

OVER 1 naproti tomu spôsobí nasledovnú vec: Zoberie sa nový text a prekryje sa so starým špeciálnym spôsobom. Predstavme si písmeno ako štvorec zložený zo 64 bodov zoradených do matice 8x8 bodov. (Podobne ako šachovnica.) Písmeno na obrazovke je pre počítač vlastne takýmto štvorcem, kde sú prázdne políčka pozadie a plné políčka tvoria obraz písmena. Napr. L si môžeme predstaviť ako prázdny štvorec s piatimi plnými bodmi nad sebou a šiestimi bodmi vedľa seba tvoriacimi pravy uhol. V zobrazení OVER 1 sa zoberie štvorec starého znaku a nového znaku a vytvorí sa nový štvorec podľa troch pravidiel.

1. Ak sú pod sebou ležiace príslušné políčka prázdne, výsledné políčko bude tiež prázdne.

2. Ak je práve jedno plné, bude plné aj výsledné políčko.

3. A nakoniec, ak sú obe plné, výsledné políčko bude prázdne.

Takto sa skontroluje všetkých 64 malých bodov a prinesie sa celý znak, písmeno, či číslica. Aj keď to teraz vyzereá komplikovane, nie je to žiadna španielska dedina. Stačí si totiž uvedomiť, že nový znak písaný na čistú plochu sa vôbec nezmení, viď pravidlo 2.

Ako pomôcku si zoberte tento príklad: Píšte dva znaky na obrazovku. Písmeno "O" a nula "0". Teraz si predstavte, čo príde, ak prevediete PRINT OVER s veľkým písmenom O a nato napíšete nulu. Predstavte si v duchu, čo očakávate. Aký vznikne obraz, keď dve čierne (plné) dajú bielu (prázdnu).

A teraz skúste aktuálny PRINT OVER:

```
PRINT OVER 1; "O"; CHR $ 8; "0"
```

Pozn.: CHR\$ 8 je kontrolný znak pre návrat späť o jeden znak, čo služi na to, aby sa 0 napísala tam, kde bolo 0. Bez CHR\$ 8 by totiž počítač písal znaky za sebou.

Skuste OVER i na trochu iných znakov a skúste predpovedať, čo sa stane.

Ako teda môžeme OVER PRINT využiť? Jedno dobre použitie je vytvorenie objektu, kedy sa tento objaví pred ostatnými alebo za inými (ostatnými). Jednoduchý príklad tohto je program OVERPRINT. Tu normalne písmeno "p" cestuje naprieč obrazovkou a dokumentuje opísaný jav. Normalne "p" prechádza nad iným momentálne sa vytvárajúcim čiernym stvorcom (spomente si, že čierna a biela dá bielu) a potom prechádza do inverzného "p".

2.1 P R E T L A Č

```
1 REM Pretlač (overprint)
5 PAPER 5: INK 1: CLS
10 PRINT INVERSE 1; AT 10, 10; "p"
15 FOR A = 0 TO 20
20 PRINT PAPER 1; INK 5; INVERSE 1; OVER 1; AT 10,A;"p"
30 PAUSE 10
40 PRINT INVERSE 1; OVER 1; AT 10,A; "p"
50 NEXT A
```

ATTR

Zmienili sme sa o prípade, kedy ste mali len INK farbu alebo PAPER farbu pre charakteristický stvorec. To je tým, že Spectrum skladuje informácie o farbe INK, PAPER, či je zapnutý FLASH alebo nie, či je BRIGHT alebo normálny stav pre každý znak obrazovky v pamäťovom priestore pre "atribúty" (lokalizácia 22528 do 23295). Toto dáva pamäťovú lokalizáciu, ktorá je 32 x 24, číslo stĺpcov je násobené počtom riadkov obrazovky počítača. Každá z týchto pamäťových lokácií obsahuje číslo medzi 0 a 255, ktoré referuje, aké farby pre daný znak. Toto číslo je zmenené, ak použijete termín zo SINCLAIR BASIC - ATTR (x,y), kde x a y sú súradnice (ako keď sa tlačí PRINT AT) vo štvorci riešeni. Tieto čísla sú ľahko zrozumiteľné pokiaľ už s nimi máte nejakú prax.

Tu je vytvorené zo štyroch súčasne sčítaných čísel. Tu sú:

```
128 alebo 0 závisí či stvorec bliká - FLASH
64 alebo 0 závisí či je BRIGHT alebo nie
8*x kde x je kód PAPER farby
a posledný kód je pre INK farbu
```

Použite ATTR nasledovne:

```
PRINT AT 10,10; FLASH 1; BRIGHT 1; INK 2;
PAPER 6; "S"; ATTR (10,10)
```

Tu píšete písmeno S a pozíciu radu 10, stĺpec 10 v červenom (INK) a v žltom podklade BRIGHT a FLASH dohromady. Čo myslíte, čo bude hlásiť počítač v ATTR v pozíci 10,10? Bude to 128 + 64 + 8 * 6 + 2 = 242. Toto

je jednoducho ľahká práca ako poznať atribúty. Ak práve zadávate odozvu na ATTR (x,y) a očakávate prácu podmienenú číslami, môžete použiť ATTR program.

Mohli by ste od premennej "a" mať rovnaké výsledky v ATTR (x,y) v programe a využiť tieto získané hodnoty pre INK, PAPER, BRIGHT a FLASH. Jedným z vhodných použití funkcie ATTR (x,y) v programe je detekcia farby znakov. Napríklad si môžete priať pohyb figurky naprieč pohorím. Vyznačíte pohorie farbou a zbytok obrazu diferencnou farbou. Testujete, či farebná informácia pre súradnice figurky zodpovedá farbe pohoria. Podľa testu rozhodujete, či presuniete figurku na novú pozíciu.

2.2 A T T R K A L K U L Á T O R

```
10 INPUT "ATTR ? "; A
20 IF A - 128 <= 0 THEN GO TO 50
30 LET A = A - 128: PRINT "FLASH je zapnutý. "
40 GO TO 60
50 PRINT " FLASH je vypnutý. "
60 IF A - 64 <= 0 THEN GO TO 90
70 LET A = A - 64: PRINT "BRIGHT je zapnutý. "
80 GO TO 100
90 PRINT "BRIGHT je vypnutý. "
100 IF A/8 = INT (A/8) THEN GO TO 140
120 PRINT "PAPER je";INT (A/8);" INK je"; A - ( 8 * INT(A/8))
130 STOP
140 PRINT "INK je 0, PAPER je "; A/8
```

V tomto prípade budú hory v obvykle známej polohe, hoci ich aranžujete pre pohybujúcu sa rovínu. V tomto prípade teda ATTR nie je veľkou veličinou. Ale v iných prípadoch nemusíte presne poznať kde sú 'prekážky', alebo môžete byť schopný označiť 'prekážky' prostredníctvom IF THEN príkazu. Pri týchto okolnostiach môže byť ATTR veľmi užitočný príkaz.

2.3 D Ů H A

Možno chcete s veľkým odhodlaním kresliť farebne. Nasledujúci program kreslí dúhu. Podávame poznámku, akými príkazmi sa kreslí oblúk a kruh a ako sa vyrobia krivky z farieb. Poznamenávame tiež, že oblúky a diferencná farba sú v priestore použité oddelene. Tu môže byť len jedna INK farba v každom charakteristickom priestore veľkosti znaku, takže ak kreslite dve farebne rozdielne čiary, súčasne ich uzamykáte a druhá bude "obsiahnutá" už pri prvej výmene pri INK hodnote v stupnici, t.j. v našom kóde.

```
1 REM Dúha
2 LET Y = 1
3 DIM A (6) : GO SUB 100
5 PAPER 7 : BORDER 0: CLS
10 FOR A = 160 TO 250 STEP 15
20 FOR U = 1 TO 5
30 INK A (Y): PLOT A + U,0: DRAW -A,A-80 + U,PI/2
40 NEXT U
50 LET Y = Y + 1
60 IF Y = 7 THEN GO TO 200
```

```

70 NEXT A
100 LET A(1) = 3: LET A(2) = 1: LET A(3) = 5: LET A(4) = 4:
LET A(5) = 6: LET A(6) = 2
110 RETURN
200 PRINT PAPER 7: AT 19,5: INK 3: "S"; INK 1: "P"; INK 5:
"E"; INK 4: "CT"; INK 6: "R"; INK 2: "UM"

```

2.4 1 2 8 F A R I E B

To, že je Spectrum schopné tlačiť dve farby v každom znakovom stvorci môže byť použité v prospech vytvorenia 128 farieb vašim Spectrum. Potrebujete najskôr definovať grafický charakter - znak, ktorý je veľmi podobný šachovnici (je to "A" klávesa na ZX 81).

Princíp spočíva v tom, že v znaku malej šachovničky budeme farebne kombinovať všetky možné farby pre PAPER so všetkými kombináciami INKU. Pozornejší čitateľ si už uvedomil, že 8 farieb podkladu PAPER skombinovaných s ôsmimi farbami INKU umožňuje 8*8 t.j. 64 kombinácií. Ak k tomu pridáme ešte BRIGHT tak máme 2*64 t.j. 128 rôznofarebných stvorcov veľkosti 1 znaku.

Ako príklad si predstavte kombináciu červenej a žltej. Spomínanou metódou vygenerujeme peknú oranžovú farbu. Ale teraz už spomínaný program, generujúci vs. možné kombinácie.

```

10 REM Farby
20 PAPER 7: BORDER 6: CLS
30 REM Šachovnica pomocou CHR$
40 FOR X = 0 TO 6 STEP 2
45 REM Písmeno P v nasledujúcom riadku musí byť písané v G-móde
50 POKE USR "p" + X,85
60 POKE USR "P" + X + 1,170
70 NEXT X
80 FOR P = 0 TO 7
90 FOR I = 0 TO 7
100 FOR B = 0 TO 1
110 PRINT PAPER P; INK I; BRIGHT B; "PP";
120 NEXT B: NEXT I: NEXT P
130 PAUSE 200
140 CLS: PRINT "a teraz plná obrazovka...": PAUSE 200
150 CLS
160 POKE 23692,100
170 FOR P = 0 TO 7
180 FOR I = 0 TO 7
190 FOR B = 0 TO 1
200 FOR K = 0 TO 31
210 PRINT PAPER P; INK I; BRIGHT B; "P"
220 REM "P" je znak P, písaný v G-móde
230 NEXT K
240 POKE 23692,100: REM Automatický scroll
250 NEXT B: NEXT I: NEXT P
260 STOP

```

Môžete touto odlišnou metódou kresliť priečne srafované línie, prípadne zamenou za DRAW a PLOT príkazy, kresliť línie, ktoré sú farebne zostavené z pixelov (bodov). Pri kreslení s INK a rozdielne farebným PAPER

vytvárame taký istý efekt ako keď sa použije 'znaková grafika' v G-mode. Pozn.: Pod pojmom G-mod rozumieme písanie v režime kurzora G, ktorý umožňuje písať hrubou stvorcovou grafikou na klávesách 1 až 8.

Spectrum môže produkovať veľmi jemný displej pri kreslení čiar pomocou príkazu DRAW s interakciou a dáva druh 'interferenčnej vzorky' program 'siet' je dobrá demostrácia na predchádzajúcu časť.

2.5 S I E Ť

Program kreslí na obrazovke centrálnu pozíciu a potom o jednu časovú jednotku kreslí (rýsuje) pre tento bod pozíciu na jej hrane. Ak budete kresliť vždy hranovú pozíciu, stretnete sa eventúelne so solidným množstvom farieb. Ale pri kreslení bodov (pixelov) sa na poslednom pixeli vyskytne veľmi zaujímavá vzorka. To je preto, že riadky Spectra nie sú kompletne kreslené priamociaro, ale kolíšu v pevnom smere v závislosti bod po bode ako sú kreslené. Program kreslí INK farby samostatne ako pravdepodobný rozdiel farieb PAPER a efekt závisí na miešaní obsiahnutých farieb. Niektoré farby sa zdajú ako interferenčné, niektoré viac než iné a niektoré vyzerajú ako pohyblivé, po obrazovke viiace znaky, zapríčinené vzorkou.

```

10 REM Siet
20 LET Z = 1
30 LET INK = INT (RND * 5) + 1
40 LET PAPER = INT (RND * 5) + 1
50 IF INK = PAPER THEN GO TO 30
55 PAPER PAPER: INK INK: CLS
60 FOR A = 1 TO 2
70 FOR X = 0 TO 254 STEP 2
80 PLOT 128,88: DRAW (-127 * Z) + (X * Z), Z * -87
90 NEXT X
100 FOR Y = 0 TO 175 STEP 2
110 PLOT 128,88: DRAW 127 * Z, Z * -87 + (Y * Z)
120 NEXT Y
130 LET Z = -Z
140 NEXT A: PAUSE 100: GO TO 20

```

Tento nasledujúci program je ešte atraktívnejší ako siet (NETWORK). Je známy ako polárna hviezda (snehová vločka). Je kreslená ako mnohoramenná hviezda v bielom a potom je použitá technika, ktorá pričíta nové dimenzie vášmu Spectrum.

S novou PAPER farbou na vrchu snehovej vločky: Akonáhle vyvoláte zmenu PAPER farby potrebuje prvý status farebného čísla (číslo farby) a potom použijete CLS príkaz. Avšak toto zničenie vzorky máte makresliť. Z tohoto dôvodu Spectrum v tomto programe mení retazce v priestore rovnakej dĺžky ako je rada na obrazovke (704 charakteristických pozícií) a pri výpise stringu je získaný skvelý efekt vo vzorke v duhových farbách.

2.6 Ľ A D O V Á H V I E Z D A

```

10 REM Ladová hviezda
20 INK 7: BORDER 3: PAPER 1: CLS
30 LET P$ = " "; LET T$ = " "

```

```

40 FOR X = 0 TO 702
50 LET P$ = P$ + I$
60 PRINT " "; : NEXT X
70 REM Zmena retazca v priestore obrazovky
80 CLS
90 LET FLAG = 0
100 FOR K = 20 TO 50 STEP 10
110 LET Y = 0: LET Z = PI
120 LET C = 100
130 LET E = (Z - Y) / C
140 FOR L = Y TO Z STEP E
150 LET S = K * COS(L*6)
160 LET H = S * SIN L
170 LET V = S * COS L
180 PLOT 128 + H, 88 + V
190 NEXT L
200 LET FLAG = FLAG + 1
210 IF FLAG = 101 THEN GO TO 230
220 GO TO 150
230 LET FLAG = 0: NEXT K
240 LET COL = INT (RND * 6) + 1
250 PRINT AT 0,0; PAPER COL; OVER 1; P$
260 PAUSE 100: GO TO 240

```

Môžete dosiahnuť tieto výsledky aj inou cestou? Viete akou? Samozrejmovú alternatívu je použiť POKE v ATTR priestore pamäte s kódom zmeny PAPER farby každej lokácie na monitore obrazovky. Atribúty (informácia o farbe znakov na obrazovke) priestoru začínajú na adrese 2328 a obrazovka s PAPER sú vymenené, okrem dvoch riadkov na dolnom okraji obrazovky, ktoré sú určené pre riadky nového vstupu programu a končia adresou 23231. Nasledujúci program teda mení farby na monitore obrazovky bez zmeny textu:

```

10 FOR X = 22528 TO 23231
20 POKE X, 32: NEXT X

```

Môžete vidieť, ktorými farbami je práve POKEované? Zapamätajte si, že PAPER farby sú uschované v 8-násobkoch kódu, tu je POKE na obrazovke s parametrom 32, t.j 32/8=4 je zelená farba pre PAPER. Nasledujúci program "nočná obloha" kreslí veľmi interesantne vyvaženy obraz na obrazovke, používajúc veľa farebných príkazov o ktorých sme pojednávali v tejto kapitole.

2.7 NOČNÁ OBL O H A

```

10 REM Nočná obloha
20 PAPER 0: INK 7: BORDER 0: CLS
30 LET A = INT (127 * RND)
40 LET B = INT (87 * RND)
50 LET Z = INT (8 * RND)
60 LET R = INT (RND * 5)
70 IF R < 2.5 THEN GO TO 140
80 INK Z
90 PLOT 127 - A, 87 + B

```

```

100 PLOT 127 + A, 87 - B
110 PLOT 127 - A, 87 - B
120 PLOT 127 + A, 87 + B
130 GO TO 30
140 PLOT OVER 1; 127 - A, 87 - B
150 PLOT OVER 1; 127 + A, 87 + B
160 PLOT OVER 1; 127 - A, 87 + B
170 PLOT OVER 1; 127 + A, 87 - B
180 GO TO 30

```

Nasledovný program kreslí farebné kruhy. Príkazy môžu byť použité v tvare: CIRCLE INK i; x,y,z, kde i je INK farba, x a y sú súradnice stredu kružnice a z je polomer. Náhodné kruhy je program, ktorý ako už napovedá meno, kreslí náhodne rozmiestnené kruhy v náhodne generovaných farbách.

2.8 N Á H O D N Ě K R U H Y

```

1 REM Náhodné kruhy
5 PAPER 0: CLS
10 LET X = (234 * RND) + 10
20 LET Y = (154 * RND) + 10
30 LET Z = (5 * RND) + 1
40 CIRCLE INK Z; X,Y,10
50 GO TO 10

```

2.9 G U L E Č N Í K O V Á G U Ľ A

'Gulečník' je program vytvárajúci a kresliaci kruhy v stúpajúcom polomere pre ten istý stred (použíje sa STEP.1 pri stúpaní polomeru). Vytvára zvláštny efekt oživených piatich častí, ktoré su v priestore gule, odtiaľ pochádza meno programu.

```

1 REM Gulečnicková guľa
5 INK 1: PAPER 6: CLS
10 LET X = 100: LET Y = X
20 FOR A = 0 TO 22 STEP.1
30 CIRCLE X,Y,A
40 NEXT A
50 LET X = 130:LET Y = 85
60 GO TO 20

```

Kreslí kruhy na počiatočný bod ostatných osí tak, ako to prikazuje program Náhodne kruhy predtým a môže tiež vyrábať mnohofarebné kruhy pretože dva kruhy sú tlačene súčasne a farba v druhej zmenenej časti je časť farby prvej. Program je použitý v programe 'kužel'.

2.10 K U Ź E Ľ

```

1 REM Kužel
5 BORDER 1: PAPER 7: CLS
10 INPUT "start x súradnice? "; X
20 INPUT "start y súradnice? "; Y

```



```

30 INPUT "výška? "; H
35 IF Y + H > 175 THEN GO TO 100
40 INPUT "dĺžka?"; L
45 IF X + L > 255 THEN GO TO 100
50 INPUT "farba?"; C
60 FOR P = 0 TO H
70 PLOT X,Y: DRAW INK C: L,P
80 NEXT P
90 STOP
100 CLS: PRINT "mimo rozsah" !! "skúste znova"
110 GO TO 10

```

3.0 EXPERIMENTOVANIE SO ZVUKOM

Ako už pravdepodobne viete, je veľmi ľahké ozvučiť Váš Spectrum. Použijete BEEP príkaz (nájdete ho vytlačený na Vašej klavesnici na klávese Z). Najskôr stlačte kláves CAPS SHIFT a SYMBOL SHIFT na pravom okraji dole súčasne a nakoniec podržte červený SHIF kľúč a stlačte Z klávesu. Aby ste vytvorili zvuk, tak to urobte podobne: BEEP 1,0 (nasleduje ENTER). Ak to skúsíte, začujete tón dlhý jednu sekundu s výškou stredného C. Prvé číslo referuje ako dlhý tón je očakávaný a druhé určuje výšku tónu. Číslo dĺžky môže byť približne od 0.00125 do 10 a je v sekundách. Čas kratší ako hlásenie chyby Out of range - mimo rozsah.

Druhé číslo môže mať rozsah -60 až +69. Každý krok (STEP) hlási medziton od stredného C (ktoré má číslo 0), nad alebo pod. Tak BEEP 1,1 bude zvuk tónu a medziton nad stredným C a pre druhé BEEP 1,-10 bude zvuk noty 10 pod stredným C. Nezabudnite vsunúť čiarku medzi obe čísla. Môžete mať akúsi predstavu o rozsahu frekvencie Spectra keď vytvoríte kolobeh nasledujúceho krátkeho programu.

```

10 FOR N = -60 TO 69
20 BEEP .2,N
30 NEXT N

```

Ako počujete, tóny sa rozprestierajú od série rýchlych ťukov až k vysokému trilkovaniu. Pre hudbu môžeme urobiť záver, že najužitočnejší rozsah je okolo stredného C okolo ± 20 na obe strany a preto pravdepodobne použil Sinclair označenie stredného C ako 0. No i tak vyššie alebo nižšie zvuky môžu poslúžiť práve tak dobre ako zvukové efekty.

AK potrebujete hlasný zvuk spojte váš počítač s vonkajším zosilovačom. Najprímeranejšie je ak spojíte zosilovač s dierkou MIC. Môžete dať váš signál do oboch zdierok EAR a MIC späť do Spectra, s EAR signál bude o niečo silnejší. Môžete tento pokus skúsiť, ale skontrolujte, či je všetko dobre pripojené k vášmu zosilovaču. Pokiaľ je pripojený externý zosilovač môže získať notáciu tým, že stlačíte každú klávesu až keď bude počuť ťuky. Tento postup so stláčaním kláves je užitočný, ale ťukot, ktorý počujete z vonkajšieho zosilovača je veľmi tichý, pre praktické použitie. Môžete to napraviť vstupom nasledujúceho príkazu: POKE 23609,100 (a potom stlačte ENTER). Teraz kedykoľvek stlačíte klávesu, ozve sa vysoké pipnutie. Toto je užitočné vtedy, keď rýchlejšie píšete a príležitostne sa pri tom divate na obrazovku. Až teraz môžete zosilniť zvuk pomocou vašeho zosilovača, pokiaľ ho chcete použiť.

Je praktické používať jeden alebo druhý, ak je Vaše Spectrum v stave

"ťukotu" a ak použijete zvuk príkazom BEEP. Môžete vypínať alebo zapínať káble i keď je to dosť hlasité. Ak použijete vonkajší zosilovač predsa len ho vypínajte, ak očakávate nahrávanie programu alebo prehranie na magnetofónovú pásku (v závislosti od toho, aký výstup použijete).

Vráťme sa späť k hudbe. Každé notové číslo v príkaze BEEP reprezentuje polotónový rast (alebo pokles pri zápornom čísle). To je 12 čísiel pre oktávu. Stredné C = 0 a tón č. 12 je tiež C, ale o oktávu vyššie.

Musíte písať hudbu v počítači usporiadaním tak, že meníte noty z notových liniek ako mriežkové linky, ktoré meníte na noty pomocou BEEP čísiel a každé trvanie noty je tak isto číslo (v takom usporiadaní, že jednoduchý úder trvá asi 0.25 sek.). Toto je tzv. časová spotreba. Musíte byť šikovný, aby ste prepísali melódiu i zo série písmen pre noty a čísla pre jednotlivé údery.

Program hudobník to robí práve tak. Volá pomocou zmenených písmen A a G ako nízku (nižšiu) oktávu, kde C je stred. C a písmeňá od a do c sú ďalšie vyššie oktávy. Zapišete melódiu ako jednoduché riadky, kde každá nota je doprevádzaná indikačným číslom s dĺžkou trvania. Volíte 1, čo reprezentuje 1 úder noty (1 krát dĺžka noty, 2 pre dvojnásobnú dĺžku atď.). To vám umožňuje výber rýchlosti ako je vidieť na riadku v programe: 5 INPUT "aká je rýchlosť? 1 je pomalá, 5 rýchla ";X
Môžete teraz použiť doplnenie riadku 370 kde je prevedený BEEP príkaz.
370 BEEP X(Z)/X, Y(Z).

A teraz vidíte použítie tohto programu, je omnoho ľahšie než keď máte písať hudbu podľa notovej osnovy.

3.1 HUDOBNÍK

```

10 REM Hudobník
20 DIM X(50): DIM Y(50)
30 LET K = 0: LET L = 1
40 BORDER 2: PAPER 4: INK 9: CLS
50 PRINT AT 0,10: INVERSE 1: "hudobník"
60 PRINT "píšte hudbu ako riadok písmen a čísiel"
70 PRINT "píšte noty ako písmena s dĺžkou uderov"
80 PRINT "máte 2 oktávy z ktorých nižšia je od A do G a
druhá od a do c."
90 INPUT N$
100 FOR A = 1 TO LEN N$ STEP 2
110 IF CODE N$(A) < 97 THEN GO TO 280
120 IF N$(A) = "a" THEN LET K = -0.1
130 IF N$(A) = "d" THEN LET K = .5
140 IF N$(A) = "e" THEN LET K = 1
150 IF N$(A) = "f" THEN LET K = 1
160 IF N$(A) = "g" THEN LET K = 1.5
170 LET Y(L) = (CODE N$(A)-87)+(2*K)
180 LET L = L + 1
190 LET K = 0
200 NEXT A
210 LET L = 1
220 FOR T = 2 TO LEN N$ STEP 2
230 LET X(L) = VAL N$(T)/2
240 LET L = L + 1

```

```

210 IF INKEY$ = "4" THEN BEEP X,-2 + K
220 IF INKEY$ = "6" THEN BEEP X,1 + K
230 IF INKEY$ = "7" THEN BEEP X,3 + K
240 IF INKEY$ = "9" THEN BEEP X,6 + K
250 IF INKEY$ = "0" THEN BEEP X,8 + K
260 IF INKEY$ = "Q" THEN BEEP X,-7 + K
270 IF INKEY$ = "W" THEN BEEP X,-5 + K
280 IF INKEY$ = "E" THEN BEEP X,-3 + K
290 IF INKEY$ = "R" THEN BEEP X,-1 + K
300 IF INKEY$ = "T" THEN BEEP X,0 + K
310 IF INKEY$ = "Y" THEN BEEP X,2 + K
320 IF INKEY$ = "U" THEN BEEP X,4 + K
330 IF INKEY$ = "I" THEN BEEP X,5 + K
340 IF INKEY$ = "O" THEN BEEP X,7 + K
350 IF INKEY$ = "P" THEN BEEP X,9 + K
360 GO TO 130
370 STOP

```

Program sa Vás pýta, ktorú oktávu očakávate a voľbou jedného z dvoch programov máme stredné "c" buď na C alebo T tlačítku, alebo je oktáva oproti prvej znížená. Avšak toto označenie končí pri prepise riadkov 140 až 350, musíme potom stlačiť inú klávesu pre vytvorenie rozdielnej notácie.

Konečne program sa Vás pýta, či si prajete "vibráto". Tu stlačenie nôt trvá od 0,003 sec do 0,3. Pokiaľ je tón s iným trvaním (zdržaním) je to vidieť na efekte. Ak zdržanie trvá ďalej, štyrikrát viac, t.j. okolo pol sekundy, vytvorí sa príliš pomalá hra a tón omnoho kratší než 3 stotiny sekundy. Je to vlastne cvakanie. Môžete používať tento spôsob ako pomôcku ku zmene tempa. Budete potrebovať zdržanie každej noty k výrobe variant "pomaly, klik, klik" (tuk, tuk). Skúste podobný program s príkazom GO SUB. Toto potrebujeme pre tlač nôt na základe trvania tónu a nasleduje potom pre viac než dve noty trvania o 1 a štvrt dlhšiu než prvá sada. Ale nesmiete zabúdať, že podprogram potrebuje trochu času k prevedeniu, takže celý program môže štartovať znateľne neskôr. Spectrum nemá len zlomkovité trvanie (zdržanie), ale tiež zlomkovité hodnoty nôt. To možno využiť, ak uznáte užitočné aplikovať na svojom Spectre varhany, alebo iný nástroj než klavír. Tu vidíte, ako sa to jednoducho robí, ak pricítate do riadkov:

```

355 IF INKEY$ = "K" THEN LET K = K + 0.05
356 IF INKEY$ = "L" THEN LET K = K - 0.05

```

Tieto noty budete hrať pri zavedení zlomku ostrejšim alebo plochším tónom. Ak stlačíte jeden z dvoch kláves K alebo L (K je ostrejší, L je plochší zvuk). Toto je potrebné pri výrobe orientálnej hudby, kde je 16 i viac nôt na jednej stupnici, použitých obyčajne u 12 tónovej stupnice.

INÉ EFEKTY

Môžete urobiť viac než sa zapodievať len klavirom. Môže byť vyrobená vizuálna displejová stupnica. Toto Spectrum čini veľmi schopným a definície nových charakteristik budú rozobrané v kapitole hier, ktoré ste 'skladateľ' tvorí základ pre ďalšie programy pomocou typov, ktoré ste našli v 'hudobníkovi' ako čísla, písmená a znaky, ale v tom istom čase môže noty zmeniť v chrochtanie, trylkovanie a aplikovať na obrazovku,

```

250 NEXT T
260 GO TO 360
270 STOP
280 IF NS (A) = "A" THEN LET K = -0.5
290 IF NS (A) = "B" THEN LET K = -0.5
300 IF NS (A) = "F" THEN LET K = 0.5
310 IF NS (A) = "G" THEN LET K = 0.5
320 LET Y(L) = (CODE NS(A)-67-K)*2
330 LET K = 0
340 LET L = L + 1
350 GO TO 200
360 FOR Z = 1 TO LEN NS/2
370 BEEP X(Z)/2, Y(Z)
380 NEXT Z

```

3.2 K L A V Í R

Ak chcete, zahrá Spectrum ako klavír. Nasledujúci program Vám to predvedie. Je použitý príkaz INKEY \$, ktorý číta klávesnicu a vy uvidíte ktorá klávesa je stlačená. Ak máme písať, je výber z troch oktáv notou "C", začínajúcou v každom prípade klávesou K. Obrazová verzia vášho "klavíra" je prezentovaná na monitore a pomáha vám spomenúť si ktorá klávesa patrí ku ktorej note. Môžete si predstaviť písmená Q-P, ako biele klavírové klávesy a čísla nad nimi ako poltóny. Spectrum sprevádza hru v jemnej a dobre temperovanej stupnici, pripomínajúcu klavír, ktorá zodpovedá realnej hre a dobre rezonuje (trochu plochšie) s hudbou vášho 'klavíru'.

```

10 REM Klavír
20 INK 7: BORDER 0: PAPER 2: CLS
30 PRINT AT 5,12: INVERSE 1: "Klavír"
40 PRINT AT 8, 2: "klávesnica klavíru - klávesy 'Q' - 'P'"
50 PRINT AT 12,8: "stlačenie 'Z' bude dávať vyššiu oktávu, 'X' dáva nižšiu oktávu"
60 PRINT AT 16,0: INVERSE 1: "stlačením 'V' dostaneme vibrato a 'M' vibrato vypne"
70 PRINT AT 19,0: INVERSE 1: "stlačením 'C' a 'T' dostaneme stredné C"
80 LET K = 0: LET X = 0.3
90 PAUSE 500
100 REM Viditeľný displej
110 CLS
120 PRINT: INVERSE 1: AT 10,4: "Q"; AT 10,6: "W"; AT 10,8: "E"; AT 10,10: "R"; AT 10,12: "M"; AT 10,14: "Y"; AT 10,16: "U"; AT 10,18: "T"; AT 10,20: "O"; AT 10,22: "P"
130 PRINT PAPER 0, AT 8,5: "2"; AT 8,7: "3"; AT 8,9: "4"; AT 8,13: "6"; AT 8,15: "7"; AT 8,19: "9"; AT 8,21: "0"
140 IF INKEY$ = "Z" THEN LET K = 12
150 IF INKEY$ = "X" THEN LET K = -1
160 IF INKEY$ = "C" THEN LET K = 0
170 IF INKEY$ = "V" THEN LET X = 0.01
180 IF INKEY$ = "M" THEN LET X = 0.3
190 IF INKEY$ = "2" THEN BEEP X,-6 + K
200 IF INKEY$ = "3" THEN BEEP X,-4 + K

```

tlacit presnou notovou osnovou. Môžete tento program zmenit hrajúc na Spectre ako na varhany alebo na klavír a tlačiť tieto noty na obrazovku.

3.3 SKLADATEL

```
10 REM Skladateľ
20 PAPER 6: INK 0: BORDER 7: CLS
30 LET X = 151
40 REM Kreslenie novej osnovy
50 FOR Y = 1 TO 5
60 PLOT 0,X: DRAW 225,0
70 LET X = X - 8
80 NEXT Y
90 FOR A = 0 TO 7
100 READ T
110 POKE USR "L" + A,T
120 NEXT A
130 FOR B = 0 TO 7
140 READ T
150 POKE USR "K" + B, T
160 NEXT B
170 PRINT OVER 1; AT 3,3; "L"; AT 4,3; "K"
180 BEEP 0.25,0
190 REM Koniec notovej osnovy
200 DATA 0,4,7,5,4,4,4,4
210 REM Spodok notovej osnovy vzoru
220 DATA 0, 60, 124, 124, 124, 56, 0, 0
230 STOP
```

SCÍTANIE ZVUKOVÝCH EFEKTOV

Pri skúmaní tejto problematiky máte pádny dôvod skúmať a študovať zvuky a efekty v počítačových hrách. Môže počítač pričítať zvuk do hier? Môžete vyrobiť zvuk použitím programu premeny ohňa, stepovacím krokom alebo vlečúci hluk? Odpoveď je oboje. Áno i nie. Môžete napr. vyrobiť akýsi vzdialujúci sa užitočný hluk, ktorý obohati Vase hry.

Premeny ohňa. Premeny ohňa 2 a Premeny č. 3 sú návodom k vytvoreniu vhodných zvukov pre hry, kde sa strieľa a 'prechádzka' je príkladom ako môžete pripojiť zvuky k pohybu. Poznámka: prvé graficke znaky začínajú na riadku 140, kde je L, ďalší je K

3.4 P R E M E N A O H Ň A

```
10 REM Premena ohňa
20 LET D = 0.0125
30 FOR X = 1 TO 1
40 FOR Y = 4 TO 16 STEP 2
50 BEEP D, Y
60 NEXT Y
70 NEXT X
```

```
10 REM Premena ohňa č.2
20 FOR X = -10 TO 0
30 BEEP 0.0125,X
40 NEXT X
50 FOR Y = 0 TO -5 STEP -1
60 BEEP 0.0125,Y
70 NEXT Y
```

```
10 REM Premeny ohňa č.3
20 FOR X = 5 TO 20 STEP 1.5
30 BEEP .008, X
40 NEXT X
50 FOR Y = 20 TO 5 STEP -1.5
60 BEEP .008,Y
70 NEXT Y
80 PAUSE 30
90 RUN
```

3.5 P R E C H Á D Z K A

```
10 REM Prechádzka
20 FOR A = 0 TO 7
30 READ X
40 POKE USR "L" + A,X
50 NEXT A
60 FOR A = 0 TO 7
70 READ X
80 POKE USR "K" + A, X
90 NEXT A
100 FOR T = 0 TO 31
110 PRINT AT 21,T; " ";
120 NEXT T
130 FOR C = 0 TO 31
140 PRINT AT 20,C; "K": PAUSE 3: PRINT AT 20,C; "L":
    PAUSE 3: PRINT AT 20,C; " "
150 BEEP 0.02,30:BEEP 0.02,40
160 NEXT C
170 GO TO 100
180 REM Data pre dve osoby
190 DATA 24,36,153,126,24,100,132,4,24,36,153,126,24,38,33,32
200 STOP
```

Iným použitím podprogramu je program, ktorý simuluje pád bomby. Tu sa vyrába frekvencia, ktorej nota klesá od vysokých tónov až o nejakých 20 poltónov nižšie. Stupeň pádu je najskôr rýchly, ale stále sa zvyšuje a toto dáva dojem urýchlenia bomby než tón uzavrie zem.

3.6 B O M B A

```
10 REM Bomba
20 FOR X = 69 TO -55 STEP -0.3
30 BEEP .05,X
40 NEXT X
```

```

50 FOR Y = 0 TO 20
60 BEEP .01,-10: BEEP .01,-50: BEEP .01,-60
70 NEXT Y

```

Vela elaborátov pre mikropočítače spomína nielen schopnosť hrať jednohlasné tóny, ale tiež spoločne tri hlasy s frekvenčnými variáciami a prípadne biely šum. Váš Spectrum, Didaktik, či Delta toto vytvoriť nemôže (na rozdiel od počítačov Commodore C64 alebo ZX Spectrum 128K,+2 a pod.), ale môže použiť niektoré triky miesto jednoduchých BEEP zvukov. Frekvenčná časová závislosť ukazuje, ako môžete vyrábať frekvenciu zakrýjúc jednotlivo druhý tónov.

V prvom programe, ktorý je schopný meniť kroky medzi dvoma notami a môžete nim meniť trvanie tónov. V tomto prípade sa ide zo stredného C s celými notami od štartu programu s krátkym trvaním a kratšie trvá ako sa približuje ku konečnej frekvencii.

3.7 T V A R O V A N I E F R E K V E N C I E A Č A S U

```

10 REM Frekvenčný časový priebeh
20 INK 7: BORDER 6: PAPER 1: CLS
30 LET DUR = .09
40 FOR F = 0 TO 2 STEP .5
50 BEEP DUR,F
60 DRAW 300 * DUR, 20 * F
70 LET DUR = DUR -.005
80 NEXT F
90 DRAW 0,-100: BEEP .5,2
100 GO TO 40

```

V druhom programe stúpa tón (alebo klesá) s danou frekvenciou s posledným krokom, ktorý nadefinujete na riadok 40. Trvanie poskakovania zvuku pre jeden tón v stúpajúcej hodnote alebo klesajúcej hodnote s úbytkom tónov v podielových hodnotách.

3.8 F R E K V E N Č N Ě T V A R O V A N I E

```

10 REM Frekvenčný priebeh
20 BORDER 4: PAPER 4: CLS
30 REM Voláte zmenu hodnôt F, FE a S
40 LET F = 0: LET FE = 3: LET S = 1
50 FOR X = F TO FE STEP S
60 DRAW 25 * S, 20 * X
70 BEEP .03,X
80 NEXT X
90 DRAW 20,0: BEEP .2,FE
100 FOR Y = FE TO F STEP -S
110 DRAW 25 * S, -20 * Y
120 BEEP .03, Y
130 NEXT Y
140 GO TO 140

```

Môžete si dopriať pokus s nasledujúcim vývojom skladania niekoľkých zvukov súčasne do väčšieho komplexu zvukov. Ak však chcete previesť jemne

odstúpané na Vašom Spectre, musíte (doporučuje sa) použiť externý zosilovač.

Konečný program v tejto kapitole je "muzigraf" so spoločným programom zvuku a farby na rovnakých riadkoch. Zvuková notácia a rozsah tónov a zdržania je pevne limitovaná. Výška not odpovedá výške na obrazovke kvapiek, ktoré sú na nej tlačene. Nižšie noty sú neskor tlačene vpravo ako kvapky (škrvny). Výsledkom je koláž farieb ktoré vidíte a odpovedajú zvukom, ktoré počujete.

3.9 M U Z I K O G R A F

```

10 REM Muzikograf
20 BORDER 0: PAPER 0: CLS
30 FOR X = 0 TO 31
40 LET NOTE = RND * 20
50 LET DUR = RND * 0.4
60 LET INK = (RND * 6) + 1
70 PRINT INK INK; AT 20 - NOTE, X; "*"
80 BEEP DUR, NOTE
90 NEXT X
100 CLS: GO TO 30

```

4.0 P O U Ž I T I E V O B C H O D E

Ako všetky počítače, samotné Spectrum nemôže nič. S primeraným programom a s nesmiernou variantou úloh môže obchodovať. Nemôžete chcieť regulárne nákupy s programami s mnohými vlastnosťami, ale základnú činnosť tu bude zaistená. Postupným zdokonaľovaním sa uroveň približí úrovni požadovaných vlastností a pokiaľ tomu nebudete ďalej rozumieť ako to chodi, neobidiete sa bez podrobnosti.

Niekoľko príkladov. Len krátko: Tak ako je zdôraznené v prvej kapitole, počítače môžu urobiť čokoľvek, ak sa redukuje pokynmi na radu elementárnych príkazov. Tieto úlohy musia byť exaktne v súlade s informáciami, ktoré sú k tomuto davané. Napríklad ak váš fakturačný program obsahuje sekciu, ktorá prináša upozornenie (pripomienku) o platení v tomto mesiaci, veľa ľudí bude dostávať upomienky, že majú platiť 0.00 Kčs. Tri stupne starostlivosti požadované pri výrobe a použití počítačových programov závisia na aplikácii. Snaha požiadaviek zaistenia absolútnej bezpečnosti pre Space Shuttle ide tak ďaleko, že sa napríklad zruší i najlepší program, ktorý nie je do detailov preskúšaný a overený. Počítačové aplikácie môžu byť zoskupené pod číslom (nie jednoznačne definovaným) klasifikácie:

OBCHOD A VÝVOJOVÉ DÁTA

Obyčajne vzdialenejšie operácie sú dopravené relatívne neskôr ako dáta. Napr. inventárna kontrola hľadá žiadané tažšie ako napr. sučty alebo odpocťy, ale je dôležité nájsť tieto dáta tak rýchle ako to je možné.

System bude zahrňovať moduly, ktoré môžu spolupôsobiť s inými modulmi. Tak predaj bude pripad zmeny vo Vašej hlavnej učetnej knihe v inventári, a účtovná poznamka bude vytvorená dodatočne. Potom sa ukáže výhodou, ak máte slovný procesorový program prepojený s Vaším obchodným

programom. V každom prípade to bude pravdepodobne blaznivý pokus. Vlastne ľudský chybný činiteľ je vylúčený a preskúšanie je možné (napr. program bude vyžadovať dozor nad vstupmi pri vedomí primeranosti ukonov).
Dobry program je nakladný ako výrobne, tak udrzovaním. Balik obchodných programov pre Spectrum, ktorý je dosaziteľný na trhu vo Veľkej Británii, je veľmi zaludný plánovací projekt, ktorý môže byť uplatňovaný len v podobných, príbuzných aplikáciách, ale predsa ale menej než napr. Spectrum 16 Kb. Kto rozmýšľa perspektívne, vynaloží viac peňazi na vývojový balik programov. Softwareový vývoj si môže vyžiadať niekoľko dní práce a to vysoko kvalifikovanou osobou.

SLOVNÝ PROCESOR

Po zapojení klávesnice Spectra je tento spôsobilý a použiteľný ako slovný procesor. Slovný procesor vám umožňuje písať text a sledovať ho na obrazovke v hrubej forme. Môžete upravovať, vkladať, mazať alebo meniť text celkom podľa svojho želania. Ak je zavedenie slov správne, môžete tlačiť Váš text uhladne formátovaný. Potom sa môžete vrátiť späť a previesť ďalšie zmeny.

VEDECKÉ ALEBO ČÍSELNÉ VÝPOČTY

Výpočty môžu byť veľmi komplexné. Môže byť požadovaný relatívne malý vstup alebo výstup alebo pamäť, ale veľa času CPU (centrálnej riadiacej jednotky) môže spotrebovať časovo náročný (reálny) výpočet.

TECHNICKÁ APLIKÁCIA

Spectrum môže byť nápomocné pre prácu ľudí ako sú architekti, plánovači a inžinieri. Elementárne programovanie môže byť užitočné aj pre tieto aplikácie, pokiaľ nenastanú primárne alebo neštandardné výpočtové aplikácie. Štandardný technický software však existuje a často je veľmi hodnotný. (Tento software je však dostupný len na britskom trhu, výnimocne sa objaví zmienka aj v domácej inzercii.)

KOMUNIKÁCIA

Počítač môže byť použitý pre komunikáciu s rôznym zariadením vrátane iných počítačov. Takýmto externým zariadením môže byť normálna telefónna linka. Vzniká tak veľa zaujímavých možností a aplikácii. Spectrum je dobre prispôbivý k malým obchodným aplikáciám, pretože je to veľkokapacitný stroj s veľmi priemernou cenou. Ak má počítač slúžiť pre použitie v obchode, potom je tu zrejme jedna vlastnosť reálnej výhody. Práve tak možno povedať, kde možno použiť písací stroj, možno použiť i slovný procesor.

Spectrum môže pracovať tiež tak trochu i časový kontrolór, a uvidíte ako sa táto schopnosť uplatňuje pri vedeckej práci. Všeobecne doporučujeme a celkom správne, že sa musíte pozrieť najskôr na software, ak kupujete počítač. Mnohí ľudia sú pri kupovaní počítačov zaujatí špecifikáciami a podmienkami pre vedeckú prácu, že software, ktorý je prístupný považujú ako druhoradý pre svoju nakladnosť. Tento postoj bude snáď modifikovaný

pri pohľade na to, aký rozsah tovaru Spectrum predstavuje. Čokoľvek potrebujete z oblasti software je buď teraz dosiahnuteľné, alebo pravdepodobne dosiahnuteľné v priebehu času. Spectrum - obchod smeruje k tomu, aby aj s lacnou programovou výbavou bol zrovnateľný s podobným software drahších strojov.

Viacmenej, po predvedení a po zakúpení, je výmena vzácnym javom. Keď vyberáte software pre Spectrum, majte vždy na mysli, že kvalita balenia a inzercia nebude vždy v korelácii s kvalitami software. Doporučujeme uvedomiť si, že existuje rozsiahla časopisecká kritika, ktorá je najväčšou hybnou silou informácii a prináša presne informácie o kvalite a užitočnosti toho ktorého programu.

Rovnako sa nerozhodnite kupovať obchodný software v niektorom okamžiku, ale vtedy, ak ho nutne potrebujete použiť pre počítač, aby Vám pomohol napísať programy alebo adaptovať vlastné.

Základom tejto kapitoly je päť programov, ktoré Vám po menších úpravách pre Vaše konkrétne podmienky umožnia a uľahčia agendu v obchodníctve.

OSOBNÉ KONTA

Program, ktorý dovoľuje špecifikovať nejaké hodnoty obeživových cien a vhodný program - obsahuje menu, ktorá vám dovoľuje medzivolbu, ktorou uchovávate hodnoty na páske.

DIÁR

Program, ktorý zoraduje vstupy do datového poradovníka a tlačí na požiadanie na tlačiarňu.

TELEFÓNNA ÚSTREDŇA

- ktorá prijíma mená a telefónne čísla (v prípustnom tvare), slúži rýchlo ako databanka pre požadované mená a môže ich datovať a vyvolať v ktorukoľvek dobu.

DATABÁZA

Táto skladuje 130, 24 charakteristik dvojmo (na 16 K počítači a oveľa viac môže byť zmien na 48 K pristrojii), metódou, ktorá môže byť sprístupnená jedným alebo dvomi kritériami. Program je zasobníkom a kolekciou 130 záznamov a môže tlačiť a listovať záznamy v abecednom poriadku napr. mená alebo titulky a dovoľuje poslužiť prostredníctvom listovania. Bude užitočné adaptovať len pre obchod alebo iné zariadenia a ak je raz či viackrát kartotéka menená a niektoré charakteristiky nie sú pre kartotéku nutné, môžeme ich viackrát vymeniť pokiaľ zachováme poriadok a ľahkosť obsluhy.

FINANČNÝ MODEL ODBYTU

Jeden z najpopulárnejších balíkov obchodného softwaru vo svete je

VISICALC. Rozširuje siet kalkulátorov, ktorá dovoľuje použiť prúdové tvary a potom vytvára projekcie zložených z týchto tvarov. Tento program vykonáva dve z úloh Visicalcu: analyzuje model odbytového tvaru od mesiaca k mesiacu (alebo ktorokolivek období, pre ktorú ste sa rozhodli), a potom tieto tvary projektuje, s projekciou založenou na priemernom odbyte za mesiac pre periódu, ktorú poznáte, alebo výsledok odbytu za posledný známy mesiac.

Tento program môže mať veľmi dobré modifikácie pre použitie v malom obchode alebo doma. Hlavnú časť zahrnujú obsah tejto kapitoly oddeľuje pre Vás bezprostredne použiteľnú, je to prehľad ciest obchodného softwaru, ktoré môžeme napísať. Obchodné software zriedkavejšie obsahujú "chyttré" kódovanie pre zabudovanie programov hier. Bliži sa viac k metodike. Úloha, ktorá je dokonalá v obchodnom software môže vám obvykle pri jasnom rozpracovaní, pokiaľ ste začiatočníci, pomôcť pri vývoji a napísaní vašich hier.

Čoskoro zistíte, že je to relatívne jednoduché, pokiaľ ste to už skúsili v obchodných programoch. T.j. treba prepísať a modifikovať program alebo si ho sám napísať od začiatku.

Ako sa teraz dívate na jednotlivé programy, potom vidíte, ako sa ich môžete učiť a písať prostredníctvom obchodného aplikačného software.

Keď si prvýkrát prejdete menu budete čítať nasledujúce:

finančná rovnováha je Kčs 0

Vstup

- 1 - štart pre zrušenie
- 2 - modifikácia splátkovej (platobnej) tabuľky
- 3 - uschovanie peňazi
- 4 - nahranie na pásku
- 5 - stop

Je to dobrý nápad obstarat menu pre zabezpečenie programu tak, že tento program môže byť použitý bez znalosti špeciálnych inštrukcii alebo bez znalosti programovania a môže byť použitý neskôr.

Bude správne vyvolať ako prvý v tomto programe vstup 1. štart pre zrušenie. Keď to urobíte uvidíte: "vstup čísel členov, ktoré musíme platiť každý mesiac". Tu vstupujú hodnoty obeživa, ktoré máme, vaša hypotéka, platenie auta, najomný výnos zo zostatku, váš kreditný úver alebo iný stály prínos. Je lepšie mať nadbytok číselných kategórii než spustíme prvý raz program, hoci je to zbytočne počítať tieto položky, ak máme štartovať použitím "balika". Ak budete vstupovať pomocou 4 ako vstupom obeživa (tak vidíte, napr. vstup nasledovaný programovým listingom), potom budete hľadať vstup položiek jednej po druhej (tiež mien, a potom ich mesačnú cenu. Mena (nazvy) a ich ceny môžu byť užitočné pre ich neskorsiu výmenu. Pokiaľ máte vstup pomocou listingu (vstup "čistý" pre šetrenie čísel, ak ich medzitým potrebujete veľa), budete môcť zvolit jednu z modifikácii, ktorú si prajete: "ak je to správne stlačiť ENTER. Ak je chyba vstup s číslom až po výmene, ktorú si prajete".

Ak program nedáva vašu bankovú rozvahú, ak neštartuje prvýkrát, budete informovaný: "Vstup poslednej známej rozvahy (rozpočtu)", a potom "vstup depozita vrátane platu, "Vstup na terminál vstupom "E"". Ak

vytvoríte vstup depozita, tvar rozpočtu uvidíte v ľavom dolnom rohu obrazovky s automatickým rastom. Pokiaľ pricitate k vašemu depozitu (vkladu) a signalizujete ich Spectru pomocou "E" dostavate ďalší výber (volbu): "Vstup 1 je výťah vášho úveru alebo 2 pre menu". Ak zadáte vstup 1 prevedie úplný výťah. Bude tlačit pred nie rozvahú a neskôr mesačnú hodnotu (cenu), ktorá bude vytlačena z výpočtov.

Vrátate sa späť k štartovaniu menu, v tej istej dobe "Tvar rozpočtu je ...", nie je (naštastie) nula a ihneď nastartuje program. Všetky premenné sú zachované v programe a uchované v kazete ak ju nahram. V každom prípade, ak štartujeme v ďalšom case program pomocou RUN alebo CLEAR bude obsah zabudnutý. Posledný riadok programu (GO TO 4430) zabezpečuje to, že po prvej dobe použitý program bude štartovaný automaticky, dodržiujúc použitý rozpočet. Poznámka - ak uvádzate program v chode použite INK a FLASH, aby tá časť programu bola uvedená do popredia. Farba a iné použité grafické efekty, sú vždy užitočné pre skocenie do programu pri pravdepodobnej operačnej chybe. A tak napr. ak je vstup "chytaná chyba", tak nemusíme pre chybu rozrušiť celý program. Napr. riadok 325 je odmietnutý a nula vstúpi do riadku 320. Vstup nuly (práve stlačením ENTER bez predchádzajúceho čísla) spôsobí rozpad na riadku 340. Podobne, hoci program hľadá (v riadku 310) pre "E" aby vstúpil terminované do vstupu depozita, program bude tiež akceptovať "e" ako inštrukciu hoci toto nie je inštrukcia, ktorú program zabezpečuje, čo má previesť CAPS LOCK.

4.1 O S O B N É K O N T O

```
10 REM Osobné konto
12 LET ROZPOČET = 0
15 GO TO 430
20 PRINT "Zadajte počet hodnôt, ktoré musia byť platené mesačne."
30 INPUT ČÍSLO:CLS
40 DIM A$ (ČÍSLO,12)
50 DIM A (ČÍSLO)
60 FOR A = 1 TO ČÍSLO
70 PRINT AT A - 1,0; "vstup čísel"; FLASH 1;A
80 INPUT A$ (A)
90 PRINT AT A-1,0; "koľko je toho"; A$ (A), "každý mesiac?/v Kčs"
100 INPUT A (A)
110 PRINT AT A-1,0; TAB 3; A$ (A); TAB 15; "Kčs";A(A); " "
120 NEXT A
130 PRINT "Ak je to správne, stlačte ",FLASH 1; "ENTER"; FLASH 0;
", ak je tu chyba, urobte zmenu stlačením čísla."
140 INPUT B$
150 IF B$ = "" THEN GO TO 260
160 LET B = VAL B$
170 PRINT "Vstup ďalšieho mena pre hodnotu: "; B
180 INPUT A$ (B)
190 PRINT "A koľko je"; A$ (B), "každý mesiac? (v Kčs): "
200 INPUT A (B)
210 CLS
220 FOR A = 1 TO ČÍSLO
230 PRINT A; TAB 3; A$ (A); TAB 15;"Kčs";A (A)
240 NEXT A
250 CLS
```

```

270 PRINT "Vstup poslednej znamej rozvahy"
280 INPUT ROZVAHA
290 CLS
300 PRINT AT 0,0; "Kčs"; ROZVAHA
310 PRINT "Vstup depozita, vratane", "platu, postupne za
sebou", "terminovane vstupy"; FLASH 1; "E"
320 INPUT Q$
325 IF Q$ = "" THEN GO TO 320
330 IF Q$ = "E" OR Q$ = "e" THEN GO TO 360
340 LET ROZVAHA = ROZVAHA + VAL Q$
350 GO TO 300
360 INPUT "Vstup 1 ako výťah všetkého, 'kreditný úver',
alebo 2 = menu "; C
365 IF C = 2 THEN GO TO 450
370 LET SPEND = 0: FOR A = 1 TO ČÍSLO
380 LET SPEND = SPEND + A (A)
390 NEXT A
400 CLS
410 PRINT "Rozvaha pred hodnotou", "stojí Kčs"; ROZVAHA
420 LET ROZVAHA = ROZVAHA - SPEND
430 PRINT "Hodnota rozvahy je Kčs"; ROZVAHA
440 PAUSE 200
450 PRINT "Vstup"
460 PRINT "1 - štart pre zrušenie"
470 PRINT "2 - modif. splátkovej tabuľky"
480 PRINT "3 - uschovanie peňazi"
485 PRINT "4 - nahranie na pásku"
490 PRINT "5 - stop"
500 INPUT C: CLS
510 IF C = 1 THEN LET ROZVAHA = 0
520 IF C = 2 THEN GO TO 210
530 IF C = 3 THEN GO TO 300
540 IF C = 4 THEN SAVE "konto"
550 IF C = 5 THEN STOP
560 GO TO 430

```

Príklad chodu "osobné konto"

```

Rozvaha je 1342.75 Kčs
1 - štart pre zrušenie
2 - modifikácia splátkovej tabuľky
3 - uschovanie peňazi
4 - nahranie na pásku
5 - stop

1 kreditná karta      123 Kčs
2 úmor                245 Kčs
3 auto                54.64 Kčs
4 doprava 17 Kčs

```

Ak je to dobré, stlač ENTER. Ak je tu chyba, vstup čísla k výmene.

```

1342.75 Kčs
Vstup depozita, vrátane platu jeden za druhým
Terminovanie vstupu pomocou vstupu E

```

Rozvaha (rozpočet) pred hotovosťou 1342.75 Kčs
Rozvaha rozpočtu je 903.11 Kčs
ENTER
1 - štart pre zrušenie

4.2 D I Á R

Diarový program môže byť použitý ako napr. kartotečný systém, hoci jeho úlohou je v každom okamžiku dňa byť modifikovaný prácou, zmenou v denných hodinách tak, ako to vyžaduje potreba.

Je to trochu jednoduchší program než ostatné v tejto kapitole, pretože je úloha vo svojom obryse triviálna:

- akceptuje dáta a udalosti (prípady)
- triedi udalosti (prípady) v dátovom poriadku
- tlačí ich, ak je to požadovane

Pokiaľ program nie je jednoduchý, budeme povinný dať sa cestou písania dát. Hoci bude program akceptovať len dáta v jednom tvare ako čísla dni, čísla mesiacov, a rokov a okamžitý dotaz ukáže, že počítač ignoruje prerušenie (/) a triedi dáta do troch skupín napr. 12. december bude pred 1. decembrom 1999 (12121999 je väčšie než 01121999).

Preto teda sa musí s programom zručne manipulovať i s dátami prichádzajúcimi ako "malé čísla", ale môžu byť vytlačené v prijateľnom tvare vhodné pre posúdenie. Ďalším problémom je, že deň môže mať jednu alebo dve cifry, rovnako tak mesiac. Počítač musí doviesť pričítaním dát nulu k jednej cifre mesiacov alebo dni (avšak musí poznať, kedy je hľadaná nula nutná).

Podprogram pre toto leží medzi riadkami 40 až 80. Riadok 40 Vám hľadá vstup dát, dáva napríklad požadovaný tvar. Riadok 50 zapína iný retazec (AS) rovnajúci sa dátú, takže môže byť jednoducho vytlačena na niekoľko časových riadkov. Riadok 60 zabráni uvidieť druhý prvok dátového retazca pri roztrhnutí (ktoré bude len ak je mesiac jednoduché číslo), a keď je známe, že je deň jednoduché číslo, pričíta nulu na jeho začiatok. Riadok 70 zabráňuje vidieť, že je piaty prvok dátového retazca roztrhnutý (môže to byť, ak je mesiac jedno číslo) a pričíta nulu pred číslo mesiaca. Potom v riadku 80 sú dáta tak usporiadané, že rok prichádza ako prvý, nasledovaný mesiacom a potom dňom). Ak čítate toto v Amerike, mimochodom máte to šťastie, že len prehodíte 4 TO 5 s 1 TO 2, faktom zostáva, že v USA mesiac vystupuje pred dňom, ale ak sa píše dátum ako celé číslo, musíte vymeniť v riadku 40 číslo 25/12/84 za číslo 12/25/84.

Riadok 90 tlačí dátum v jeho originálnom tvare (teda kedy sa B\$ rovná vstupujúcemu dátú) v pravom hornom rohu. Hľadáte práve vstup "prípadu", kedy si prajete záznam až po dáte. Tak ako v programe "osobné konto", máte teraz možnosť (spôsobilosť) akceptovať vstup v danom tvare alebo vstup zamietnuť.

Môžete si domysliť, ak to máte akceptovať. Ako program vytvára prvky D\$ priestora (usporiadanie), ktorý je vo Vašom diári tak, že sa rovná dátum v počítačovom usporiadaní.

Podprogram od riadku 220 triedi obsah diára, tlačí potom do tvaru, ktorému budete rozumiť, a od riadku 330 vracia program dátovanie späť do potvrdeného tvaru (uznaného tvaru).

Ďalej je spúšťanie krátkeho programového príkladu. A ako cvičenie môžete očakávať menu - program ako je "osobné účty" - dovoľuje nahrat diar alebo ho doplniť. Rovnako tak je prezentovaná forma, ak nahráte diar na kazetu a potom ju štartujete pomocou GO TO 40 (radšej ako s RUN), podržíte si predchádzajúci dátový obsah a bude tak schopný automaticky pričítať k pričítanému a triediť v dátovom poriadku.

```

10 REM Diar
20 DIM D$ (200, 32)
30 FOR D = 1 TO 200
40 INPUT "vlož dátum (vzor 25/12/90) "; B$
50 LET A$ = B$
60 IF A$(2) = "/" THEN LET A$ = "0" + A$
70 IF A$(5) = "/" THEN LET A$ = A$(TO 3) + "0" + A$ (4 TO)
80 LET A$ = A$ (7 TO 8) + A$ (4 TO 5) + A$ (1 TO 2)
90 PRINT AT 0,0; "Datum: "; B$
100 PRINT "vlož poznámku, nie dlhšiu ako 22 znakov"
110 INPUT C$
120 CLS
130 PRINT AT 0,0; "-"; C$
140 PRINT "ak je to dobre, stlač "; INVERSE 1;
"ENTER"; INVERSE 0, " Oprava, stlač ";
FLASH 1; " E "; FLASH 0, " potom ENTER"
150 INPUT E$: CLS
160 IF E$ <> "" THEN GO TO 40
170 LET D$(D) = A$ + C$
180 PRINT "stlač "; INVERSE 1; "ENTER", INVERSE 0, " pre vstup
nasledovného údaju. Vlož Z a Enter, ak chceš zoradený zoznam"
190 INPUT E$: CLS
200 IF E$ = "" THEN NEXT D
210 PRINT PAPER 2; FLASH 1; "Chvilu strpenia, prosím": PAUSE 100:CLS
220 LET B = 0
230 LET G = D
240 LET Z = 1
250 LET B = Z + 1
260 IF B > G THEN GO TO 330
270 IF D$(B) > D$(Z) THEN GO 290
280 LET Z = Z + 1:GO TO 250
290 LET Q$ = D$(Z)
300 LET D$(Z) = D$(B)
310 LET D$(B) = Q$
320 GO TO 280
330 PRINT D$(G) (5 TO 6); "/" ; D$(G) (3 TO 4); "/" ; D$(G)
(1 TO 2); "-" ; D$(G) (7 TO)
340 LET G = G - 1
350 IF G > 0 THEN GO TO 240

```

Príklad chodu "diar"

02/02/84 - Zubár
03/04/84 - kupa nového plásta
07/06/84 - doktor

4.3 TELEFÓNNA ÚSTREDŇA

Ste už schopný vidieť závislosť medzi "diarom", "telefónnou ústredňou" a nasledujúcim programom "databáza", ktorý nasleduje. Všetky tri programy súhrne akceptujú informácie používania - triedia niektorú cestu (ale spôsobom ako u databázy) a potom jedna po druhej tlačia v poradi v akom boli položky zoradené. Štruktúra týchto troch programov dá kľúč k tomu, ako konštruovať triedenie - doplnkový program, ktorý slúži tak ako to potrebujete. Menu "telefónnej ústredne" po riadkoch:

- 1 - štart novej ústredne
- 2 - pričítanie nových mien
- 3 - hľadanie (nových) čísel
- 4 - ukladanie "telefónnej ústredne"
- 5 - tlač "telefónnej ústredne"
- 6 - stop

V tejto danej forme, ktorá bude akceptovaná 200 vstupmi o 32 znakoch. Možno máte vhodné, kresťanské jednoduché meno, môžete tak jednoducho zväčšovať číslo vstupu zmenou riadkov 20 až DIM D\$(400,16). Váš vstup mien a potom čísel a máte šancu súhlasu pokiaľ sú pričítané do telefónnej ústredne.

Ak si prajete vstup nového mena, potom stlačte ENTER a vrátite sa na riadok 40 pre ďalší prvok do ústredne.

Ak si prajete triediť ústredňu v nejakom poradí, stlačte nejakú klávesu (okrem BREAK) a potom stlačte ENTER. Ústredňa je triedená v abecednom poradí založená na prvom liste vstupujúcich mien (ak bude potrebovať rodné mená ako prvé, očakávajte, že ústredňa bude triediť rodné mená). Žnovu sa vrátite k menu. Ak je vstup 3 "hľadanie čísel" program, ide na riadok 500 a tlačí "vstup mien je potvrdený". Program hľadá prostredníctvom ústredne (v prekvapujúco krátkej dobe) a ďalší vstup čísel, ak ich hľadáme, alebo hlási "meno nemožno nájsť".

Tento program tiež ako "osobné konto" ide nahrat na pásku a potom ho automaticky vyvolať (bez straty zásahu) a opäť nahrat do počítača. Ak si prajete manuálny štart bez straty obsahu (ústredne) štartujte s GO TO 360 radšej ako s RUN.

```

10 REM Telefónna ústredňa
15 GO TO 360
20 DIM D$(200, 32)
30 FOR D = 1 TO 200
40 INPUT "vlož meno "; B$
90 PRINT AT 0,0; "meno: "; B$
110 INPUT "vlož telefónne číslo "; C$
120 CLS
130 PRINT AT 0,0; B$ : " "; C$
140 PRINT "Ak je to správne, Enter"; INVERSE 0, " , ak
chceš opravu stlačiť"; FLASH 1; "E";
150 INPUT E$: CLS
160 IF E$ <> "" THEN GO TO 40
170 LET D$(D) = B$ + " " + C$
180 PRINT "stlač "; INVERSE 1; "ENTER"; INVERSE 0; : " pre ďalší
vstup údajov, alebo ľubovoľnú klávesu + Enter pre triedenie"
190 INPUT E$: CLS
200 IF E$ = "" THEN NEXT D

```

```

210 PRINT PAPER 2; FLASH 1; "triedenie..."
215 POKE 23692,0
220 LET B = 0
230 LET G = 0
240 LET Z = 1
250 LET B = Z + 1
260 IF B < G THEN GO TO 230
270 IF D$(B) > D$(Z) THEN GO TO 290
280 LET Z = Z + 1: GO TO 250
290 LET Q$ = D$(Z)
300 LET D$(Z) = D$(B)
310 LET D$(B) = Q$
320 GO TO 280
330 PRINT D$(G)
340 LET G = G - 1
350 IF G > 0 THEN GO TO 240
360 PRINT "vstup čísel":
370 PRINT "1 - štart novej ústredne"
380 PRINT "2 - pričítanie nových mien"
390 PRINT "3 - hľadanie čísel"
400 PRINT "4 - nahranie ústredne"
405 PRINT "5 - vytlačenie ústredne"

```

4.4 D A T A B Á Z A

```

10 REM Databáza
20 PRINT "triedenie podľa volby"
25 PRINT "1 - zmena nových buniek"
30 PRINT "2 - pričítanie k nim"
40 PRINT "3 - tlač mien umelcov"
50 PRINT "4 - tlač titulov"
60 PRINT "5 - hľadanie-triedenie umelcov"
70 PRINT "6 - hľadanie titulov"
80 PRINT "7 - nahranie databázy na pásku"
90 PRINT "8 - stop"
100 INPUT A
110 CLS
120 IF A = 1 THEN GO SUB 210
130 IF A = 2 THEN NEXT J
140 IF A = 3 THEN GO SUB 630
150 IF A = 4 THEN GO SUB 670
160 IF A = 5 THEN GO SUB 710
170 IF A = 6 THEN GO SUB 800
180 IF A = 7 THEN SAVE "DATA"
190 IF A = 8 THEN STOP
200 GO TO 20
210 DIM F$(130,24): DIM E$(130,24)
220 POKE 23692,0
230 FOR J = 1 TO 130 : CLS
240 PRINT AT 0,0; INK 2; "vlož 'z' pre ukončenie, alebo nové vstupy"
250 PRINT AT 10,10; "porad. číslo "; FLASH 1; J
260 INPUT "vstup umelec /skladateľ "; A$
270 IF A$ = "Z" THEN GO TO 390
280 PRINT " A$

```

```

290 INPUT "vstup titulu "; T$
300 PRINT T$
310 INPUT "vstup klasifikačného kódu "; C$
320 PRINT C$
330 PRINT "ak je to správne stlač "; INVERSE 1; "Enter";
INVERSE 0, " , ak nie, stlač nejakú klávesu."
340 INPUT Z$
350 IF Z$ <> "" THEN GO TO 240
360 LET F$(J) = A$ + ":+T$ +": "+C$
370 LET E$(J) = T$ + ":+ A$ + ":+ C$
380 NEXT J
390 CLS
400 PRINT INK 1; FLASH 1; "Prosím, čakajte triedim."
405 LET G = J: LET B = 0
410 LET Z = 1
440 LET B = Z + 1
450 IF B > G THEN LET G = G - 1: IF G > 0 THEN GO TO 410
455 IF B > 130 THEN GO TO 20
460 IF F$(B) < F$(Z) THEN GO TO 480
470 LET Z = Z + 1: GO TO 440
480 LET Q$ = F$(Z)
490 LET F$(Z) = F$(B)
500 LET F$(B) = Q$
510 GO TO 470
520 LET G = J: LET B = 0
530 LET Z = 1
540 LET B = Z + 1
550 IF B > G THEN LET G = G - 1: IF G > 0 THEN GO TO 530
560 IF G = 0 THEN CLS: GO TO 20
570 IF E$(B) < E$(Z) THEN GO TO 590
580 LET Z = Z + 1: GO TO 540
590 LET Q$ = E$(Z)
600 LET E$(Z) = E$(B)
610 LET E$(B) = Q$
620 GO TO 560
630 FOR M = 1 TO J
640 LPRINT F$(M)
650 NEXT M
660 RETURN
670 FOR M = 1 TO J
680 LPRINT E$(M)
690 NEXT M
700 RETURN
710 PRINT "vstup umelcovho mena"
720 INPUT N$: FOR F = LEN N$+1 TO 24: LET N$=N$+" ": NEXT F
730 FOR M = 1 TO J
740 IF F$(M) (TO 4) = N$ (TO 4) THEN GO TO 770
750 NEXT M
760 PRINT "umelec nenađený": PAUSE 200: RETURN
770 PRINT F$(M)
780 PAUSE 0
790 RETURN
800 PRINT "vstup titulov"
810 INPUT N$: FOR F = LEN N$+1 TO 24: LET N$=N$+" ": NEXT F
820 FOR M = 1 TO J

```

```

830 IF E$ (M) (TO 4) = N$ (TO 4) THEN GO TO 360
840 NEXT M
850 PRINT "titul nájdený": PAUSE 200:RETURN
860 PRINT E$ (M)
870 PAUSE 0
880 RETURN

```

Priklad na "databázu"

```

BEATLES : HELP: 008
CHOPIN : WALTZ : 34/3
. . .
FANTASIA : MOZART : K367
HELP : BEATLES : 008

```

4.5 F I N A N Č N Ý M O D E L O D B Y T U

Ako už bolo povedané pri štarte tohoto programu je finančný model hrubá verzia niekoľkých ľahkých prípravkov pre rozšírenie archov kalkulátorového programu ako je VISICALC. Tento program je označený ako forma mesačného odbytu, alebo počet zamietnutí, alebo podpora finančného obratu alebo nejaký iný výpočet inej udalosti, ktorá sa prihodí a má byť zapísaná v regulárnom intervale a pre túto informáciu extrapolovať budúci mesačný cyklus, a prijať iné faktory.

Ak zadáte RUN, najskôr sa stretnete s otázkou na 'hardcopy' (t.j. kópia obrazovky na tlačiareň), ktorá je produkt programu. Ak je na vstupe 1 - áno všetok dôležitý produkčný program vo Vašom programe bude vytlačený (LPRINT) a zároveň sa objaví na obrazovke. Nasledovne dostanete otázku: 'Pre koľko mesiacov sú tieto tvary nemenné?' Vymeníte mesiace za dni alebo roky, alebo akúkoľvek časovú periódu a môžete pracovať. Budete potom ďalej odpovedať na vstup foriem pre každý mesiac. A od toho okamihu môže program získať dáta od vstupu 19. mesiaca. Vymeníte 'A-1' na štarte riadku 100 za nulu a máte dáta viac než na 19. mesiacov.

Spectrum bude pracovať pravdepodobne od mesiaca k mesiacu. Bude porovnávať mesiac dva s mesiacom jedna, mesiac tri s mesiacom dva atď. Potom bude daná priemerná percentuálna zmena. Teraz ste schopný vytvoriť extrapoláciu výsledkov, špecifikujúce číslo mesiaca, pre ktorý si prajete projektovej tvar a či očakávate tuto projekciu založenú v poslednom mesiaci, pre ktorý máte prislusný tvar alebo priemerný výsledok za mesiac. A teraz si máte zjednať menu závislé na výbere (voľbe) čísla:

1. vstup projekcie (ktorú podmienuje možnosť zmeny štartovacieho tvaru pre priemer z posledného mesiaca alebo za kratší či dlhší časový okamih)
2. vstup pomocou output ale bez vstupu (hoci bude práve odpovedať ochotne pre projekciu)
3. vstup programu pre začiatky (inicializácia dát)

4. nahranie tohoto spustenia na pásku (ktorým budete opravovať optimalizačnú voľbu 2 tohoto menu po prehraní)

Ak ste zadal vstup pomocou tohoto programu a ak tlačiareň nepracuje a vy sa rozhodnete použiť hard copy, použijete 5 - stop programu, keď dosiahnete menu reštartujte program s GO TO 10 radšej ako s RUN. Ďalšie triedenie poradia pomocou menu a môžete zadať Vaše tvary (figúry) do printu, bez toho, že máte potom opakovaný vstup.

```

10 REM Finančný model odbytu
15 GO SUB 490
20 PRINT "pre koľko mesiacov je prístupných tvarov - foriem?"
30 INPUT M : LET TOT = 0
40 IF M < 2 THEN GO TO 30
50 CLS: DIM A(M) : DIM B (M)
60 POKE 23692,0
70 FOR A = 1 TO M
80 PRINT AT 20,0; "vstup foriem pre mesiac"
90 INPUT A(A)
100 PRINT AT A-1,0; BRIGHT 1; PAPER 0; INK 7; "Mesiac"; A; " ";
    TAB (17 - LEN STR$ A(A); A(A); "
105 IF Z = 1 THEN LPRINT "mesiac"; A; " "; TAB (17 - LEN STR$ A(A));
    A(A); "
110 LET TOT = TOT + A(A) : NEXT A
115 PRINT AT 20,0; "
120 LET AV = TOT/M: FOR B = 2 TO M
130 LET B(B) = (100 - (A(B-1)*100/ A(B)))
140 NEXT B
145 PAUSE 100
150 PRINT "FLASH 1; INK 1; "diferencia medzi mesiacmi:"
155 IF Z = 1 THEN LPRINT "diferencia medzi mesiacmi:"
160 FOR A = 2 TO M
170 PRINT "mesiac"; INK 1; A; INK 0; " - "; INT (B(A) +.5);"%
175 IF Z = 1 THEN LPRINT "mesiac"; A - 1, "do mesiaca"; A; " - ";
    INT (B(A) +.5); "%
180 NEXT A
185 PAUSE 100
190 LET TOTAL = 0
200 FOR A = 2 TO M
210 LET TOTAL = TOTAL + B (A)
220 NEXT A
230 LET PRIEMER = INT (TOTAL * 100)/(M-1)/100
235 PRINT INK 2; "-----"
236 IF Z = 1 THEN LPRINT "-----"
240 PRINT "priemerná výmena je"; INK 1; FLASH 1; PRIEMER; INK 0;
    FLASH 0; "%
245 IF Z = 1 THEN LPRINT "priemer je"; PRIEMER; "%
250 PAUSE 100
255 PRINT INK 2; "-----"
256 IF Z = 1 THEN LPRINT "-----"
260 PRINT "teraz je projekcia výmeny:"
265 IF Z=1 THEN LPRINT "teraz je projekcia výmeny:$"
270 INPUT "koľko očakávaš mesiacov pre projekciu?"; ČÍSLO
280 PRINT "finálny zaznamenaný mesiac"; A(M)
290 PRINT "priemer za mesiac", ; 10 * (INT AV/10)

```

```

300 INPUT "prejdite si zakladnu projekciu 1 - konecny mesiac;
alebo
2 - priemer mesiaca?"; D
305 PRINT INK 2; "-"
306 IF Z = 1 THEN LPRINT "-----"
310 LET E = (A(M) AND D = 1) + (( 10 * INT AV/10) AND D = 2)
315 PRINT "mesiac 1, zaznam "; A(M)
316 IF Z = 1 THEN LPRINT "mesiac 1, zaznam: "; A(M)
320 FOR A = 2 TO CÍSL0
325 POKE 23692,0
330 LET E = E + PRIEMER * E/100
340 PRINT "mesiac "; A; " projektovane: "; INT E
342 IF Z = 1 THEN LPRINT "mesiac"; A; " , projektovane: "; INT E
345 NEXT A
350 PAUSE 100: POKE 23692,0
355 PRINT INK 2; "-"
356 IF Z = 1 THEN LPRINT "-----"
360 PRINT INK 2; INVERSE 1; "vstup vašej volby:"
370 PRINT TAB 3; " 1 - vstup pomocou output - samotný,"
380 PRINT TAB 3; " 2 - vstup pomocou output - samotný, ale bez
vstupu jednotlivých foriem / figúr"
390 PRINT TAB 3; " 3 - vstup - spustenie primárne/pre začiatok/"
400 PRINT TAB 3; " 4 - nahranie na pásku"
405 PRINT TAB 3; " 5 - stop"
410 LET A$ = INKEY$
420 IF A$ = "" THEN GO TO 410
430 IF A$ = "1" THEN GO TO 260
440 IF A$ = "2" THEN GO TO 150
450 IF A$ = "3" THEN RUN
460 IF A$ = "4" THEN SAVE "odbyt"
470 IF A$ = "5" THEN STOP
480 GO TO 360
490 INPUT "čakáte výstup hardcopy? 1 - áno 2 - nie"; Z
500 IF Z = 1 OR Z = 2 THEN RETURN
510 GO TO 490

```

Časť output (výstupu) - "finančného modelu"

```

mesiac 1 1345
mesiac 2 1456
.
.
mesiac 5 1789

```

diferencia medzi mesiacmi

```

mesiac 1 až mesiac 2 - 8 %
mesiac 2 až mesiac 3 - 7 %
.
.
mesiac 4 až mesiac 5 - 6 %
-----

```

Priemerná výmena je 6.88 %

Teraz projekcia výmeny:

```

mesiac 1, zaznamenané: 1789
mesiac 2, projektované: 1912
.
.

```

```

mesiac 6, projektované: 2496
-----

```

- 1 - spustenie samotnej projekcie
- 2 - spustenie pomocou output, ale bez vstupu foriem
- 3 - spustenie programu pre začiatok
- 4 - nahranie tohoto spustenia
- 5 - stop

5.0 P O U Ž I T I E V O V Ý C H O V E

Táto kapitola je písaná pre tých, ktorí si prajú použiť Spectrum ako výpomoc pre vyučovanie alebo pre rodičov, ktorí pomáhajú svojim deťom alebo učiteľom používajúcim počítač v triedach. Tak, či tak je tu príčina, prečo nemôžete použiť programovanie len ako nápad, ak si prajete pomáhať si s opravovaním cvičení, alebo práve pre cvičenie v namáhaní mozgu. Je celkom zaujímavé, ako sa vela môžete naučiť práve skúšaním písania programov, ktoré pomáhajú zasa v inom učení. Ale ako hlavné môžete veľmi dobre pozdvihnúť povrchné znalosti z francúzštiny, fyziky alebo čohokoľvek iného, pokiaľ sa diviate na uskladnené informácie ako na kvízové programovanie DATA príkazov.

Obchodné, výchovné programy pre Spectrum (alebo iné počítače pre tieto predmety majú sklon dovolovať jeden alebo dva skoky. Pre niektoré len prehľadne a jednoducho pre čírobné použitie, alebo len pre veľmi málo špecializácii. Zamerajte sa s našim programovaním, s ktorým sa u počítača stretnete oboma týmito smermi. Môžete si urobiť program presne tak, ako ho potrebujete, učiť a testovať práve tie partie a časti, avšak tie, ktoré sa objavja pre čiastočnú potrebu. Podprogramy a programy rozoberané v tejto kapitole môžu byť užitočné pre adaptáciu Vašich špecifických potrieb.

Kapitola obsahuje programy, ktoré dávajú praktickú základnú znalosť. Také ako numerácia alebo (v prípade jedneho programu) znalosť francúzštiny. Je tu vela iných typov výukových programov. Môžeme napr. písať, o ktorých nových materiáloch sa bude učiť alebo hadam dať miesto laboratornej práci (takýto program, ktorý kombinuje chemické kvanta a ukazuje výsledky zlučenia).

V mnohých prípadoch nájdete jeden alebo viac programov tvoriacich vzor k výrobe programov, ktoré potrebujete.

Vyučovacie programy, ako uvidíte neskôr, nemusia byť ťažkopádne. Niektoré výukové programy môžu byť písane rovnako jasne ako hry. Výukové hry požadujú na hráčoch vývoj a aplikáciu na takom stupni, aby sa mohlo súťažiť s inými študentmi.

Je to samozrejme pracné, ale bude úspech, ak použijete efektívnejšie Vašich znalostí, napríklad je dobrým spštením programov náhodný faktor, ktorý určuje náhodný výsledok hier.

Program pre výuku nových materiálov môže byť veľmi dlhý, pretože programátor potrebuje predvídať existenciu chýb a zmien koncepcie, ktoré môžu vzniknúť pri prevádzaní korekcií programov. Ten istý materiál budeme potrebovať pre nespočet variantov.

Napríklad, keď deti neporozumejú prvému výkladu, podari sa to po chvíli, keď informáciu prezentujeme inou cestou. Existuje trieda

programov, ktoré sú zapísané a slúžia ako príručka študentom, ktorí vytvárajú programy. Potom tento program môže byť použitý viackrát ako inteligentný sprievodca knihou ako 'stály, trvalý učiteľ'.

Teraz sa pozrieme, ako porozumieme lámaizmu vzniknutému a uctievanému oddelením Mahayanskej formy z Budhizmu. Ideme na 26 stranu knižky a dozvieme sa, ako bol zavedený do Tibetu. Ak sa vrátíme späť k počítaču, zaznamenáme pod heslom lámaizmu titul knižky a stranu. Ak sa v budúcnosti vrátíme k tomuto materiálu, stačí na klávesnici vytukáť heslo lámaizmu a počítač nám vypíše dostupné pramene, t.j. aj archívny poznatok, že dôležité informácie si ľahko vyhladáme na strane 26.

Používajúc program takýmto spôsobom zistíme, že programový čas nie je viazaný na nekonečný počet PRINT príkazov, a pritom je možné pokryť takmer všetky pravdepodobnosti a použiť svoj Spectrum k pomoci vo výuke subjektov inak zdanlivo veľmi vzdialených komplexov pre počítač. Projekty majú regulárne rysy základného školskeho vyučovania, pretože študenti majú často problémy so systematickým zaradením primeraných informácií.

Spectrum môže byť použitý k skladovaniu informácií. Môže byť použitý k týmto účelom s podobným menu ako systém PRESTEL. (Prestel je jedným z viacerých telekomunikačných sieťových systémov, do ktorého sú zapojené mnohé západné domácnosti prostredníctvom domáceho počítača - môže si to predstaviť ako kvalitnejší teletext.) Alebo môže byť použitý ako pomoc pri lokalizácii primeraných zdrojov informácií.

Okrem bezprostredného prospechu pre týchto študentov môže nám pomôcť vedomie stúpajúceho významu informácií v modernej spoločnosti. Ak školské materiály alebo materiály pre domácnosť nemajú prístup k Prestelu, tak poslúži Prestel-simulácia s limitovaným počtom výmen, čo pomáha študentom zoznámiť sa dôverne s technikou použitia podobných systémov.

Simulácia iných typov dovoľuje viesť výskum (vyšetrenie), ktoré nemôžu byť prevedené v reálnom čase. Spectrum môže použiť matematický model pre predpovedanie výsledku z vypracovania zmien nejakého systému. Môžeme sa napríklad pýtať, aký je trend znečistenia rýbníkov, aký bude mať vplyv na ryby, hmyz a rastliny. Simulácia v ekonomike pripúšťa zmenu zásobovania peniazmi a uvidíme, ako tento efekt využijeme. Chemická simulácia dovoľuje meniť katalýzu zmenou teploty a tlaku a dovoľuje optimalizáciu procesu alebo hovoriť, koľko kyslíčnika siričitého alebo kyseliny sirovej ohrozuje rastliny.

Avšak simulácia môže byť úplne presnou reprezentáciou systému a totálne verná, ak je model úplnou reprezentáciou systému. Rovnako, ak nie je úplná presnosť, študenti sú oboznámení so zjednodušenou simuláciou, čo môže byť ešte názornejšie.

A na koniec tejto kapitoly budeme diskutovať o určitom špecifickom priblížení sa k výkove programu ktorým máme vyskúmať úspešne ine programy s ideou o povahе veci, ktorá môže byť užitočná pri transformovaní programov. Ako prvý použijeme špecifický program, ktorý bude ukazovať diferenciálne priblíženie k produkcii výukových programov. Rovnako tento program nie je pre bezprostredné použitie, takže vy v tejto forme máte pracovný program pre výrobu vášho programu.

Vela vyučovacích programov používa funkciu RND, t.j. generovanie náhodných znakov, takže sa učenie nestáva nudným pri predpovedaní postupnosti riešení (otázok). Náš prvý program je postupnosť náhodne generovaných jednoduchých delení, ale v širšom pohľade to môže byť užitočné pri tvorbe programu na skúšanie základných aritmetických operácií

V riadku 5 vytvoríme pomocou USR definovanej grafiky symbol D v normalne značených minusových znamienkach a je to použité v školských

pracáčach. Počiatočné skóre S je rovné nule. Program zadáva 20 otázok použitím FOR-NEXT slučky, jadro cyklu leží na riadkoch 20 až 120.

Otázky sa zadávajú formou "Koľko je 32 delené 4-mi?" - Spectrum delenie matematicky zapisuje ako X/Y. Riadok 30 dáva hodnotu medzi 2 - 12 a riadok 40 produkuje hodnotu X, ktorou zadávame celé čísla medzi 1 - 12 ako konečnú odpoveď. Riadok 50 zobrazí otázku na obrazovku a akceptuje odpoveď použitím INPUT. Riadok 60 vyberá medzi postupnosť riadkov 70 a 80 alebo chváli na riadku 90, čo vývrcholí výpisom dosiahnutého skóre.

5.1 T E S T D E L E N I A

```
5 FOR F=0 TO 7: READ A: POKE USR "D" + F,A:NEXT F : DATA
  0,16,0,254,0,16,0,0
15 LET S = 0
20 FOR Q = 1 TO 20
30 LET Y = 2 + INT (RND * 11)
40 LET X = Y * INT (RND * 12 + 1)
50 PRINT AT 9,0: PAPER 6; "OTÁZKA Č. "; Q :PRINT "KOĽKO JE"; X;
  "DELENE"; Y; "-mi ?": INPUT A
60 IF A = X/Y THEN GO TO 90
70 PRINT PAPER 6; "Zle, toto nie je správna odpoveď"; A
80 PRINT" X; "D"; Y; " = "; X/Y: PRINT PAPER 6; "(spomeň si,
  že"; Y; " "; X/Y; " = "; X; "): GO TO 100
90 PRINT "dobré."; A; " je správne": LET S = S + 1
100 INPUT "stlač ENTER pre pokračovanie";C$:IF C$ = "s" THEN STOP
110 CLS
120 NEXT Q
130 PRINT AT 0,4; "máte skóre"; S;
```

Je isté, že obtiažnosť otázok závisí na schopnosti žiakov. Nedopustíte, aby Spectrum produkovalo otázky druhu: koľko je 19 delené 7?, pretože žiak vie málo (v porovnaní s počítačom, ktorý okrem iného zvykne drzo zaokrúhľovať) o podiele nesuditeľných čísel a desatinách.

Môžete vytvoriť svoj program s väčším ohlasom tak, aby používateľ použil kontrolu skóre pri obtiažnosti otázok. Jeden spôsob vytvorenia tejto algoritmickej perličky v programe delenia je prepísaný riadok 30 takto: 30 LET Y = 2 + INT (RND *(3 + (S >= 5) * 4 + (S >= 10)*.4)) Pokiaľ korekčné skóre odpovede dosahuje 5, bude sada v rozsahu 2-4, pretože logický výraz (S >= 5) a (S >= 10) sa bude pre oba rovnat nule: potom obidva výrazy budú nepravdivé. Ak skóre je 5 a viac, potom výraz (S >= 5) obdrží hodnotu pravda (true) a dostane hodnotu 1 a tak bude dané v ako hodnota medzi 2 až 8. Konečne, ak skóre dosahuje 10, bude pre Y dané ako maximálna hodnota 12. Podobne vymeníme riadok 40

```
40 LET X = Y * INT (RND * ( 5 + (S >= 5) + 1)
```

Výhodou tohoto typu programového kontrolovania je, že prechod tohto tvrdeho problému je hľadší a skúšanie bude pre oboch ľahšie a zábavnejšie. Hoci je program napísaný pre delenie, je ľahké tu látku meniť i pre iné typy výpočtov. Môžeme zmeniť druhú časť riadku 50: "Koľko je 1/"; Y; " z "; X; " ?"

Táto nová otázka typu "koľko je 1/4 z 32?" je pre prax v zlomkoch. Budete potrebovať zmeniť riadok 80 kvôli adekvátnej forme nového formátu otázok. Skúste písať verzie tohoto programu na sčítanie, odčítanie

(pamätajte na to, aby bolo prvé číslo väcsie ako druhé, ak študenti nepoznajú záporne čísla) a násobenie.

Veľa programátorov rado tvorí pre veľa osôb a vtedy je vhodné, aby počítač korespondoval s viacerými menami. Ak si to tiež prajete, budete potrebovať napísať podprogram, ktorý konverzuje nasledujúcim štýlom.

Spectrum: "Haló, aké je Vaše meno?"

Studentka: "Samantha"

Spectrum: "Dobre Samantha. Dufam, že si pripravená odpovedať na niektoré otázky o anatómii potkanov"

Popripade môže byť k dispozícii 'pochvalná fráza' v niekoľkých mutáciách, z ktorých je jedna náhodne vybraná pre každú správnu odpoveď.

"To je správne Samantha - správne je holenná kosť"

"Dobre Samantha, správne je femur"

Táto metóda dá veľkú satisfakciu deťom, špeciálne tým, ktoré nadšene rozprávajú o počítačoch a robotoch z science fiction.

Spectrum môže využiť detský súťažiaci inštinkt pre zlepšovanie záznamu dobrého skóre, v závislosti od času, ak vytvoríte premenne usporiadanie s časom ohraničeným pre odpoveď a otázky. Túto prácu študenti veľmi radi skúšajú, či on alebo ona prekona predchádzajúci výkon, ale môže slúžiť k stráženiu pomalších detí v skupine, takže musíte používať opatrnejšie formulácie pri výhodnocovaní.

Nasledujúci program "aritmetický kvíz" ukazuje, ako študenti začínajú používať počítač a obecné ako sa priateľsky približia k programu s delením. Takže je dovolené študentom meniť typ otázok (scítanie, odčítanie, násobenie alebo delenie), čím zlepšíme pestrosť programu.

5.2 A R I T M E T I C K Ý K V Í Z

```
10 REM Aritmetický kvíz
30 PRINT "Ahoj,"; "";"skúsime niekoľko výpočtov!"
```

```
40 PAUSE 200
```

```
50 CLS
```

```
60 PRINT;"Tak, aké je Tvoje meno?"
```

```
70 INPUT A$
```

```
80 CLS
```

```
90 PRINT;"Teší ma "; FLASH 1; BRIGHT 1;
```

```
INK 1; A$
```

```
100 PRINT;"Stlač akúkoľvek klávesu, ak si pripravený";
```

```
110 PAUSE 0: LET SCORE = 0
```

```
120 CLS
```

```
130 PRINT;"teraz ľa, "; A$; ", vyskúšam z týchto úkonov:"
```

```
140 PRINT;"1 - súčet"
```

```
150 PRINT;"2 - odčítanie"
```

```
160 PRINT;"3 - násobenie"
```

```
170 PRINT;"4 - delenie."
```

```
180 PRINT;"Vyber si z čísel "; FLASH 1; "1 až 4" ; FLASH 0;
```

```
" podľa toho, na čo máš chuť."
```

```
190 INPUT B$: LET B = VAL B$
```

```
200 IF B<1 OR B>4 THEN GO TO 190
```

```
205 CLS
```

```
210 PRINT;"O.K., "; A$; ", vybral si si "
```

```
220 PRINT INK 2; FLASH 1; ("scítanie" AND B = 1) +
```

```
("odčítanie" AND B = 2) + ("násobenie" AND B = 3) +
```

```
("delenie" AND B = 4)
```

```
230 PAUSE 300
```

```
240 CLS
```

```
250 PRINT;"Zadaj ľ ak očakávaš ľahké počty ", " alebo 5,
```

```
ak očakávaš náročnejšie,"; "môžeš vložiť 2,3
```

```
alebo 4, podľa požadovanej úrovne."
```

```
260 INPUT C$: LET C = VAL C$
```

```
270 IF C<1 OR C>5 THEN GO TO 200
```

```
275 CLS
```

```
280 LET D$ = ("+" AND B = 1) + ("-" AND B = 2) + ("*" AND B =
```

```
3) + ("/" AND B = 4)
```

```
290 FOR E = 1 TO 10
```

```
300 LET A = INT (RND * (RND * (10 + C) + C)
```

```
310 LET B = INT (RND * (10 + 4 * C) + C)
```

```
320 LET G$ = STR$ A + D$ + STR$ B
```

```
325 IF ABS (INT VAL G$ )<> VAL G$ THEN GO TO 300
```

```
330 PRINT "teraz, "; A$; " je "; TAB 8;" otázka číslo"; E
```

```
340 PRINT;"koľko je "; G$; "?"
```

```
350 PAUSE 50
```

```
360 PRINT;"napiš tvoju odpoveď."
```

```
370 INPUT F
```

```
380 IF F = VAL G$ THEN BEEP .08,2.5 * SKÓRE: BEEP 1,5 *
```

```
SKÓRE: PRINT FLASH 1; INK RND * 7; PAPER 9; "Dobre, "; A$; LET
```

```
SKÓRE = SKÓRE + 1
```

```
390 IF F<>VAL G$ THEN PRINT "Ľutujem, "; A$; ", ale", " odpoveď
```

```
je", VAL G$
```

```
400 PAUSE 50
```

```
410 PRINT;"Vaše skóre je teraz"; INK 1; FLASH 1; SKÓRE; INK
```

```
0; FLASH 0; " z počtu pokusov "; E
```

```
420 IF E<10 THEN PRINT;"Stlač ktorúkoľvek klávesu pre ďalšiu
```

```
otázku.": PAUSE 0: CLS: NEXT E
```

```
430 PAUSE 100
```

```
440 CLS
```

```
450 PRINT;"ku koncu nám to dáva", " toto resumé, "; A$;"."
```

```
460 PAUSE 100
```

```
465 PRINT;"Skóre bolo "; INK 1; FLASH 1; SKÓRE; INK 0;
```

```
FLASH 0; " z počtu "; E; TAB 8; FLASH 1; BRIGHT 1; INK 2; SKÓRE *
```

```
100 /E; "%"
```

```
470 PRINT;" Zopakujeme si to ? ", " Stlač "A" ak
```

```
áno, alebo", "stlač ENTER k ukončeniu."
```

```
480 INPUT Y$
```

```
490 IF Y$ = "A" OR Y$ = "a" THEN GO TO 100
```

```
500 PRINT A$;" dnes to nebolo najhoršie. Teším sa na budúce."
```

```
520 PRINT;"FLASH 1; " DOVIDENIA"
```

```
530 STOP
```

Ako uvidíte, keď spustíte program, bude použité FLASH. A farba uvedie do popredia (v pozornosť) dôležité veci v programe. Tak ako predtým je použité meno študenta. Konverzácia v tomto programe postupuje nasledovne:

"Ahoj, "

"Skúsime niekoľko výpočtov",
"Tak, ako je Tvoje meno? (Student tu vstúpi so svojim menom)"
"Teší ma (FLASH, BRIGHT, modrý INK), Samantha"
"Stlač akukoľvek kľávesu, ak si pripravený."

PAUSE 0 je tu použitá (riadok 110) k tomu, aby udržala displej pokiaľ je stlačená kľávesa. Premenná SKORE (ktorá uchováva skóre, prirodzene ak je) je inicializovaná s 0 a zmaže sa obrazovka (riadok 120). Počítač potom prevezme sám konverziu:

"A teraz fa Samantha vyskúšam z týchto úkonov:"

- 1 - sčítanie
- 2 - odčítanie
- 3 - násobenie
- 4 - delenie

Vstup čísla pomocou (blikanie FLASH 1) 1 až 4 (FLASH 0 vypnuté) nám určí, ktorý typ výpočtu môžeme vyskúšať. Vstupy študentových čísiel, ktoré môžeme vidieť na riadku 140 su akceptované stringom (retazcom), ktorý je konvertovaný použitím príkazom VAL na číselnú hodnotu.

Mimochodom je dôležité písať programy tak, že nekorektný vstup nezavlní prípad zastavenia, či zruštenia programu. Toto nie je pine kryté v prvom programe tejto kapitoly. Program delenia, kde je vstup inštalovaný pomocou písmen potom vstup čísiel (čísel) zastaví ako hlásenie chyby. Tento problém môže byť prekonaný keď vstúpujeme vždy s retazcovými premennými. Môžeme použiť funkciu CODE čím bezpečne vytvoríme vstup s číslami ak sú tieto požadované. Potom môžeme premeniť retazec opäť v čísla používajúc VAL. Skuste nasledujúce riadky pre jednoduché čísllice:

```
10 INPUT N$
20 IF LEN N$ > 1 THEN PRINT "prosím jednoduché čísllice.":
GO TO 10
30 IF CODE N$ <48 OR CODE N$ >57 THEN PRINT "prosím len
čísla": GO TO 10
40 LET N = VAL N$: PRINT "číslo ";N;" je zobrazené znakom ";
N$;" a tento má hodnotu ASCII "; CODE(N$)
```

Spustíte a môžete vidieť kompletný program pre vstup jednoduchých čísiel. Čísla majú znak s ASCII hodnotou od 48 do 57 vrátane. Pre mnohociferné čísla môže zapojiť oddelený vstup a použitím rozdelenia retazca na cifry a identifikáciu znaku v slučke určiť ASCII hodnotu cifier väčších čísel. Popripade môžeme použiť alternatívu použitím INKEY\$. Tak, aby prichádzali čísla ako jednotlivé čísllice a pomocou CODE INKEY\$ = 13 (čo je kód pre stlačenie ENTERU) uzavrieme vstup čísla. A teraz späť k 'aritmetickému kvizu'. Počítač čaká na voľbu vstupu čísiel od jednej do stýroch podľa typu otázky, ktorú hľadáme.

Poznámka: Všimnite sa ako pracuje Spectrum pri spracovaní logického výrazu AND pri voľbe jedného zo stýroch slov (sčítanie, odčítanie, násobenie alebo delenie) na hodnote, ktorá je daná B. Nasleduje PAUSE (riadok 230) a čistenie obrazovky. Zhovorčivé Spectrum pokračuje:

"Zadaj 1, ak očakávaš jednoduché počty alebo 5 ak očakávaš náročnejšie (môžeš zvolit 2,3 alebo 4 podľa požadovanej úrovne."

Zadanim stupňa obtiažnosti dosiahneme variabilitu výuky, čím sa program prispôbi schopnostiam osoby, ktorá s počítačom pracuje.

Vstup je akceptovaný ako retazec, ale vzápätí ho prevracia v číslo a potom maže obrazovku. Riadok 280 používa AND na logické vyhodnotenie, určujúce aritmetickú signatúru (sign.), tj. (+,-,*,/), ktorá bude použitá pre žiadanú otázku.

Riadok 290 štartuje cyklus pre 10 otázok, ktoré pre študenta vygenerujú riadky 300 a 310 vo forme náhodných čísel s ohľadom na stupeň obtiažnosti. Riadok 320 sčíta čísla a známenko prislušnej operácie dohromady do jedného retazca. Riadok 325 zabráni tomu, aby bol výsledok aritmetickej oprácie záporný, prípadne desiatinné číslo. Číže, ak by výsledok nebol celé nenulové číslo - matematicky nazývané prirodzene číslo, program vygeneruje nový príklad, až kým nie sú splnené tieto podmienky.

Riadok 330 tlačí výstup otázky používajúc meno študenta. Riadok 380 obdrží odpoveď a ak je táto správna, odmeni sa študentovi, čo je spojené s pipnutím. Výška pipnutia je v relácii s číslom korektnosti (správnosti) odpovedi, ktorú dostane v priebehu testu. Pochvalné slová sú vytlačené na obrazovke s náhodne triedeným blýskaním (FLASH) farieb. Použitie INK 9 zabezpečuje, že môžu byť prečítané slová (so stupaajúcim skóre). Ak je odpoveď zadaná zle, vylučí riadok 390 uzornenie na chybu a zobrazí sa správny výsledok.

Teraz je tu iná krátka PAUSE a Spectrum vypíše aktuálne skóre. Prebehne všetkých desať príkladov - koniec cyklu a na konci desiatej otázky, po pauze sa obrazovka vycisti. Potom sa výhodnoti úspešnosť odpovedi.

Riadok 480 akceptuje študentovu odpoveď a riadok 490 zabráni popluteniu "y" s "y" a určuje, či má počítač CAPS LOCK alebo nie.

Sú viaceré spôsoby ako človeka a dieta predovšetkým motivovať k práci s výukovým programom. Velmi populárnym spôsobom je kombinácia výuky s hrou. Takéto niečo si ukážeme v nasledujúcom programe. Na začiatku programu bude na monitore obrazovky zobrazená rieka s nekompletným mostom s tankom na ľavom brehu. Každú chvíľku sa musí správne odpovedať a potom sa plocha mostu sa rozširuje.

AK JE 12 ODPOVEDÍ SPRÁVNÝCH , PREJDE TANK MOST A PÁLI Z DELA.

Prvá zmena je v riadku 5 programu "delenie":

```
5 GO TO 150
```

a okrem toho ešte pričítame ku koncu riadku 90 príkaz

```
GO SUB 250
```

Výmena riadku 110:

```
110 PRINT AT 8,0: FOR N = 1 TO 11: PRINT B$: NEXT N
```

Teraz sa pričítajú nasledujúce riadky do Vášho programu:

Pričítané riadky do programu "delenia"

```
140 IF S = 20 THEN PAUSE 75: GO TO 260
```



```

145 STOP
150 FOR N = 1 TO 4: READ S$
160 FOR M = 0 TO 7: READ A: POKE USR S$ + A, A: NEXT M
170 NEXT N
175 REM Znaky a,b,c,d v nasledovných dátach sú v G-móde
180 DATA "A", 7,15,7,255,127,63,27
190 DATA "B", 192,254,192,255,255,255, 254,108
200 DATA "C", 255,0,0,0,0,0,0,0
210 DATA "D", 0,16,0,254,0,16,0,0
220 LET B$ = ""
230 PRINT AT 3,0: "AB" : PRINT PAPER 4: " " : PRINT AT 4,5;" " :
    PRINT AT 4,24: PAPER 4;" " : PRINT PAPER 1; BS ; B$
240 GO TO 10
250 PRINT AT 4,3 + S; OVER 1; "C": RETURN
260 FOR N = 0 TO 10: FOR A = 4 TO 8 STEP .5: BEEP 0.01, M*M:
    NEXT M: NEXT N
270 FOR N = 0 TO 23: PRINT AT 3,N: "AB": PAUSE 5: NEXT N
280 FOR N = 1 TO 50: PLOT 215,150: DRAW INK 2; 36,0:
    PLOT 215,150: DRAW INK 7,36,0: NEXT N
290 PRINT AT 9,16: "dobre"

```

Riadky 150 a 170 sú pre definovanie grafických symbolov ku ktorým určené dáta sú na riadkoch 180 až 210. Definované b\$ je na riadku 210 a použije sa pre riadok 110 len pre čistenie obrazovky, ponechávajúc obraz mosta a tanku neporušený. Obraz je zmenený riadkom 230. Píšeme to grafickým kurzorom, čiže ak napíšeme "AB", tak po predefinovaní to bude tank. Tak isto bude "C" na riadku 250 a "AB" na riadku 270 v grafickom režime.

Riadok 250 kreslí nový úsek mostovky po každej správnej odpovedi. Riadok 260 hrá víťazne tóny ak sú všetky odpovede správne, riadok 270 kreslí tank na moste a riadok 280 páliace delo.

Tu vidíme program v činnosti:

Otázka č. 2

Koľko je 10 delené 5timi?

Dobre, 2 je správne

Stlač ENTER na pokračovanie

Otázka 12

Koľko je 18 delené 9timi?

Zle, odpoved' nie je 3

18/9 = 2

Stlač ENTER na pokračovanie

Skorujete 20 z 20:

"Dobrá práca!"

Môžete však pripojiť 'tank' k akémukoľvek výučovaciemu programu alebo k vyskúšaní sučtovej motivácie. No pozor, ak tento podprogram budete používať príliš často, môže sa Vám sprotiviť, pretože Vám zovšednie a bude Vás omielaný efekt unavovať.

Radšej skuste vytvoriť varianty tohoto priebehu, zmenou vzťahov bodov v pamäti. Zabezpečte, aby sekcia s odmeňovaním neinterferovala s výučovacím aspektom programu. Ak je trvanie dlhšie ako sekundu, alebo dlhšie ako otázka, bude to skoro zdĺhavé a otravné. Dráždivosť je podnecujúca a povzbudzuje deti naháňať sa za uspechom a to radšej než by sa vyvíjávali pohrôm. Netrešajte nedostatky a opomenutia, pretože toto vyvoláva u niektorých detí pocit úzkosti, ktorý nevytvára pocit radosťného učenia.

Môžete sa jemne prispôbiť pri vašom styku s písaním výučovacích programov. Tu je čiastočný dôvod, prečo na príklad bude počítač potrebovať oba generované problémy a zmeny odpovedí. Naš budúci program 'výučovanie' (napríklad s generovaním výučovania typu ax + by = z) sa len vypracuje problém a vytlačí na obrazovku. Keď študent zadáva odpoveď, on alebo ona proste stlačí nejakú klávesu (okrem BREAK) a odpoveď s hodnotou udanou v počítačovej otázke je daná. Môžete si žiadať vstup tohto programu do vášho Spectra pre skúšanie, pričítanie a tvorenie práce a udržania skóre programu a k akceptovaniu a zhodnoteniu študentských odpovedí.

5.3 R O V N I C E

```

10 REM Rovnice
20 INK 7: PAPER 1
30 BORDER 1: CLS
40 DEF FN M(N) = INT (RND * N) + 1
50 LET A = FN M (10)
60 LET B = FN M (10)
70 LET C = FN M (10)
80 LET D = FN M (10)
90 LET X = FN M (10)
100 LET Y = FN M (10)
110 LET E = A * X + B * Y
120 LET F = C * X + D * Y
130 PRINT INK 6: "rovnice sú:"
140 PRINT TAB 8; INVERSE 1; A; "x + "; B; "y = "; E
150 PRINT TAB 8; INVERSE 1; C; "x + "; D; "y = "; F
160 PRINT "musíte riešiť.": INVERSE 1; "x"; INVERSE 0;
    " a "; INVERSE 1; "y"
170 PRINT "stlač nejakú klávesu pre riešenie"
180 PAUSE 0
190 CLS
200 PRINT "TAB B; A; "x + "; B; "y = "; E
210 PRINT "TAB C; "x + "; D; "y = "; F
220 PRINT, TAB 4; "hodnota z x je "; FLASH 1; X
230 PRINT, TAB 4; "hodnota y je "; FLASH 1; Y
240 PRINT, TAB 8; A; "*" ; X; " + "; B; " * "; Y; " = "; E
250 PRINT, TAB 8; C; "*" ; X; " + "; D; " * "; Y; " = "; F
260 PRINT "stlač ktorukoľvek klávesu pre nový problém"
270 PAUSE 0
280 RUN

```

Příklad na rovnice

rovnice sú: $3x + 1y = 13$
 $7x + 1y = 25$

musíte riešiť pre x a y

stlačte ktorúkoľvek klávesu pre riešenie

$3x + 1y = 13$
 $7x + 1y = 25$

Hodnota x je 3

Hodnota y je 4

$3 * 3 + 1 * 4 = 13$
 $7 * 3 + 1 * 4 = 25$

Stlač nejakú klávesu pre nový problém atd.

Náš nasledujúci program "mačky a ich veci" je pre veľmi malé deti. Je využiteľný príkaz INKEY\$ a program má presný postup. Náhodné čísla tankov, nákladných vozov a mačiek sú na obrazovke a deti môžu hádať, či lepšie povedané odhadovať, ich presné číslo (správny počet).

Poznámka k rozdielu medzi použitím INPUT a INKEY\$. INPUT dáva možnosť na opravu omylov. Narozdiel od toho INKEY\$ má výhodu, že nemusíte navyše tlačiť ENTER. Program proste prebehne daným miestom s INKEY\$ a skontroluje, čo ste stlačili. V každom prípade sa vôbec nečaká ako u INPUT, ale program pokračuje. Ak nebolo nič stlačené, tak výraz INKEY\$=" má pravdivosťnú hodnotu 1.

Toto urýchľuje program a dáva mu profesionálny image. Tanky, nákladky a mačky sú tlačené pomocou UDG (užívateľom definovanej grafiky) symboly A a B, užitočné dáta sú držané v riadkoch 510 až 530. Typ objektu je triedený generátorom náhodných čísiel na riadku 20 a vypočítané RESTORE pre DATA pointer (ukazovateľ) správnych riadkov.

Riadky od 30 do 70 označujú mená objektov N\$ a vypínajú grafické symboly.

Riadkami 80 až 140 si volíme číslo objektov, tlačiac ich na obrazovku do polohy špecifikovanej dátami v riadku 600 a hľadanej detmi ako iné dané objekty. Poznámame, že musíme použiť kurzor G, keď píšeme AB na riadok 120. BEEP je použité v návaznosti na vizuálny dojem pre pozornosť deti do správnej sekcie, riadkami 180 - 230. Pochvala doprevádzaná so stúpajúcou postupnosťou poznámok je daná riadkami 250 a 260.

Môžete si priať zdokonalenie Vášho programu spštením o zvláštne návrhy k vytvoreniu skóre a vytlačením ho na koniec sady otázok. Môžete teda vytvoriť displej s atraktívnejším zavedením generátora náhodných čísiel pre INK farby na riadok 140.

Skúste zväčšovať rozsah objektov. A čo skúsiť to s pomocou pavúkov alebo vtákov. Dajte deťom nejaký štvorcový papier a nech si nejaký nový typ samé vynájdú. Napíšte súčtové DATA riadku od 540 a zväčšujte číslo na riadku 10 z 3 na požadované číslo rozdielnych typov objektov. Iné variácie môžu byť vytvorené náhodnými číslami pozícií objektov. Variácii je

bezpečet a smer vašich experimentov určuje, aké zázraky z toho vytlčíte.

5.4 MAČKY A PREDMETY

```
5 LET B$ = ""
10 LET I = INT (RND * 4)
15 REM Výmena tvarov objektov
20 RESTORE 500 + 10 * I
30 READ N$
40 FOR N = 1 TO 2: READ S$
50 FOR M = 0 TO 7
60 READ A: POKE USR S$ + M,A
70 NEXT M: NEXT N
75 REM Netlačené na objekty
80 LET Z = 1 + INT (RND * 9)
90 RESTORE 600
100 FOR N = 1 TO Z
110 READ X,Y
120 PRINT AT X,Y: "AB"
130 NEXT N
140 PRINT AT 15,0: "koľko"; N$; "je ich tu?"
150 GO SUB 410
160 IF CODE I$ < 48 OR CODE I$ > 57 THEN GO TO 150
170 IF I$ = STR$ Z THEN GO TO 250
180 PRINT AT 15,0: B$: RESTORE 600
190 FOR N = 1 TO Z
200 READ X, Y
210 PRINT AT X, Y-1: PAPER 5; N
220 BEEP 1,N
230 NEXT N
240 GO TO 280
250 PRINT AT 16,0: "dobre"; B$: AT 7,0: FLASH 1; Z
260 FOR M = 0 TO 4 STEP .1: BEEP .02,M * M: NEXT M: BEEP 1,20
270 PRINT AT 7,0: Z
280 PRINT AT 21,0: "stlač ENTER"
290 GO SUB 410
300 IF CODE I$ <> 13 THEN GO TO 290
310 CLS: GO TO 10
400 REM Vstup jednoduchých kláves (klúčov)
410 IF INKEY$ <> "" THEN GO TO 410
420 IF INKEY$ = "" THEN GO TO 420
430 LET I$ = INKEY$
440 RETURN
500 REM Data
505 REM "A" a "B" su samozrejme písané v grafickom móde - kurzor G
510 DATA "tankov ", "A", 7,15,7,255,255,127,63,31,"B",
192,254,192,255,255,254,252,248
520 DATA "mačiek ", "A",0,0,128,127,63,56,40,40,"B",20,28,
28,252,240,80,80,80
530 DATA "aut ", "A", 0,0,0,255,255,255,255,255,
48, "B",28,30,30,255,255,255,255,12
600 DATA 4,5,10,5,4,11,10,11,7,8,4,18,10,16,4,24,10,24
```

Tak doteraz sme sa sustredili na numerické problémy v tejto kapitole,

ktoré nám uľahčia písanie programov na Spectre. Ale toto nie je prekážka, aby sme sa nezozámili aj s príkladmi užitočných programov, ktoré sú založené na relatívne jednoduchom princípe.

Nasledujúci program 'Anglický slovník' kladie otázky a odpovede do príkazov DATA a je skutočne jednoduché využívať ich alebo ich adaptovať pre jednoduché slovné odpovede ako v kvízovom programe. Poznávame, že v tomto programe musíme zaistiť CAPS LOCK tak, že otázky a odpovede sú zadávané iba vo veľkých písmenách.

Máme za to, že je v tomto programe dôležité poistiť sa proti opakovaniu tých istých otázok t.j., že sa nebudu opäť hľadať v priebehu jedného skúšania. Ak si to prajete, môžete použiť alebo adaptovať mechanizmus, ktorý v tomto programe zabezpečí, že slova nebudu hľadané úplne náhodne, ale vždy sa zapamätá, že výraz už bol preskúšaný a druhý raz sa pri vyhľadávaní preskóci. Na riadku 30 je rozmer poľa a ako už vieme, nulový prvok poľa počítac ignoruje. V DATA príkazoch vidíte, že po slovenskom slove nasleduje anglické, ktoré je doprevádzané číslom.

Keď je tento pár slov volený, potom (pri čítaní náhodného čísla z DATA príkazov - vid' riadok 110 až 130) je označená hodnota ako premenná X. Každý pár slov má vlastné poradové číslo a tento element priestoru (poľa) táto pamätová bunka poľa sa v prípade vyhľadania toho ktorého výrazu nastaviť na hodnotu 1 na riadku 160. Riadok 150 zabráňuje, aby ste tento element videli, ak bol už raz pri skúšaní použitý, a potom beží voľba iného páru slov.

Vidíme, že tento program i napriek tomu, že je jednoduchý, nájde širšie využitie, najmä v rôznych variáciách.

5.5 A N G L I C K Ý S L O V N Í Č E K

```
10 REM Slovník angličtiny
20 LET SKORE = 0: PAPER 1: INK 7: BORDER 1: CLS
30 DIM A (20)
40 PRINT TAB 5: "Vitajte na angličtine!"
50 PRINT TAB 10: "Slovný test"
60 PRINT "Koliko výrazov si chcete preskúšať?"
70 PRINT "Zadajte počet: (1 - 20)"
80 INPUT B
90 IF B<1 OR B> 20 THEN GO TO 80
100 FOR C = 1 TO B
110 FOR D = 1 TO INT (RND * 20) + 1
120 READ E$: READ F$: READ X
130 NEXT D
140 RESTORE: CLS
150 IF A(X) = 1 THEN GO TO 110
160 LET A(X) = 1
170 PRINT: INK RND * 7: PAPER 9: "Otázka číslo": C:":
180 PRINT: "Ako je po anglicky "; TAB 4: E$: " ?"
190 INPUT A$
200 IF A$ = F$ THEN LET SKORE = SKORE + 1: BEEP.1,2.5 * SKORE:
PRINT FLASH 1: INK 2: PAPER 5: "áno, "; F$: " je správne"
210 IF A$ <> F$ THEN PRINT "Nie, ľutujem.", "Anglický výraz nie je ";
A$ ; TAB 8: ", ale je: "; F$
220 IF C < B THEN PRINT INK 3: FLASH 1: BRIGHT 1: "Vaše skóre je
teraz: "; SKORE: "
```

230 PAUSE 150

240 NEXT C

250 INK 2: PAPER 7: FLASH 1: CLS: PRINT AT 6,3: "výsledné skóre: ";

SKORE

260 PAUSE 200: FLASH 0

270 RUN

280 DATA "1. slovenský výraz", "anglický ekvivalent", 1,
"2. slovenský", "2. anglický", 2 ...atd. až "20. slovenský",
"20. anglický", 20

Slova píšete podľa možnosti veľkými písmenami.

Príklad na anglický slovník

Vitajte na angličtine!

Slovný test

Koliko výrazov si chcete preskúšať?

Zadajte počet: (1 - 20)

Otázka č. 1:

Ako je po anglicky deň?

DAY

Áno, DAY je správne

Vaše skóre je teraz 1

Otázka č. 2 atd.

Skúste použiť nasledujúci program 'majster kvízu' pre týždenné opakovania z odbornej učebnice, napríklad z biológie. Urobíme to jednoducho tak, že si najskôr obstaráme otázky. Otázky aj s možnosťami odpovedí zanesieme do programu a môžeme sa oddávať pohodlnému učeniu. Je to ľahšie ako si myslíte a nadaní študenti budú týmto spôsobom veľa pracovať.

Program je použitý pre uloženie a selekciu otázok alebo kľúčov (stop), spoločne s primeranými odpoveďami. Tu je niekoľko príkladov použitia:

Ako je po anglicky "dať" (odpoveď: "to give")

Meno minerálu dolovaného v St. Austellu? (odpoveď: čínska bridlica)

Ktorá mikroskopická (atómová) časť má jednotku hmoty a žiadne znamienko? (Odpoveď: neutrón)

atd.

Program najskôr zobrazí kompletnú sadu odpovedí krátko na obrazovke. Otázky sa prezentujú pomocou náhodných čísiel, ale bez opakovania. Ak je odpoveď zlá alebo jednoducho stlačíte ENTER, tak sa zobrazí ako pomôcka prvé a posledné písmeno ekvivalentu. Čiže máte dané "r....r" pre "rytlier" alebo "p.k" pre "psik".

RUN 500 opäť inicializuje celý program do pôvodného stavu a znovu zapojí nové otázky a odpovede.

5.6 M A J S T E R K V Í Z U

```

10 REM Inicializacia
20 LET S = 0: LET X = M: LET BS = "-----": LET ES = " "
30 FOR N = 1 TO M: LET Y(N) = N: NEXT N
40 GO SUB 400: CLS: FOR N = 1 TO M: PRINT AT 10,8;
  A$ (N): PAUSE 50: CLS: NEXT N
50 REM Hlavná časť programu
60 LET Z = INT (RND * X + 1): LET Q = Y(Z)
70 LET X = X - 1
80 FOR N = Z TO X: LET Y(N) = Y(N+1): NEXT N
90 CLS: LET T = 2
100 PRINT AT 3,0; C$ (Q): INPUT I$
110 IF I$ > A$ (Q) (TO L(Q)) THEN GO TO 240
120 LET S = S + 1: GO TO 210
130 GO SUB 410: CLS
140 IF X > 0 THEN GO TO 60
150 REM Konečná skóre
160 PRINT AT 5,0; "Vaše skóre bolo"; S; "mimo"; M * 2
170 STOP
200 REM Chválenie
210 PRINT AT 7,0; "Áno "; I$; " je správna odpoveď."; AT 9,15; E$
220 GO TO 130
230 REM Zlá odpoveď
240 IF I$ = " " THEN GO TO 260
245 PRINT AT 7,0; I$: "Zle "; E$: AT 9,0; E$
250 GO SUB 410
260 IF T = 2 THEN GO TO 310
270 PRINT AT 7,0; "Správna odpoveď: "; E$: AT 9,15; A$
280 GO TO 130
300 REM Pokyn - skús
310 LET T = 1
320 PRINT AT 7,0; "Tu je pomôcka: "; E$: AT 21,0; E$
330 PRINT AT 9,15; A$(Q,1) + B$(1 TO L(Q)-2) + A$(Q,L(Q))
340 GO TO 100
400 REM Vstup k pokračovaniu
410 PRINT AT 21,0; "Stlač ENTER."
420 GO SUB 460
430 IF CODE K$ = 13 THEN RETURN
440 GO TO 420
450 REM Vstup dvojitého kľúča
460 IF INKEY$ <> " " THEN GO TO 460
470 IF INKEY$ = " " THEN GO TO 470
480 LET K$ = INKEY$: RETURN
500 REM Vstup otázok a odpovedí
510 PRINT "Koľko otázok? "; INPUT M
520 DIM C$(M,64): DIM A$(M,15): DIM L(M): DIM Y(M)
530 FOR Q = 1 TO M
540 CLS: PRINT "Číslo otázky "; Q;
550 PRINT "Znenie otázky alebo kľúčové slovo ": INPUT C$(Q)
560 PRINT C$(Q) ' "Napiš príslušajúcu odpoveď ": INPUT I$
570 PRINT I$: AT 20,0; "Ak je to zadané správne S; ", "zrušit D"
580 GO SUB 460
590 IF K$ = "q" THEN GO TO 540
600 IF K$ <> "s" THEN GO TO 580

```

```

610 LET L(Q) = LEN I$: LET A$(Q) = I$
620 NEXT Q
650 REM SAVE a RUN
660 CLS: INPUT "aké je meno programu? "; P$
670 SAVE P$ LINE 20

```

Pre pochopenie ako tento program pracuje je najlepšie spustiť program najprv od riadku 500 príkazom RUN 500. Tu sa totiž zadávajú otázky a odpovede, ktoré budeme používať pri skúšaní. Kompletný počet otázok M, získaný na riadku 510 je použitý pre dimenzovanie poli na riadku 520. Pole C\$ si pamätá otázky, každú max. 64 znakov dlhú (dva riadky textu na počítači). Pole A\$ obsahuje odpovede, každú dlhú max. 15 znakov.

Ak chcete zadávať dlhšie otázky alebo odpovede, potom modifikujte parametre príkazov DIM. Pole L je použité pre podzranie skutočných dlžok odpovedí, takže príslušné retazce sú zapamätané v jednotlivých zložkách pola podľa svojej dĺžky, zbytok voľnej vyhradenej časti danej zložky pola nás nezaujima. Dĺžka odpovedí je použitá pri porovnávaní odpovede s odpoveďou správnu.

Odpovede sú počítačne vkladané použitím INPUT I\$, takže tieto dĺžky môžu byť spracované pred skladovaním do pola A\$. Riadky 660 a 670 nahrávajú celý program na magnetofón. Akonáhle máte zapamätané všetky otázky a odpovede v poli, musíte si zafixovať, že pri každom použití RUN sa polia aj premenné vyčistia a informácie by sa stratili. Preto pri návrate do programu použite miesto RUN príkaz GO TO 0. Tak máte zaručené, že zostane zachovaný obsah premenných.

Ak si prajete návrat do časti programu, ktorá už len skúša, musíte použiť GO TO 20 (a tým zachováte nedotknuté spomínané otázky i odpovede). Riadky 20 a 30 inicializujú premenné S je skóre a X je počet otázok, z ktorých sa bude skúšať. B\$ je použité ako oddelovač a E\$ je použité ako čistiaci znamienko pre riadok na obrazovke.

Riadok 40 zaisťuje zobrazenie všetkých odpovedí na obrazovke. užitočný posuvač pamäte pre kratšie sady otázok, ale tento sa vypúšťa, ak je počet otázok väčší. Priestor pola Y dimenzovaný v riadku 520 slúži k zabráneniu opakujúcich sa otázok, ak je program v chode. Princíp spočíva v tom, že do pola Y zapíšeme postupnosť všetkých X otázok v tvare Y(1) = 1, Y(2) = 2, ..., Y(X) = X. Keď náhodne vyberieme číslo Z, prečítame si z pola Y poločku Y(Z), ktorá určí číslo otázky. Aby sa otázky neopakovali, tak všetky prvky pola od Z+1 posunujeme o jedno dole (Y(Z)=Y(Z+1)). Tak sa zabudne staré Y(Z), a keď vygenerujeme opäť rovnaké Z, tak sa prečíta síce Y(Z), ale tu už máme číslo nasledovnej inej otázky Z+1. To je celý princíp ako vyberať vždy rôzne otázky bez opakovania.

Riadok 30 inicializuje pole tak, že Y(1) = 1, Y(2) = 2 atď. Ak je v teste len málo otázok (napr. štyri otázky), potom riadok 60 bude dávať hodnotu medzi 1 a 4. Predpokladáme, že vygenerujeme Z = 2: Číslo otázky Q je rovnaké ako Y(2) a je zatiaľ 2. Riadok 70 znižuje X, počet otázok. Riadok 80 eliminuje otázky, ktoré boli použité:

```

Pred riadkom 80      Po riadku 80
Y(1) = 1             Y(1) = 1
Y(2) = 2             Y(2) = 3
Y(3) = 3             Y(3) = 4
Y(4) = 4             Y(4) = 4

```

- táto hodnota sa presunula, ale nezmenila, čo nás už nezaujima, lebo najbližšie budeme vyberať

len z troch otázok Y(1) až Y(3).

Dva body održi skúšaná osoba, ak je odpoveď na otázku správna pri prvom pokuse, jeden bod za správnu odpoveď na druhý pokus. Riadok 110 je použitý na kontrolu správnosti odpovede. L(Q) je pomockou, aby sa porovnávali len retazce znakov rovnakej dĺžky.

Ak ste prvý raz odpovedali nesprávne, zobrazí sa náhoda. Riadok 330 totiž zobrazí prvé písmeno odpovede, prislúšný počet chýbajúcich písmen nahradí znakom '-' a na konci zobrazí posledné písmeno. Dĺžka správnej odpovede sa uchováva v L(Q), čo pomáha v programe zistiť, presne aký počet znakov sa má vynechať.

Môžete použiť tento program pre zdokonalenie Vašich detí v školskej výuke, ale i pre mnoho iných subjektov. Môžete nakrmiť počítač slovníkom francúzštiny, definíciami z chémie, otázkami z anglickej literatury a čoskoro si zostavíte skutočnú banku informačných materiálov, ktorú budete pravdepodobne i sam používať. Úvodne zobrazenie otázok pri spustení programu a pomôcka pri voľbe odpovedi vytvára nadväznosť a oživuje pamäť, keď spustíme program napríklad po roku. Toto bude dávať dôveru a väčšiu sebaistotu pri skúšaní.

Pozrime si ešte jednu aplikáciu výukového programu. Ide o tréning pamäte a zároveň test čítania. Skúšaná osoba si prečíta krátky text, ktorý sa náhodne vybral z určitého počtu fráz. Po krátkej fáze zapamätávania sa text z obrazovky zmaže a program sa opýta na presné znenie videneho textu. Zadáme text a snažíme sa aby ani v najmenšom nelišil od predtým videneho textu. Program výhodnoti naše snaženie. Ak je odpoveď nesprávna, tak sa text zobrazí znova a opakuje sa dovtedy, kým neodpovieme úplne správne.

Tento program je relatívne jednoduchý, ale ak doň vložíme nejaké zaujímavé frázy, tak sa môže stať aj celkom zaujímavý a pútavý. Bližšie sa zrokovanim jednotlivých riadkov zaoberať nebudeme, pretože sa už určite zorientujete v uvedenom algoritme aj bez pomoci. Tvorivé variácie sa môžu zamerať na menenie dĺžky zobrazenia textu - riadok 60, prípadne môžete kontrolovať, či ma odpoveď aspoň zodpovedajúcu dĺžku vzoru a pod.

5.7 T E S T Č Í T A N I A

```
10 INK 1: PAPER 7: BORDER 7: CLS
20 PAUSE 50: RESTORE
30 FOR J = 1 TO INT (RND * 11) + 1
40 BEEP 1 / J, 3 * J: READ A$: NEXT J
50 PRINT AT 10,0: A$
60 PAUSE (LEN A$) * 20
70 CLS
80 PRINT INK 2: "Vstup frázy, môžete teraz zadať text."
90 INPUT INVERSE 1: " "; B$; " "
100 PAUSE 50
110 IF A$ (2 TO) = B$ ( TO) THEN BEEP .3, LEN A$: PRINT INK 3:
    FLASH 1: "Dobre, je to správne": PAUSE 300: RUN
120 PRINT: "Lutujem, toto nie je správne."
130 PRINT: "Zopakujeme si to."
140 PAUSE 200: CLS
150 GO TO 50
160 DATA " Hľadajúci pes"
170 DATA " Mačka má štyri nohy"
180 DATA " Vajce je biele"
```

```
190 DATA " Slnko je horúce"
200 DATA " Liška je rýchla"
210 DATA " Kačka pláva pomaly"
220 DATA " Ľava má hrb"
230 DATA " Zvonček zvoní"
240 DATA " Ryba je mokrá"
250 DATA " Kniha je otvorená"
260 DATA " Autobus je červený"
```

Deti majú pôžitok z nasledujúceho programu - 'Kreslička'. Vytvárame s ním rôzne obrázky, siete máp a diagramov. Môžeme kresliť priame línie horizontálne, vertikálne alebo šikmo diagonálne používajúc nasledovné klávesy

```
T Y U
G H J
B N M
```

Vnútorná klávesa H v strede nemá funkciu. Poloha ostatných kláves určuje smer pohybu kresliaceho bodu.

5.8 K R E S L I Č K A

```
10 BORDER 6
20 INPUT "začiatok x súradníc? 0-255 "; X
30 INPUT "začiatok y súradníc? 0-175 "; Y
40 PLOT X,Y
70 LET C$ = INKEY$
80 LET X = X + (C$ = "u") + (C$ = "j") + (C$ = "m") - (C$ = "t") -
  (C$ = "g") - (C$ = "b")
90 LET Y = Y + (C$ = "t") + (C$ = "y") + (C$ = "u") - (C$ = "b") -
  (C$ = "m") - (C$ = "n")
100 LET X = X - (X > 255) + (X < 0)
110 LET Y = Y - (Y > 175) + (Y < 0)
120 IF C$ = "s" THEN STOP
130 GO TO 40
```

Napište a spustite tento program. Určte svoju štartovnú pozíciu pomocou súradníc X a Y bodu jemnej grafiky. Riadok 70 kontroluje a pamäta si v premennej C\$, ktorú klávesu ste stlačili.

Riadky 80 a 90 menia podľa toho súradnice X a Y. Ak stlačíte 'm' potom výraz (C\$ = "m") je pravdivý a získame hodnotu 1. Zbytok vo výraze na riadkoch 80 a 90 po stlačení 'm' nie je pravda a tak sa hodnoty ostatných výrazov rovnajú 0. Tak teda X = X + 0 + 0 + 1 - 0 - 0 = 0, stručne X=X+1, t.j. posun na osi X vpravo a podobne aj druhá premenná Y = Y + 0 + 0 + 0 - 0 - 0 = 0, v skratke Y = Y - 1, t.j. posun bodu na osi Y dole o 1 bod. Nasledujúci bod bude vykreslený na novej pozícii, juhovýchodne od prvého.

Riadky 100 a 110 zabraňujú tomu, aby bod kreslenia opustil plochu obrazovky. Riadok 120 preruší kreslenie náčrtku stlačením 's' klávesy. Potom môžete nahrat nakreslenú prácu pomocou SAVE "meno" SCREEN\$. Tak uložíte obsah obrazovky na pásku.

Jednoduchú narysovanú predlohu môžete vykresliť rôznymi spôsobmi. Môžete napr. prepísať riadok 120 takto:

120 IF C\$ = "f" THEN GO TO 140

Kontrola, či bolo stlačené 'f' prenesie po stlačení inkriminovanej klávesy beh programu na riadok 140. Tu potom môžete písať príkaz ako INPUT "Ink farba?"; ATRAMENT. Vpíšte príkaz INK ATRAMENT; do riadku 40 po príkaze PLOT a môžete kresliť rozmanitými farbami. Môžete zapojiť podobné príkazy z okruhu PAPER, OVER, FLASH, BRIGHT a INVERSE. Nezabudnite však premennú ATRAMENT nastaviť na začiatku programu na nejakú rozumnú hodnotu. Napr. 1 LET ATRAMENT = 0.

Získané výsledky môžete aplikovať v iných programoch, a tak ich celkom prepracovať ako na nasledujúcich príkladoch, ktoré ukazujeme:

Príklad:

Nakreslená mapa a k tomu body na mape.

Pracovná otázka: Aké sú mená týchto miest?

Aké je meno tohoto mesta?

Nakreslená mapa a zakreslené štáty Austrálie.

Pracovná otázka: Aké je číslo štátu Queensland?

Mapa Austrálie je samostatným programom, ktorý obsahuje aj jadro z kresličky modifikovanej vložením dvoch nasledujúcich riadkov:

50 IF INKEY\$ <> "" THEN GO TO 50

60 IF INKEY\$ = "" THEN GO TO 60

Pomalosť tohoto programu je zavinená tým, že sa kreslí len jeden bod v dobe, keď sa stlačí klávesa.

Potrebuje najst na mape presne pravú polohu pre správne kreslenie plôch na Vašej obrazovke. Trasuje mapové obrisy na útržok priehradnej plastiky a priložte ju na televíznu obrazovku. Použije sa tak program kopirujúci mapu na obrazovku. Ponechajte nejaký priestor na monitore pre otázky a odpovede.

Ak je Vaš náčrt kompletný, nahrajte ho. Môžete priložiť Vašu mapu používajúc priame povelý: CIRCLE 55,0,2, ktorý vyrobí kruhy pre mestá. Na skúšanie z mapy musíte zostaviť samostatný program. Polohu významných bodov nám pomôže zistiť kreslička. Upravíme ju tak, aby nám do rohu obrazovky zobrazovala momentálne súradnice X,Y, prebehne mapu a zapíšeme si potrebné súradnice miest Sydney, Melbourne atď. Zistené údaje vložíme do programu skúšanie zo zemepisu Austrálie a je to.

Môžete aplikovať rovnakú techniku k vytvoreniu takmer podobných programov používajúc kruhové diagramy vo fyzike, alebo premenu sekcií rastlinných tried v biológii.

Posledná úloha je zaviesť do Vašich map a diagramov program odpovedí a otázok. Ak chceme, dá sa použiť program 'majster kvízu', no potrebuje určitú modifikáciu. Premiestnenie v riadku 40:

40 LOAD "map" SCREEN\$

Budeme potrebovať priamo príkaz PRINT na spodnom okraji obrazovky. Nasledujúci podprogram môže pomôcť:

350 REM Vymaž spodné tri riadky
360 PRINT AT 19,0: E\$; E\$; E\$
370 PRINT AT 19,0
380 RETURN

Teraz vložte RUN 500 pre otázky a odpovede. Ak ich máte nahrané na páске, potom BREAK a nahrajte Vašu mapu. Nahrajte to použitím SAVE "mapa" SCREEN\$ na tu istu pásku. Bezprostredne potom nasledujúci program otázok a odpovedí.

Teraz, keď je nahraný "majster kvízu" bude automaticky nahraná mapa a ide sa k otázkam.

Ako vidíte, v tejto kapitole sme predviedli skupinu programov, ktoré študenti pravdepodobne radi použijú, a ktoré môžeme napísať bez dlhého investovaného času. Tieto programy spadajú do kategórie výcvikových a praktických cvičení. Tak efektívne učenie študenti potrebujú: hojnosť praxe, okamžitá istota, či sú odpovede správne alebo zlé, schopnosť zadať stupeň vývoja a veľmi dôležité, pochvalu, keď študent predviedol úspešný výkon.

Sú tu tiež iné námety na vyučovacie programy - napr: Viacnásobná výmena odpovedí, základné tvary a formy v geometrii, o ktorých nebolo diskutované. Tu predložené materiály sú dobrým východiskom pre vývoj všetkeho vlastného software.

Teraz ako prichádzame ku koncu tejto kapitoly, vlastnime nejaký počet jednoduchých, ale efektívnych programov pre výukové účely. Skôr ako na využijete tejto kolekcie spolu s obchodnými programami pozrite sa na nasledujúce body:

- neurýchľujte výkon pomocou niekoľkých prehustení obrazovky predbežným výpisom pri spúšťaní programu.

- materiál na obrazovke je nutné minimalizovať v čase a so zvýšenou pozornosťou sa zamerať na dôležité detaily.

- pri prezentácii textu použite definených farieb ako napríklad v programe 'delenie'

- ak to bude možné, rozdeľte skúšanie na čo najkonkrétnejšie otázky

- ak sa odpovedá zle, musí byť jasne ukázaná správna odpoveď

- vyvarujte sa stresových metód skúšania pomocou vyhradeného času na otázku, radšej nechajte skúšaného, nech rozmýšľa bez časového obmedzenia

- v konečnej dôsledku je dobrý program taký, ktorý je bez diskusie objektívny, a ktorý sa s potešením používa.

6.0 H R Á M E H R Y

Nie je látka, ktorá by sa netýkala Vašeho Spectra, ale je zrejme, že mnoho času strávite nad hraním hier. Je teda pravdepodobné, že akonáhle sa rozhodnete, že zrealizujete niektorý z veľkých trňakov vymyslených pre Vás počítač a pre vaše programy - hry alebo čokolke iné - vytvárate si prácu. V tejto kapitole spoznáte štruktúru niekoľkých hier. Velakrát v nich

budu použijete finty, ktoré vám spočiatku nebudu jasné. Niektoré vám aj napriek tomu môžu pomôcť, ak chcete meniť Vaše vlastné programy.

Ak chcete, môžete sa pokúsiť o jednoduchý vstup do konečných verzii hier bez čítania predchádzajúceho materiálu a listingu. Ale ak si nebudete istý s takýmto programom, na ktorom pracujete, tak ste minuli hlavný cieľ tejto kapitoly.

Skúste hotové programy krokovať a kontrolovať sám a sledovať riadok po riadku a v programoch, ktoré majú úvod s menu prejsť každú partiu, kým sa dostanete k samotnému prepisu programu do vlastného programu.

Dva prvé programy "Nočný prípad" a "Jack-mann" využívajú oveľa viac detailov než ostatné. Tieto čítajte určite starostlivejšie. Ved metódy z nich potom môžete aplikovať a využiť neskôr pre iné programy. Tu v týchto programoch sú myšlienky o ktorých si myslíme, že je dobré ich poznať, ale nie len im rozumieť ale vedieť ich aj aplikovať i na iné programové problémy.

6.1 N O Č N Ý P R Í P A D

Prvá hra obsahuje let lietadla nad mestom a redukciu oblaku nad mrakodrapom. Lietadlo letí horizontálne po obrazovke, klesá nižšie a pristáva na zemi. Eventuelne narazi do mrakodrapu, čomu sa vyhne iba tak, že najskôr zničí budovu.

Prvorádou úlohou je nakreslenie mrakodrapu. To je celkom ľahké ak sa použije slučka FOR-NEXT od 0 do 31 (všetky stĺpce na obrazovke) s vnorenou slučkou na vykreslenie steny budovy v každom stĺpci. Skúsme tento program:

```
10 FOR A = 0 TO 31
20 FOR B = 11 TO 21
30 PRINT AT B,A; "#"
40 NEXT B
50 NEXT A
```

Bude vykreslený celý rad mrakodrapov s rovnakej výškou na obrazovke. Skúste meniť interval pre B na riadku 20 pre variáciu výšky mrakodrapov. Tak, či onak, tento podprogram nevyrába veľmi zaujímavé obrýsové linie. Podobá sa to skôr dlhej stene než mestu. Preto potrebujeme pre zadanie výšok náhodné číslo. Skúste zmeniť riadok 20 takto:

```
20 FOR B = INT (RND * 22) TO 21
```

Obrýsová čiara mrakodrapov bude mať oveľa viac variačných priebehov. Tento podprogram vytvorenia mesta musí byť trochu pozmenený pre tvorbu viacerých miest. Prvý podprogram vytvárajúci mesto je uniformovaný, druhé mesto je príliš členité. Potrebujeme ohraničenie, v ktorom môže náhodný faktor pracovať. Skôr ako budeme faktor náhodný meniť, budeme ho kontrolovať. Ku koncu môžeme prirobiť tzv. 'obtiažnostný faktor'.

Postupne kreslíme presvedčivý obrýs mesta, potrebné kontrbaté výšky okolo budovaných mrakodrapov. Niektoré sú utle a tenké, niektoré sú väčšie, ale žiadny nie je veľmi tenký alebo veľmi veľký. V poslednom podprogramme má priemerná výška 11 znakov výšky, ale niektoré budovy majú 21 znakov výšok, zatiaľ čo niekde je len jedna výška. Obtiažnosť hry upresňuje priemer výšky budov na cca 18 znakov. Od tejto výšky plánujeme náraz do budovy. Ľahšia hra bude, keď je priemerná výška okolo 5. Plánujeme celkom dlhý čas, keď bude náraz do budovy. Samozrejme, že hra bude hľupa,

majúc rozsah výšok 5 a nájdú sa len 1 alebo 2 budovy, ktoré majú 20 znakov výšok, preto ľahká hra nebyva dlhá.

Na začiatku hry stanovíme stupeň obtiažnosti (napríklad rozsah od jednej do deväť). Meníme ju práve výškou mrakodrapov. Skúsme nasledujúci program:

Po vložení čísla 1 až 9 nakreslí Spectrum celkom realistické obrýsy budov. Skúste zadať nízke číslo a zbadáte, že mesto bude nižšie a pretiahnute. Všimnite si, že program prijme aj vstupné číslo mimo rozsah, ale potom zopakuje načítanie novej hodnoty. Prečo? Program bude totiž obsahovať podprogram odmietajúci chybný vstup.

```
1 REM Posledný program kreslenie mesta
10 INPUT " Aká má byť obtiažnosť? /1-9/", D
20 IF D<1 OR D>9 THEN GO TO 10
30 LET D = 12 - D
35 CLS
40 FOR A = 0 TO 31
50 FOR B = D + INT (RND * d) TO 21
60 PRINT AT B,A; "#"
70 NEXT B
80 NEXT A
90 GO TO 10
```

Ľudia často robia chyby, keď vkladajú informácie prostredníctvom klávesnice. A ak sa nevyvarujú omylu, môže chýbná vstupná hodnota stopnúť kompletne program alebo môže zmiast svojimi generovanými výsledkami. A jednoduchý riadok, ktorý kontroluje rozsah vystupujúcich čísel môže problém elegantne vyriešiť.

Ak už rozumiete ako program pracuje, je to v poriadku a smelo sa pozrime na ďalšiu partiu, ktorá vytvorí pohybujúce sa lietadlo, lietajúce nad mestom.

Najskôr zo všetkého si vytvoríme samotné lietadlo. Pátrajúc po klávesnici asi neobjavíme nejaký konkrétny symbol vhodného tvaru, znaky skôr vyzierajú neurčito ako konkrétne lietadlo. Pôjdeme na to prostredníctvom užívateľom definovanej grafiky - UDG. Tu je návod, ako na to:

1. Nakresli štvorcovú mriežku 8x8.

2. Do tejto mriežky zakreslite body, ktoré sa akosi podobajú na lietadlo z profilu. Ak sa vám nezdá, netrápte sa, na obrazovke to bude v poriadku.

No a teraz choďte po každej rade štvorcovej mriežky a pište do tabuľky nulu ak je štvorec prázdny a jednotku, ak je štvorec plný. Dostanete takúto tabuľku:

```
00000000
00000000
01100000
01111110
01111111
00001000
00010000
00100000
```


Každá rada nula a jedničky predstavuje vlastné binárne číslo.

3. Teraz choďte po jednotlivých riadkoch a pište PRINT BIN (séria nul a jednotiek v danom riadku). Zapište postupne čísla, ktoré takto dostanete. (Bude ich osem.)

Pre príklad uvádzame tretiu radu, ktorú zapíšete

```
PRINT BIN 01100000
```

a počítač bude mať na displeji 96. Napíšte toto pre porovnanie na kus papiera nasledovne:

```
0 1 1 0 0 0 0 0 = 96
```

neignorujte žiadnu okrajovú nulu ! Pretože napr.:

```
BIN 0 1 1 0 0 0 0 0 = 96, ale BIN 1 1 0 0 0 = 24 !
```

Budete mať teraz 8 čísel. Ak získate z uvedeného príkladu čísla 0, 0, 96, 126, 127, 8, 16 a 32 je to v poriadku.

Vymažte RAM použitím NEW a pište:

```
10 DATA (získaných osem čísel, oddelených čiarkou)
```

```
20 FOR A = 0 TO 7
```

```
30 READ B: POKE USR "A" + A, B
```

```
35 REM Netreba pripomínať, že A v úvodzovkách je napísané v G-režime
```

```
40 NEXT A
```

Riadok s DATA bude obsahovať Vašich 8 čísel oddelených čiarkami. Cyklus FOR-NEXT na riadkoch 20 až 40 načíta postupne týchto osem čísel príkazom READ a nadefinuje znak grafické A = CHR\$ 144 na tvar lietadla. Dokonca môžete zadať príkaz NEW a i tak zostane lietadielko na znaku A. Opätovné predefinovanie znaku A prevedieme tak, že zadáme patričné nové hodnoty do riadku za DATA. Ako týchto osem hodnôt získame, sme si vysvetlili pomocou PRINT BIN. Samozrejme, že nadefinovaná grafika nezostane v počítači večne. Po vykonaní NEW sa síce zachová, ale po odpojení zo siete sa inormácia stráca a zostáva nám štandardné A,B,C... podľa znaku, ktorý sme predefinovali.

Vybrojený aeroplánovým znakom môžeme teraz vytvoriť pohyb nad mestom a pohyb dole. Napíšte posledný zo spomínaných podprogramov 'kreslenie mesta' a pričítajte nasledujúce riadky

```
90 LET U = 0
```

```
100 LET P = 0
```

```
140 PRINT AT U,P: CHR$ 144
```

```
160 LET P = P + 1
```

```
170 PRINT AT U, P - 1: " "
```

```
180 IF P = 32 THEN LET P = 0: LET U = U + 1: BEEP .1,U
```

```
190 GO TO 140
```

Aktuálna verzia programu bude po spustení príkazom RUN žiadať odpoveď na stupeň obťažnosti a pri jej zavedení uvidíte ako sa zobrazí mesto. Potom budete vidieť malé lietadlo letiace naprieč obrazovkou a klesajúce

nižšie. 'Pipot', ktorý prichádza, značí stúpajúcu hladinu nebezpečenstva.

Lietadlo v tomto štádiu klesá až k dolnému okraju obrazovky a v momente, keď sa jeho súradnice dostanu mimo obrazovku, vypíše sa chybové hlásenie a program sa zastaví. Našou najhlavnejšou úlohou je preto vyriešiť dva problémy. Skontrolovať, či sa lietadlo nezrazilo s budovou a za druhé zostaviť podprogram imitujúci výbuch lietadla. Vďaka tomu, že budova je zostrojená pomocou písmen X, môžeme na kontrolu zrážky lietadla s budovou použiť SCREEN\$.

Úprava je nasledovná:

```
60 PRINT AT B,A: INVERSE 1: "X"
```

Na riadku 140:

```
140 IF SCREEN$(U,P) = "X" THEN STOP
```

SCREEN\$ môže čítať písmená a preto bude program zastavený, keď lietadlo narazí na "X".

Druhá metóda je nenahraditeľná vtedy, keď na zobrazenie budovy použijeme nejaké grafické znaky, nekontrolovateľné pomocou SCREEN\$. Vtedy nám pomôže ATTR, ale s tou podmienkou, že budovu farebne odlišíme od ostatných častí na obrazovke. Ako sme si spomínali, ATTR zisťuje farebnú informáciu o štvorcí veľkosti 1 textového znaku. Každá farebná kombinácia INKU, PAPERU, BRIGHTU a FLASHu určuje jednoznačné číslo, t.j. miesta s rôznymi farbami vracajú pre funkciu ATTR rôzne hodnoty, čo patrične využijeme.

Použitie ATTR funkcie - riadok 60:

```
60 PRINT AT b,a: INK 5: PAPER 0: "grafický blok"
```

Riadok 140:

```
140 IF ATTR (U,P) = 5 THEN STOP
```

Ak potrebujete zastaviť program, keď lietadlo zasiahne do žltej zóny, zmeňte riadok 140 takto:

```
140 IF ATTR (u,p) = 134 THEN STOP
```

Chcete vedieť, ako to pracuje?

```
128 ak je FLASH 1 - blikanie, 0 ak je FLASH 0 - bez blikania
```

```
1 x 128 = 128
```

```
64 ak je BRIGHT 1 - zvýšený jas, 0 ak je BRIGHT 0 - normálny jas
```

```
0 x 64 = 0
```

```
8 x farba podkladu PAPER (čierna = 0) 8 x 0 = 0
```

```
1 x farba textu INK (žltá = 6) 1 x 6 = 6
```

```
-----
```

```
sučet = 134
```

Okrem týchto dvoch metód je ľahšie použiť SCREEN\$ v tvare

```
PRINT AT b,a; INVERSE 1; "X"  
IF SCREEN$(u,p) = "X" THEN .....
```

A lietadlo bude prechádzať naprieč vzduchom a mrakodrapmi. Táto funkcia vyhovie vašim požiadavkám. Premiestnite riadok 140

```
140 IF SCREEN$(u,p) = "X" THEN STOP
```

pričítajte riadok 142

```
142 PRINT AT U,P; CHR$(144)
```

a nezabudnúť riadok 60

```
60 PRINT AT B,A; INVERSE 1; "X"
```

Program sa teraz zastaví, keď lietadlo vrazí do budovy.

STOP je núdné a nevzrušujúce. Čo potrebujete k vyvolaniu explózie, aby sa lietadlo roztrieštilo. Skúste nasledujúce:

```
200 LET X = P * 8; LET Y = (21-U) * 8  
205 FOR A = 1 TO 40  
207 PLOT X,Y  
210 DRAW INT (RND * 256) - X, INT (RND * 158) - Y  
220 BEEP .1,20; BEEP .01,10  
230 NEXT A
```

Označte bod na obrazovke pridelením súradníc pre u a p, hoci napríklad LET U = 11; LET P = 16. Vstup je teraz priamy. Teraz napíšte:

```
GO TO 200 a stlačte ENTER.
```

Uvidíte, že bod na obrazovke exploduje, doprevádzaný príslušným podfarbeným zvukom.

Spustíte pomocou RUN program celý a dostanete:

- a) mesto nakreslené na obrazovke
- b) lietadlo iduce nižšie a nižšie
- c) lietadlo zrazené s budovou a explóziou

Teraz si prečítajte program sám. Zistíte si bezpečne, že rozumiete ako pracuje. A na koniec pripojte záverečnú časť. Nesmieme totiž vynechať bombu, ktorú necháme padnúť z lietadla za účelom zničenia mrakodrapu, aby sme si tak vyčistili letovú dráhu a predĺžili let.

Skúsení profíci si samozrejme funkčnú verziu hry vysperkujú efektami aby sa hra stala čo najpriťažlivejšou.

Áby ste však nezapali na vavrinoch, pamätajte na to, že i tá najdokonalejšia verzia sa dá neustále obohacovať o nové myšlienky a rátať s tým, že niekto príde a ukáže vám ako možno vašu veľkú ideu ešte zdokonaľiť.

Ale vráťme sa ešte k pôvodnému programu, aby nikto nemohol povedať

že sme niečo rozrobili a nechali nedokoncené. Obohatime program o nenahradielnú 'bombu'. Pre to potrebujeme 'flag' (odbočenie).

Flag je indikátor nejakého stavu. Obvyčajne je daná hodnota 1 pre aktívny stav a 0 pre vypnutý - pasívny stav. V tomto programe bude flag nazvaný 'F' a bude rovný jednej keď je bomba vo vzduchu a 0 keď nie. Pre začiatok potrebujeme flag inicializovať, čiže nastaviť na nulu, pretože bomba nepada. Výsledok vyzereá nasledovne:

```
10 INPUT "obtiažnosť (1-9)"; D  
20 IF D < 1 OR D > 9 THEN GO TO 10  
30 LET D = 12 - D  
35 CLS  
40 FOR A = 0 TO 31  
50 FOR B = D + INT (RND * d) TO 21  
60 PRINT AT B,A; INVERSE 1; "X"  
70 NEXT B  
80 NEXT A  
90 LET U = 0  
100 LET P = 0  
110 LET F = 0  
140 IF SCREEN$(U,P) = "X" THEN GO TO 200  
142 PRINT AT U,P; CHR$(144)  
160 LET P = P + 1  
170 PRINT AT U,P-1; " "  
180 IF P = 32 THEN LET P = : LET U = U + 1: BEEP .1,U  
190 GO TO 140  
200 LET X = P * B: LET Y = (21-U) * B  
205 FOR A = 1 TO 40  
207 PLOT X,Y  
210 DRAW INT (RND * 256) - X, INT (RND * 158) - Y  
220 BEEP .1,20; BEEP .01,10  
230 NEXT A
```

Bomba je momentálne na palube lietadla - F = 0. Čo potrebujeme je zhodenie bomby a súčasná zmena indikátora na F = 1. Jednoducho pripojíme riadok:

```
148 IF INKEY$ <> " " THEN LET F = 1
```

Takto dosiahneme, že po stlačení ktorejkoľvek klávesy sa indikátor prepne do stavu F = 1, bomba je zhodená. Podmienka INKEY\$ <> " " je splnená vždy, keď stlačíme ktorejkoľvek kláves. Ak kláves nestlačíme, indikátor F svoju hodnotu nemení.

Nasledovná úprava spôsobí, že bombu zhodíme jedine stlačením klávesy nula.

```
148 IF INKEY$ = "0" THEN LET F = 1
```

Doporučujeme použiť 1. variant - zhodenie bomby ľubovoľnou klávesou, pretože sa tak ľahšie hrá. Nie je nutné si pamätať špeciálne klávesy.

Ďalej si ukážeme metódu ako pracovať s polohou bomby na obrazovke. Ak je bomba neodistená vo vnútri lietadla, v tom prípade klesá súhlasne s polohou lietadla.

144 IF F = 0 THEN LET A = U: LET T = P

(kde A a T sú súradnice bomby)

Ďalej budeme potrebovať predefinovaný znak, pomocou ktorého budeme bombu vykresľovať. Použijeme nasledovný tvar bomby:

```
0
0
0
0
32
28
32
0
0
```

Definovanie prebehne v cykle s príkazom

POKEUSR "B" + A,B

a nie

POKEUSR "A" + A,B

A tak získame plánovaný znak bomby.

Pristup k novému obrazu bomby je pomocou grafického B alebo cez znak CHR\$ 145. Ak preferujete jednoduché metódy, použijete plnú stopu pre bombu. To znamená, že bomba pri páde vytvorí stĺpec znakov, kým nedopadne a nezmižne. V takom prípade stačí vykresliť potrebný počet grafických B-čiek a nemusíme si robiť starosti s mazaním starej pozície bomby ako sme to robili pri pohybe lietadla.

S tvarom bomby je to už zariadené. Sme pripravení tlačiť. Ale musíme ošetriť alebo zohľadniť taký prípad, že bomba neopustila palubu lietadla. V takom prípade ju samozrejme zobrazovať nebudeme. Jej súradnice sa menia v závislosti od pohybu lietadla a v takomto prípade teda nebudeme nič s bombou meniť. Zaujímá nás však prípad, keď bomba letí vzduchom a to je vtedy, keď F = 0. Pre túto alternatívu zariadime zobrazenie bomby v jej súradniciach.

158 IF F = 1 THEN PRINT AT A,T; CHR\$ 145: BEEP .01,60-A

Bomba bude teraz vytlačena akonáhle stlačíte tlačítko. Nebude zatiaľ padať, ale pokiaľ pričítate nasledujúci riadok 155, napravíte spomínaný nedostatok. Okrem toho, že sa zmení súradnica bomby, zmaže sa predtým stará pozícia bomby aby vznikol dojem jednej padajúcej bomby.

155 IF F = 1 THEN PRINT AT A,T; " ": LET A = A + 1

Hoci teraz máme krásne padanie bomby, nie je to ešte úplné víťazstvo. Bomba totiž padá, padá, až klesne mimo obraz a vypíše sa chybové hlásenie.

Nutnosťou je ošetriť prípad, keď súradnice bomby opustia obrazovku a zároveň prípad, keď bomba zasiahne budovu.

157 IF SCREEN\$ (A,T) = "X" OR A = 21 THEN GO TO 300.

Teraz popíšeme podprogram pre zásah budovy bombou na riadkoch od 300

```
300 FOR A = A TO 21
310 IF RND > .99 THEN GO TO 340
315 BEEP .005, A -20
320 PRINT AT A,T; " "
330 NEXT A
340 LET F = 0: GO TO 140
```

Teraz ostáva doplniť posledný riadok a máme súvislú hru.

235 GO TO 10

Tu je kompletný listing tejto hry:

```
10 INPUT "obtiažnosť (1-9)", D
20 IF D < 1 OR D > 9 THEN GO TO 10
30 LET D = 12 - d
35 CLS
40 FOR A = 0 TO 31
50 FOR B = D + INT (RND * D) TO 21
60 PRINT AT B,A; INVERSEL; "X"
70 NEXT B
80 NEXT A
90 LET U = 0
100 LET P = 0
110 LET F = 0
140 IF SCREEN$ (U,P) = "X" THEN GO TO 200
142 PRINT AT U,P; CHR$ 144
144 IF F = 0 THEN LET A = U: LET T = P
148 IF INKEY$ <> " " THEN LET F = 1
153 IF F = 1 THEN PRINT AT A,T; " ": LET A = A + 1
157 IF SCREEN$ (A,T) = "X" OR A = 21 THEN GO TO 300
158 IF F = 1 THEN PRINT AT A,T; CHR$ 145: BEEP .01,60-A
160 LET P = P +1
170 PRINT AT U,P-1; " "
180 IF P = 32 THEN LET P = 0: LET P = U + 1: BEEP .1,U
190 GO TO 140
200 LET X = P * 8: LET Y (21 -U) * 8
205 FOR A = 1 TO 40
207 PLOT X,Y
210 DRAW INT (RND * 256) - X, INT (RND * 158) - U
220 BEEP .1,20 BEEP .01,10
230 NEXT A
235 GO TO 10
300 FOR A = A TO 21
310 IF RND > .99 THEN GO TO 340
315 BEEP .005, A-20
320 PRINT AT A,T; " "
330 NEXT A
340 LET F = 0: GO TO 140
```

Hra pracuje aj pri nedostatku niekoľkých vecí. V prezentovanej forme

nie je farba a hodnotenie úspešnosti. Skúste to síkonne napraviť (použite premennú S a pričítajte ju zakaždým, keď sa mrakodrap rozpadne pod bombou - v mieste okolo riadku 300).

Okrem toho budete potrebovať vytlačiť skóre na koniec hry. V konečnej verzii je na konci v tomto odstavci zahrnuté 'najväčšie skóre' ako hlavný bod. Toto sa jednoducho pričíta. Skúste pričítať niekoľko bajtov pre vylepšenie programu. Pripadne skúste definovať vašu vlastnú mrakodrapovú stenu s oknami pomocou UDG miesto obyčajného INVERSE "X".

Záverčná verzia má DATA príkazmi definované znaky pomocou užívateľom definovanej grafiky, takže môžete písať:

Záverčná farebná verzia Nočný nálet

```

2 LET H = 0
4 IF H = 0 THEN GO SUB 1000
5 LET S = 0
10 PAPER 0: INK 6: BORDER 1
30,PRINT AT 21,0: "obťažnosť (1-9)?:": LET D$ = INKEY$: IF LEN D$
<> 1 OR CODE D$ < 49 OR CODE D$ > 57 THEN GO TO 30
32 CLS
35 LET D = 12 - VAL D$
40 FOR A = 0 TO 31
50 FOR B = D + INT (RND * D) TO 21
60 PRINT PAPER 0: AT B,A: CHR$ 146
70 NEXT B
80 NEXT A
90 LET U = 0
100 LET P = 0
110 LET F = 0
140 PRINT AT U,P,: IF PEEK (PEEK 23684 + 256 * PEEK 23685) = 255
THEN GO TO 200
142 PRINT INK 4: CHR$ 144
144 IF F = 0 THEN LET A = U: LET T = P
148 IF INKEY$ <> "" THEN LET F = 1
155 IF F = 1 THEN PRINT AT A,T: " ": LET A = A + 1: IF A = 22 THEN
GO TO 340
157 PRINT AT A,T,: IF PEEK (PEEK 23684 + 256 * PEEK 23685) = 255
THEN GO TO 300
158 IF F = 1 THEN PRINT INK 2: CHR$ 145: BEEP .01,60-A
160 LET P = P + 1
170 PRINT AT U,P-1: " "
180 IF P = 32 THEN LET P = 0: LET U = U + 1: BEEP .1,U
185 IF U = 22 THEN GO TO 1200
190 GO TO 140
200 LET X = P * 8: LET Y = (21 - U) * 8
205 FOR A = 1 TO S STEP 10
207 PLOT X,Y
210 DRAW INK 6: INT (RND * 256) - X, INT (RND * 158) - Y
220 BEEP .1,20: BEEP .01,10
230 NEXT A
235 GO TO 400
300 FOR A = A TO 21
310 IF RND > .99 THEN GO TO 340

```

```

315 BEEP .005,A-20
320 PRINT AT A,T: " "
325 LET S = S + 1: IF S/250 = INT (S/250) THEN POKE 23624, PEEK
23624 + 8
330 NEXT A
340 LET F = 0: GO TO 140
400 IF H < S THEN LET H = S
410 PRINT AT 0,0: "skóre:":S, "naj-skóre:": H
420 GO TO 5
1000 DATA "A",0,0, 96,120,127,8,16,0,"B",0,0,0,32,28,32,0,0,
" C",255,153,153,255,255,153,153,255
1005 FOR F = 1 TO 3
1007 READ A$
1010 FOR A = 0 TO 7
1020 READ B: POKE USR A$ + A,B
1030 NEXT A
1045 NEXT F
1050 RETURN
1200 DATA 0,4,7
1205 RESTORE 1205
1210 FOR A = 1 TO 3
1215 READ C
1220 FOR B = 1 TO 3
1230 BEEP 1/3,C
1240 NEXT B
1250 NEXT A
1260 BEEP 1,12
1270 CLS: GO TO 40

```

Dúfame, že prevedenie a námet tejto hry Vám dá akúsi ideu ako vyvíjať nové, ešte pútavejšie a zaujímavejšie hry.

ZX SPECTRUM – BĀDATEĽ

Vydal: ULTRASOFT, spol. s r.o.
poštový priečinok, pošta 29
826 07 BRATISLAVA

V roku 1991 ako svoju 2. publikáciu
3. vydanie v roku 1993, náklad 1000 kusov

ISBN 80-85440-01-6