Cvičné úlohy pro jazyk Tiny BASIC (ver 1)

Příklad 1:

Zadání:
Vytvořte program pro výpočet plochy obdélníka. Po spuštění programu bude uživatel vyzván, ať zadá délky stran. Pak se vypíše velikost plochy. Vše počítejte s celočíselnými hodnotami.

Rada:
Použijte příkazy PRINT a INPUT. Plocha obdélníka se vypočítá jako součin délek jeho stran.

Příklad řešení:

```
0 REM Vypocet plochy obdelnika
1 PRINT “Zadej delku stran:\n“
2 INPUT A; INPUT B
3 PRINT “Plocha je: “ , A*B
```

Poznámky k řešení:
Příklad by se dal napsat i na jeden řádek:

```
1 INPUT A; INPUT B; PRINT “Plocha je: “ , A*B
```

Příklad 2:

Zadání:
Vytvořte program pro výpočet plochy kruhu s přesností na 2 desetinná místa. Po spuštění programu bude uživatel vyzván, ať zadá poloměr kruhu a pak se vypíše velikost plochy.

Rada:
Použijte příkazy PRINT, LET, IF-THEN a INPUT. Tiny BASIC počítá pouze s celými čísly. Použijte pro výpočet plochy hodnotu π jako 314 a pak zobrazte výsledek postupně. Nejprve vypište celou hodnotu získanou dělením 100, znak pro desetinnou čárku a pak zbytek po dělení 100. Nezapomeňte, že zbytek po dělení je nutné zobrazit i s první nulou – např. 08.
Vzorec pro výpočet plochy je známý:

$$S = \pi r^2$$

Příklad řešení:

```
0 REM Vypocet plochy kruhu
1 PRINT “Zadej polomer:\n“
2 INPUT R
3 LET S=R*R*314; LET C=S/100
4 PRINT “Plocha je: “ , C, “.“
5 LET Z=S-C*100
6 IF Z<10 THEN PRINT “0“
7 PRINT Z
```

Poznámky k řešení:
Tiny BASIC nemá operaci pro mocniny čísel. Proto je druhá mocnina zapsána jako R*R (řádek 3). A také nepracuje s desetinnými čísly. Násobením 314 místo 3,14 získáme výsledek 100krát větší. Abychom dostali celou část výsledku musíme S dělit 100 (řádek 3). Zbytek, který vypočítáme po odečtení celočíselné části násobené 100, pak musíme zobrazit i s případnou první nulou (řádek 6).

Příklad 3:

Zadání:
Vytvořte program, který pod sebe vypíše všechna lichá čísla do 20.

Rada:
Použijte příkazy PRINT, LET, GOTO a IF-THEN

Příklad řešení:

```
0 REM Licha cisla do 20
1 LET A=1
2 PRINT A, “\n“
3 LET A=A+2
4 IF A<20 THEN GOTO 2
```

Poznámky k řešení:
Jde o velmi jednoduchý počítaný cyklus. Začínáme od 1 (řádek 1) a postupně čísla vypisujeme na samostatné řádky (řádek 2) se zvětšováním po 2 (řádek 3). Vyhodnocovací podmínka (řádek 4) vrací program na výpis čísla, dokud není dosažena 20.

Rev. B

Příklad 4:

Zadání:
Vytvořte program, který vykresli na obrazovce pravoúhlý trojúhelník ze znaků ***** vysoký 10 řádků.

Rada:
Použijte příkazy PUTCH, LET, GOTO a IF-THEN. Znak pro nový řádek má hodnotu 10 a znak ***** má hodnotu 42.

Příklad řešení:

```
0 REM Trojuhelnik z *
1 PUTCH 10
2 LET A=1; REM Pocita radky
3 LET B=1; REM Hvezdy na radku
4 PUTCH 42; LET B=B+1
5 IF B<=A THEN GOTO 4
6 PUTCH 10; LET A=A+1
7 IF A<11 THEN GOTO 3
```

Poznámky k řešení:
Jsou zde použity 2 počítané cykly vnořené do sebe. V proměnné A se počítají řádky (řádky 2,6,7) a v proměnné B se počítají znaky vypisované na řádku (řádky 3,4,5).

Příklad 5:

Zadání:
Vytvořte program, který vykresli na obrazovce čtverec o délce strany 10 znaků.

Rada:
Použijte příkazy PUTCH, PRINT, LET, GOTO a IF-THEN. Abyste si ušetřili práci, použijte pro vykreslení vodorovné čáry podprogram s příkazy GOSUB a RETURN. Nezapomeňte program ve správném bodě ukončit příkazem END.
Znak pro vymazání displeje má hodnotu 12, nový řádek 10, levý horní roh 144, pravý horní roh 145, levý dolní roh 146, pravý dolní roh 147, svislá čára 156 a vodorovná čára 157.

Příklad řešení:

```
0 REM Ctverec
1 PUTCH 12
2 LET A=1; REM Pocita radky
3 PUTCH 144; GOSUB 10
4 PUTCH 145; PUTCH 10
5 PUTCH 156; PRINT “        “
6 PUTCH 156; PUTCH 10; LET A=A+1
7 IF A<8 THEN GOTO 5
8 PUTCH 146; GOSUB 10
9 PUTCH 147; PUTCH 10; END
10 REM Vodorovna cara
11 LET I=1
12 PUTCH 157; LET I=I+1
13 IF I<9 THEN GOTO 12
14 RETURN
```

Poznámky k řešení:
Řádky 3 a 4 vykreslí horní stranu čtverce, řádky 8 a 9 spodek čtverce. Využívá se podprogram na řádcích 10 až 14, který dělá vodorovnou čáru. Řádky 5, 6 a 7 vykreslí svislé čáry. Počet mezer v příkazu PRINT na řádku 5 musí být 8, aby se vyprázdnilo tělo čtverce.

Příklad 6:

Zadání:
Vytvořte program, který umožní posunovat znak **X** po obrazovce. Začíná se uprostřed obrazovky a stisknutí klávesy **2** znak **X** posune dolů, **4** vlevo, **6** vpravo a **8** nahoru.

Rada:
Použijte příkaz PUTCH s využitím jeho parametrů, CURSOR, INKEY, GOTO a IF-THEN.
Znak pro vymazání displeje má hodnotu 12, znak **X** je 88, prázdný znak je 32, **2** je 50, **4** je 52, **6** je 54 a **8** je 56.

Příklad řešení:

```
0 REM Posun X
1 PUTCH 12
2 CURSOR 400; PUTCH .88
3 INKEY K
4 IF K=50 THEN PUTCH v32; PUTCH .88
5 IF K=52 THEN PUTCH <32; PUTCH .88
6 IF K=54 THEN PUTCH >32; PUTCH .88
7 IF K=56 THEN PUTCH ^32; PUTCH .88
8 GOTO 3
```

Poznámky k řešení:
Program nejprve vyčistí obrazovku a umístí doprostřed displeje (pozice 400) znak X pomocí příkazu PUTCH. Symbol tečky zajistí, že kurzor zůstane na dané pozici. Když uživatel stiskne správnou klávesu, nejprve se vymaže stará pozice prázdným znakem a posune se kurzor na novou pozici (směr určuje symbol před hodnotou 32). Na nové pozici se pak vypíše X a ponechá kurzor.

Příklad 7:

Zadání:
Vytvořte program, který bude automaticky „hrát ping-pong“. Míček v podobě písmene **o** bude poletovat na řádku z levé strany na pravou stranu a naopak. Na kraji displeje se vždy odrazí a vrátí zpět.

Rada:
Použijte příkaz PUTCH s využitím parametrů, CURSOR, DELAY, GOTO a IF-THEN.
Znak pro vymazání displeje má hodnotu 12, znak **o** je 111, prázdný znak je 32.
Pozice uprostřed prostředního řádku má hodnotu 400, znak na levém okraji 384, vpravo 415.

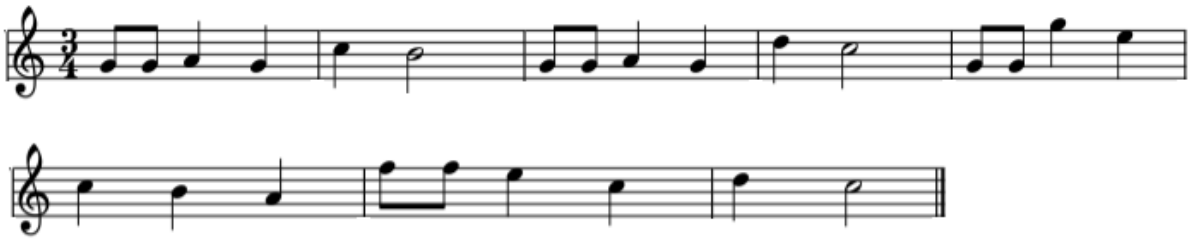
Příklad řešení:

```
0 REM Ping-pong
1 PUTCH 12; LET C=400
2 CURSOR C; PUTCH .111
3 LET D=1
4 DELAY 100; PUTCH .32
5 LET C=C+D; CURSOR C; PUTCH .111
6 IF C=415 THEN LET D=-1
7 IF C=384 THEN LET D=1
8 GOTO 4
```

Poznámky k řešení:
Nejprve se program nastaví do výchozí pozice a směr pohybu míčku v proměnné D se určí na hodnotu 1 – tj. míček poletí vpravo(řádky 1 až 3). Ve smyčce, která začíná na řádku 4, se chvíli počká, smaže starý obraz míčku a pak se posune míček podle hodnoty v D na novou pozici. Řádky 6 a 7 hlídají dosažení kraje obrazovky a mění směr letu tam (D=1), nebo zpět (D=-1).

Příklad 8:

Zadání:
Napište program, který zahraje melodii podle not na obrázku. Na začátku uživatel zadá dobu trvání 1/8 noty v rozmezí 200 až 400 milisekund.



Přibližné frekvence základních tónů:

c‘: 262	d‘: 294	e‘: 330	f‘: 349	g‘: 392	a‘: 440	h‘: 494
c“: 523	d“: 587	e“: 659	f“: 698	g“: 784	a“: 880	h“: 989

Rada:
Použijte příkaz BEEP s využitím parametrů, INPUT, GOTO a IF-THEN. Pokud se v melodii objevují stejné pasáže, můžete použít podprogram GOSUB a RETURN, abyste nemuseli opisovat stejné části. Nezapomeňte program ukončit příkazem END.

Příklad řešení:

```
0 REM Melodie
1 PRINT “Zadej tempo:\n“
2 INPUT T
3 IF T<200 THEN GOTO 1
4 IF T>400 THEN GOTO 1
5 GOSUB 15
6 BEEP 523,2*T; BEEP 494,4*T; GOSUB 15
7 BEEP 587,2*T; BEEP 523,4*T; BEEP 392,T
8 BEEP 392,T; BEEP 784,2*T; BEEP 659,2*T
9 BEEP 523,2*T; BEEP 494,2*T; BEEP 440,2*T
10 BEEP 698,T; BEEP 698,T; BEEP 659,2*T
11 BEEP 523,2*T; BEEP 587,2*T; BEEP 523,4*T
12 END
15 BEEP 392,T; BEEP 392,T; BEEP 440,2*T
16 BEEP 392,2*T; RETURN
```

Poznámky k řešení:
Uživatel je nejprve vyzván k zadání tempa a je zkontrolováno povolené rozmezí hodnot (řádky 1 až 4). Protože se první čtyři tóny opakují ve třetím taktu, je použit podprogram na řádcích 15 a 16. Jednotlivé tóny jsou zadány frekvencí a dobou trvání. Čtvrťová nota je dvojnásobná a půlová čtyřnásobná proti zadané době osminové noty.

Příklad 9:

Zadání:
Napište program pro vytvoření velmi jednoduchého klavíru. Po stisknutí klávesy **1** až **9** zahraje po dobu 200 až 600 milisekund (čas se zadá po startu) tón c‘ až d“.
Přibližné frekvence základních tónů:

c‘: 262	d‘: 294	e‘: 330	f‘: 349	g‘: 392	a‘: 440	h‘: 494
c“: 523	d“: 587					

Rada:
Použijte příkaz BEEP s využitím parametrů, INPUT, INKEY, GOTO a IF-THEN. Zkuste využít toho, že klávesy **1** až **9** vrací v příkazu INKEY hodnoty 49 až 57. Pak můžete vypočítat potřebný skok v příkazu GOTO.

Příklad řešení:

```
0 REM Piano
1 PRINT “Zadej dobu tonu:\n“
2 INPUT T
3 IF T<200 THEN GOTO 1
4 IF T>600 THEN GOTO 1
5 INKEY K
6 IF K<49 THEN GOTO 5
7 IF K>57 THEN GOTO 5
8 GOTO K-40
9 BEEP 262,T; GOTO 5
10 BEEP 294,T; GOTO 5
11 BEEP 330,T; GOTO 5
12 BEEP 349,T; GOTO 5
13 BEEP 392,T; GOTO 5
14 BEEP 440,T; GOTO 5
15 BEEP 494,T; GOTO 5
16 BEEP 523,T; GOTO 5
17 BEEP 587,T; GOTO 5
```

Poznámky k řešení:
Uživatel je nejprve vyzván k zadání doby tónu a je zkontrolováno povolené rozmezí hodnot (řádky 1 až 4). Ve smyčce, která začíná na řádku 5 se nejprve zkontroluje, zda klávesa patří mezi povolené a pak se vypočítá na řádku 8 skok na řádek s odpovídajícím tónem. Hodnota první povolené číslice 49 musí zamířit na řádek 9, proto se odečítá hodnota 40.

Příklad 10:

Zadání:
Vytvořte program, který bude představovat losovací zařízení pro čísla 1 až 49. Vždy po stisknutí libovolné klávesy zobrazí náhodné číslo z uvedeného rozsahu.

Rada:
Použijte příkazy TIME, INKEY, PRINT, GOTO a IF-THEN.
Po stisknutí klávesy vezměte aktuální čas. Jeho zbytek po dělení 49 bude dostatečně náhodné číslo. Nezapomeňte, že zbytek po dělení 49 je v rozsahu 0 až 48. Získanou hodnotu je potřeba zvýšit o 1.

Příklad řešení:

```
0 REM Nahodne cislo
1 INKEY K
2 IF K=0 THEN GOTO 1
3 TIME T; LET A=T/49
4 PRINT T-A*49+1, "\n"
5 GOTO 1
```

Poznámky k řešení:
Na řádcích 1 a 2 se čeká na stisknutí libovolné klávesy. V okamžiku, kdy uživatel stiskne klávesu se naplní proměnná T příkazem TIME. Hodnota získaná příkazem TIME se mění přibližně 32000krát za sekundu. Protože potřebujeme náhodná čísla do 49, vypočítáme zbytek po dělení 49. Výsledek zvětšený o 1 zobrazíme.

Příklad 11:

Zadání:
Vytvořte program ukazující čas uplynulý od spuštění počítače ve formě HH:MM:SS.

Rada:
Použijte příkazy TIME, PRINT, PUTCH, CURSOR, GOTO a IF-THEN.
Pokud hodnotu získanou příkazem TIME podělíte 31914, získáte čas od spuštění počítače v sekundách. Získaný čas pak stačí dělit 3600, abyste dostali počet uplynulých hodin. Ze zbytku pak dělením 60 získáte minuty. Nezapomeňte, že v zadání je zobrazení čísla na 2 místa. Nestačí vypsat jenom např. 5, je třeba zobrazit jako 05. Na to se může hodit podprogram s GOSUB a RETURN.
Znak **:** má hodnotu 58, znak **0** má hodnotu 48.

Příklad řešení:

```
0 REM Hodiny
1 PUTCH 12
2 DELAY 300
3 TIME T; LET S=T/31914
4 LET A=S/3600; LET S=S-A*3600
5 CURSOR 0; GOSUB 10
6 LET A=S/60; LET S=S-A*60
7 PUTCH 58; GOSUB 10
8 LET A=S; PUTCH 58; GOSUB 10
9 GOTO 2
10 REM Vypis z A na 2 mista
11 IF A<10 THEN PUTCH 48
12 PRINT A; RETURN
```

Poznámky k řešení:
Čas získaný příkazem TIME podělíme hodnotou 31914, abychom získali celkový čas v sekundách. Hodiny do proměnné A získáme dělením 3600. Ve zbytku po celočíselném dělení zůstanou minuty a sekundy. Celočíselným dělením 60 dostaneme minuty a zbytek z výpočtu minut jsou sekundy. Vše musíme zobrazovat na 2 místa, takže doplňujeme případnou první nulu u jednociferných čísel při zobrazování hodnoty z A v podprogramu (řádek 11).

Příklad 12:

Zadání:
Vytvořte pomocí počítače „kuchyňskou minutku“. Po startu programu uživatel zadá čas od 1 do 99 minut. Po zadání začne na displeji odpočítávání po minutách a po dosažení 0 se ozve 10 sekund signál.

Rada:
Použijte příkaz TIME, BEEP, INPUT, PRINT, GOTO a IF-THEN. Po zadání požadovaného času si můžete vypočítat čas, při kterém časování skončí. Cílový čas získáte přičtením počtu minut násobených 60*31914 (sekundy v minutě krát počet tiků za sekundu = 1914840) k aktuálnímu času.

Příklad řešení:

```
0 REM Minutka
1 PRINT “Zadej cas [min]:\n“
2 INPUT T
3 IF T<1 THEN GOTO 1
4 IF T>99 THEN GOTO 1
5 TIME C; LET C=C+T*60*31914
6 PUTCH 12
7 DELAY 300
8 TIME T; LET D=C-T
9 IF D<0 THEN BEEP 800,10000; END
10 CURSOR 0; PRINT D/1914840, “ “
11 GOTO 7
```

Poznámky k řešení:
Uživatel je nejprve vyzván k zadání doby a je zkontrolováno povolené rozmezí hodnot (řádky 1 až 4). Po správném zadání je do proměnné C uložen cílový čas, který vypočítáme jako aktuální čas s dobou zadanou uživatelem vynásobenou 31914 (počet tiků za sekundu) a 60 (počet sekund v minutě). Ve smyčce, která začíná na řádku 7, chvíli čekáme, protože není nutno zobrazovat hodnoty tak rychle, že by to uživatel stejně nepostřehl. Na řádku 8 zjistíme rozdíl mezi cílovým a aktuálním časem. Při záporném výsledku časování ukončíme. Jinak rozdíl zobrazíme. Konstanta 1914840 vznikne vynásobením 60*31914.

Rev. B

Příklad 13:

Zadání:
Naprogramujte hru „Lepší postřeh“ pro dva a více hráčů. Po stisknutí klávesy čeká 5 sekund, přičemž start časování je zahájen *****. Pak zobrazí na displeji cílový znak. Klávesa, která byla zmáčknutá jako první po zobrazení cílového znaku, bude označena jako vítěz.

Rada:
Použijte příkazy INKEY, DELAY, PUTCH, PRINT, GOTO a IF-THEN. Při tvorbě programu nezapomeňte na to, že musíte zkontrolovat, zda hráč nezmáčknul klávesu před zobrazením cílového znaku.
Znak ***** má hodnotu 42, cílový znak má hodnotu 143.

Příklad řešení:

```
0 REM Lepsi postreh
1 PRINT “\nStart hry\n“
2 INKEY K
3 IF K=0 THEN GOTO 2
4 PUTCH 42; DELAY 5000; INKEY K
5 PUTCH 143; IF K>0 THEN GOTO 10
6 INKEY K
7 IF K=0 THEN GOTO 6
8 PRINT “\nVitez: “; PUTCH K; DELAY 2000
9 GOTO 1
10 PRINT “\nProhral: “; PUTCH K; DELAY 2000
11 GOTO 1
```

Poznámky k řešení:
Na řádcích 1 až 3 počkáme na start hry. Pak zobrazíme *, čekáme 5 sekund a sejmeme, zda někdo nestiskl klávesu předčasně (řádek 4). Na řádku 5 zobrazíme cílový znak a zkontrolujeme předčasné stisknutí klávesy. Je-li vše v pořádku, na řádcích 6 a 7 čekáme na stisknutí klávesy a na řádku 8 vyhlásíme vítěze.

Příklad 14:

Zadání:
Vytvořte program pro akustický generátor. Uživatel zadá frekvenci požadovaného tónu v rozsahu 100 až 4000 Hz a dokud nebude program ukončen, počítač píská zadaným tónem.

Rada:
Použijte příkazy BEEP, INPUT, PRINT, GOTO a IF-THEN.
Aby bylo možné program zastavit zadáním příkazu END, použijte v časovém parametru u příkazu BEEP nějakou přijatelnou hodnotu v rozsahu jednotek sekund.

Příklad řešení:

```
0 REM Zvukovy generator
1 PRINT “Zadej frekv. 100-4000\n“
2 INPUT F
3 IF F<100 THEN GOTO 1
4 IF F>4000 THEN GOTO 1
5 BEEP F,5000
6 GOTO 5
```

Poznámky k řešení:
Uživatele nejprve vyzveme k zadání frekvence a je zkontrolováno povolené rozmezí hodnot (řádky 1 až 4). Pak je ve smyčce na řádcích 5 a 6 trvale pouštěn tón o zadané frekvenci.

Rev. B

Příklad 15:

Zadání:
Napište program, ve kterém uživatel zadá nějaké *jméno* ukončené klávesou Enter. Počítač pak vypíše, že *jméno* je jeho přítel. Maximální délka jména může být 30 znaků.

Rada:
Použijte příkazy INKEY, PRINT, PUTCH, GOTO a IF-THEN. Zadávané jméno je výhodné ukládat do proměnných v poli @. Počítejte si přitom počet uložených písmen.

Příklad řešení:

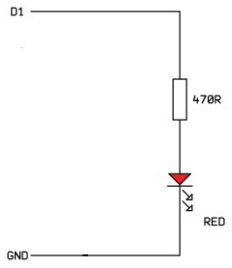
```
0 REM Pritel pocitace
1 LET I=0; PRINT “\nZadej jmeno:\n“
2 INKEY K
3 IF K=0 THEN GOTO 2
4 IF K=10 THEN GOTO 10
5 PUTCH K; LET @I=K; LET I=I+1
6 IF I=30 THEN GOTO 10
7 GOTO 2
10 LET J=0; PUTCH 10
11 PUTCH @J; LET J=J+1
12 IF J<I THEN GOTO 11
13 PRINT “ je muj pritel\n“
```

Poznámky k řešení:
Řádky 2 a 3 čekají na zadání znaku. Řádek 4 kontroluje, zda nebylo zadávání ukončeno klávesou Enter. Na řádku 5 je znak uložen do pole. Řádek 6 kontroluje maximální povolenou délku jména. Na řádcích 10 až 12 je jméno opět vypsáno.

Příklad 16:

Zadání:
Vytvořte program pro ovládání LED připojené k výstupu D1 pomocí klávesnice. První zmáčknutí libovolné klávesy LED rozsvítí, druhé zhasne, další opět rozsvítí atd.

Doplňkový HW:
Na výstup D1 připojte LED s rezistorem podle následujícího obrázku. Červená barva LED bude vhodná pro následující úlohu.



Rada:
Použijte příkazy INKEY, DOUT, GOTO a IF-THEN. Příkaz DOUT 1,1 pak LED rozsvítí a DOUT 0,1 ji zhasne.
Protože program nemusí být rychlý, můžete do cyklu čekání na stisk klávesy vložit drobné zpoždění příkazem DELAY.

Příklad řešení:

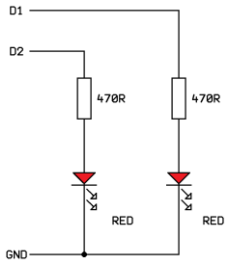
```
0 REM Ovladani LED
1 LET L=0
2 DELAY 100; INKEY K
3 IF K=0 THEN GOTO 2
4 IF L=0 THEN GOTO 6
5 LET L=0; DOUT 0,1; GOTO 2
6 LET L=1; DOUT 1,1; GOTO 2
```

Poznámky k řešení:
Proměnná L ukládá aktuální hodnotu výstupu, abychom věděli, jak příště výstup změnit. Na řádcích 2 a 3 se čeká na stisk klávesy. Řádek 4 rozhodne podle stavu L, zda se LED rozsvítí nebo zhasne.

Příklad 17:

Zadání:
Vytvořte program, který z TinyBasRV udělá ovládání výstražných světel na železničním přejezdu i se zvukovou signalizací.

Doplňkový HW:
Na výstupy D1 a D2 připojte dvě LED (nejlépe červené) s rezistory podle následujícího obrázku.



Rada:
Použijte příkazy BEEP, DELAY, DOUT a GOTO. Zvukovou signalizaci si vyzkoušejte a upravte podle vlastních představ.

Příklad řešení:
```
0 REM Zeleznice - vystraha
1 DOUT 1,1; DOUT 0,2
2 BEEP 2000,100; BEEP 800,200; DELAY 200
3 DOUT 0,1; DOUT 1,2
4 BEEP 2000,100; BEEP 800,200; DELAY 200
5 GOTO 1
```

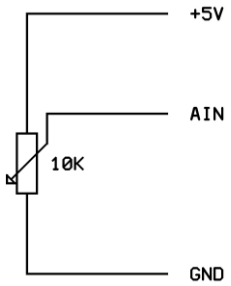
Poznámky k řešení:
Řádky 1 a 3 rozsvítí vždy pouze jednu z obou LED. Na řádcích 2 a 4 je generováno stejné zvukové znamení. Uvedený příklad jen velmi vzdáleně připomíná cinkání zvonce. Můžete je ladit a upravovat podle svého ucha...

Rev. B

Příklad 18:

Zadání:
Vytvořte program pro měření vstupního napětí na vstupu AIN v mV. Na vstup přiveďte napětí z běžce potenciometru, které bude zobrazováno s periodou ½ sekundy.

Doplňkový HW:
Na výstup AIN připojte lineární potenciometr s hodnotou odporu v rozmezí 1 až 100 kΩ podle následujícího obrázku.



Rada:
Použijte příkazy AINP, DELAY, CURSOR, PRINT a GOTO. Hodnota získaná z A/D převodníku je v rozsahu 0 až 1023. To představuje napětí 0 až 5V – tedy 0 až 5000 mV. Abychom přepočítali rozsah A/D převodníku na mV, musíme hodnotu nejprve vynásobit 5000 a pak dělit 1023.

Příklad řešení:

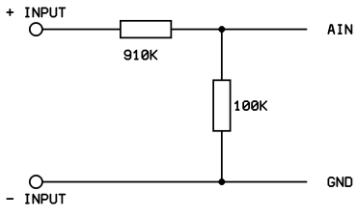
```
0 REM Analog vstup mV
1 PUTCH 12
2 CURSOR 397; AINP A
3 PRINT A*5000/1023, “ mV   “
4 DELAY 500
5 GOTO 2
```

Poznámky k řešení:
Na řádku 2 se nejprve nastaví kurzor přibližně do středu obrazovky a pak se změří napětí na vstupu AIN. Přepočet se provede při výpisu hodnoty v příkaze PRINT podle návodu v odstavci **Rada**. Po výpisu jednotek jsou ještě doplněny 3 mezery, aby se vymazal případný delší zápis z minulého měření.

Příklad 19:

Zadání:
Upravte počítač TinyBasRV tak, aby pracoval jako jednoduchý voltmetr s rozsahem do 50 V. Napětí bude zobrazováno s periodou ½ sekundy a s rozlišením na 2 desetinná místa.

Doplňkový HW:
Na výstup AIN připojte vstupní odporový dělič s rezistory s poměrem odporů 9:1. Ideální by bylo např. 900 kΩ a 100 kΩ. Hodnota 900 kΩ je ale mimo běžnou řadu dostupných rezistorů. Můžete použít nejbližší hodnotu 910 kΩ z řady E24. V nouzi lze použít i 1 MΩ a upravit konstantu v programu.



Rada:
Použijte příkazy AINP, DELAY, CURSOR, PUTCH, PRINT, GOTO a IF-THEN. Hodnota získaná AINP z A/D převodníku je v rozsahu 0 až 1023. To představuje napětí 0 až 50V (musíme zobrazit 0.00 až 50.00 V). Abychom přepočítali rozsah A/D převodníku na V, je třeba hodnotu nejprve vynásobit 5000 a pak dělit 1023. Výsledné číslo zobrazíme s desetinou tečkou mezi číslicemi. Násobící konstanta 5000 je ideální případ, kdy se předpokládá přesné referenční napětí v mikropočítači a přesné hodnoty odporu obou rezistorů v děliči. V praxi se to bude lišit. Proto je potřeba u konkrétního kusu počítače tuto hodnotu upravit tak, aby zobrazené napětí se co nejvíce blížilo napětí změřenému referenčním voltmetrem.

Příklad řešení:
```
0 REM Voltmetr 50V
1 PUTCH 12; LET K=5320
2 CURSOR 397; AINP A
3 LET V=A*K/1023; LET C=V/100
4 LET Z=V-C*100
5 PRINT C, “.“
6 IF Z<10 THEN PRINT “0“
7 PRINT Z
8 DELAY 500
9 GOTO 2
```

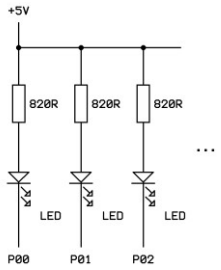
Poznámky k řešení:
Na řádku 1 je nastavena přepočtová konstanta, kterou si musíte upravit podle konkrétního kusu počítače a vstupního odporového děliče. Další postup již kopíruje instrukce uvedené v odstavci **Rada**. Nejprve je vypsána celá část z proměnné C a pak zbytek po dělení 100 ze Z. Pokud je zbytek menší než 10, je nutné doplnit úvodní nulu (řádek 6).

Rev. B

Příklad 20:

Zadání:
Pomocí expanderu PCF8575 připojeného k TinyBasRV vyrobte běžící světlo na řadě LED připojených k výstupům expanderu.

Doplňkový HW:
Na výstupy modulu s expanderem PSC8575 připojte více LED podle obrázku. Záleží na množství LED, které máte k dispozici, maximálně je možné připojit 16 LED.



Rada:
Použijte příkazy I2CW, DELAY, LET, GOTO a IF-THEN. Pokud chcete rozsvítit první LED, musíte do expanderu poslat binární číslo b1111111111111110 nebo dekadicky -2. Druhá LED vyžaduje poslat binární číslo b1111111111111101 nebo dekadicky -3. Další by bylo binární číslo b1111111111111011 nebo dekadicky -5. Rozsvícení LED se posune na další pozici vždy vynásobením 2 a přičtením 1.
Po rozsvícení poslední LED se musí světlo vrátit na začátek.

Příklad řešení (pro 8 LED):

```
0 REM Bezicich 8 LED
1 LET A=-2
2 I2CW A; DELAY 200
3 LET A=A*2+1
4 IF A < -300 THEN LET A=-2
5 GOTO 2
```

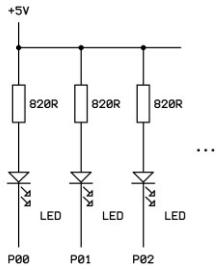
Poznámky k řešení:
Program postupuje podle instrukcí uvedených v odstavci **Rada**. Po daný příklad bylo použito pouze 8 LED, při dosažení hodnoty menší než -300 se hodnota v A vrátí na začátek. Poslání čísla -257 do expanderu (poslední odeslané v cyklu) na chvíli vypne všech 8 LED. Pokud byste to nechtěli, zmenšete podmínku např. na A<-150.

Rev. B

Příklad 21:

Zadání:
Napište program, který pomocí expanderu PCF8575 připojeného k TinyBasRV rozsvítí po stisknutí klávesy **0** až **9** určený počet LED (vámi zvolený obraz).

Doplňkový HW:
Na výstupy modulu s expanderem PSC8575 připojte více LED podle obrázku. Záleží na množství LED, které máte k dispozici, maximálně je možné připojit 16 LED.



Rada:
Použijte příkazy I2CW, DELAY, INKEY, GOTO a IF-THEN. Pro jednotlivé číslice si můžete určit libovolný vzorek rozsvícených LED. Např. po stisknutí klávesy **1** se rozsvítí každá lichá LED příkazem I2CW b1010101010101010, klávesa **2** rozsvítí dolní polovinu LED příkazem I2CW b1111111100000000 atd.
Klávesy **0** až **9** vrací v příkazu INKEY hodnoty 48 až 57.

Příklad řešení:

```
0 REM Vzory z LED
1 DELAY 100; INKEY K
2 IF K<48 THEN GOTO 1
3 IF K>57 THEN GOTO 1
4 GOTO K-43
5 I2CW b1100110011001100; GOTO 1
6 I2CW b1010101010101010; GOTO 1
7 I2CW b1111111100000000; GOTO 1
8 I2CW b1111100001100101; GOTO 1
9 I2CW b0000000011001100; GOTO 1
10 I2CW b1100110000000000; GOTO 1
11 I2CW b1110111011101110; GOTO 1
12 I2CW b1101110111011101; GOTO 1
13 I2CW b1011101110111011; GOTO 1
14 I2CW b0111011101110111; GOTO 1
```

Poznámky k řešení:
Nejprve se na řádcích 1, 2 a 3 zajistí přijetí pouze povolené číselné klávesy. Řádek 4 pak zajistí odskok na patřičný vzorek rozsvícených LED s návratem na začátek.

Příklad 22:

Zadání:
Naprogramujte měřič postřehu uživatele. Po stisknutí klávesy čeká program náhodný počet sekund (1 až 10), přičemž start časování je zahájen *****. Pak zobrazí na displeji cílový znak. Uživatel musí co nejrychleji zmáčknout libovolnou klávesu. Na displeji se zobrazí reakční čas v milisekundách.

Rada:
Použijte příkazy INKEY, TIME, DELAY, PUTCH, PRINT, GOTO a IF-THEN. Při tvorbě programu nezapomeňte na to, že musíte zkontrolovat, zda hráč nezmáčknul klávesu před zobrazením cílového znaku. Pro měření času v milisekundách budeme počítat, že TIME vrací každou milisekundu hodnotu o 32 větší.
Znak ***** má hodnotu 42, cílový znak má hodnotu 143.

Příklad řešení:

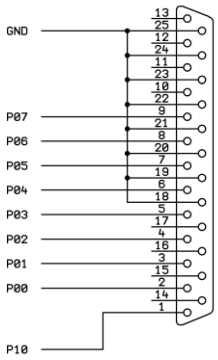
```
0 REM Meric postrehu
1 PRINT "\nStart\n"
2 INKEY K
3 IF K=0 THEN GOTO 2
4 TIME T; LET C=T/10
5 LET D=T-C*10+1; LET D=D*1000
6 PUTCH 42; DELAY D
7 INKEY K; PUTCH 143; TIME S
8 IF K>0 THEN GOTO 14
9 INKEY K
10 IF K=0 THEN GOTO 9
11 TIME E; LET R=E-S
12 PRINT "\nReakce: ", R/32
13 PRINT " milisec"; GOTO 1
14 PRINT "\nPredcasne!"
15 GOTO 1
```

Poznámky k řešení:
Řádky 2 a 3 čekají na start měření. Pak se získá pomocí příkazu TIME a zbytku po dělení 10 náhodný čas, který se uloží do proměnné D v milisekundách (řádky 4 a 5). Na řádku 6 se zobrazí * a provede čekání. Řádek 7 sejme stav kláves, zobrazí cílový znak a uloží začáteční čas pro měření postřehu. Na řádku 8 se zkontroluje případné předčasné stisknutí klávesy. Řádky 9 a 10 čekají na reakci uživatele. Čas, kdy zareagoval se uloží do E a pak na řádcích 11, 12 a 13 se výsledek vypíše.

Příklad 23:

Zadání:
Máte-li možnost připojit k TinyBasRV jehličkovou tiskárnu s paralelním portem přes expander PCF8575, vytvořte z ní jednoduchý psací stroj.

Doplňkový HW:
Na výstupy modulu s expanderem PCF8575 připojte konektor Cannon 25 samici podle obrázku.



Rada:
Použijte příkazy I2CW, DELAY, INKEY, GOTO a BEEP. Příkaz BEEP se použije pro zvukovou signalizaci stisknutí klávesy krátkým pípnutím. Není nutné jej použít.
Na vývodu 1 paralelního portu je signál STROBE ovládaný výstupem P10, který potvrdí a zapíše data do tiskárny. Aktivní hodnota je logická 0. Po zapsání dat, je nutné tento signál vrátit do klidové hodnoty 1. To lze nejsnáze udělat zapsáním hodnoty o 256 větší než zapisovaná data.

Příklad řešení:

```
0 REM Psaci stroj
1 INKEY K
2 IF K=0 THEN GOTO 1
3 I2CW K; PUTCH K
4 DELAY 10
5 I2CW K+256; BEEP 1000,50
6 GOTO 1
```

Poznámky k řešení:
Řádky 1 a 2 čekají na stisk klávesy. Řádek 3 pošle znak do tiskárny a okopíruje ho na displej. Na řádku 4 se chvíli počká (pro případ pomalé reakce tiskárny). Na řádku 5 se musí nastavit signál STROBE pro tiskárnu zpět na logickou 1 (hodnota znaku se zvýší o 9tý bit nastavený na 1) a krátce se pípne.

Příklad 24:

Zadání:
Vytvořte z TinyBasRV jednoduchou celočíselnou kalkulačku. Uživatel zadá první číslo, pak operand (+ - * /) a na závěr druhé číslo. Na displeji se objeví vypočítaný výsledek ve tvaru např. 256*2=512.

Rada:
Použijte příkazy INPUT, INKEY, PRINT, PUTCH, GOTO a IF-THEN. Znaky operandů mají následující hodnoty:
+ 43 - 45 ***** 42 / 47
Nezapomeňte, že nulou nedělíme!

Příklad řešení:

```
0 REM Kalkulacka
1 PUTCH 10; INPUT A
2 INKEY K
3 IF K=42 THEN GOTO 8
4 IF K=43 THEN GOTO 8
5 IF K=45 THEN GOTO 8
6 IF K=47 THEN GOTO 8
7 GOTO 2
8 PUTCH K; PUTCH 10; INPUT B;
9 PRINT A; PUTCH K; PRINT B
10 IF K=43 THEN PRINT "=",A+B; GOTO 1
11 IF K=45 THEN PRINT "=",A-B; GOTO 1
12 IF K=42 THEN PRINT "=",A*B; GOTO 1
13 IF B=0 THEN PRINT " Nulou nelze!"; GOTO 1
14 PRINT "=",A/B; GOTO 1
```

Poznámky k řešení:
První hodnota se uloží do proměnné A na řádku 1. Pak se na řádcích 3 až 7 čeká na typ operace. Na řádku 8 vytiskneme znak operace a počkáme na druhou hodnotu. Řádky 10 až 14 provedou výpočet podle stavu K, přičemž se na řádku 13 zkontroluje, zda se nebude dělit 0.

Příklad 25:

Zadání:
Napište program pro hádání tajného čísla 0 až 99. Po startu si počítač vybere náhodné číslo a hráč má toto číslo uhodnout na co nejmenší počet pokusů. Po každém pokusu počítač uvede, zda je tajné číslo větší nebo menší než to, které zadal hráč.

Rada:
Použijte příkazy INPUT, PRINT, TIME, LET, GOTO a IF-THEN. Nejprve si vygenerujte pomocí příkazu TIME a zbytku po dělení 100 náhodné číslo.

Příklad řešení:

```
0 REM Hadani cisla
1 TIME T; LET C=T/100; LET X=T-C*100
2 LET P=0; PRINT "\nHadej cislo\n"
3 INPUT A; LET P=P+1
4 IF A=X THEN GOTO 8
5 IF A>X THEN GOTO 7
6 PRINT P,". Hadej vic\n"; GOTO 3
7 PRINT P,". Hadej mene\n"; GOTO 3
8 PRINT "Uhodl jsi na ", P, " pokusu\n"
```

Poznámky k řešení:
Na řádku 1 se pomocí příkazu TIME a zbytku po celočíselném dělení 100 uloží do proměnné X náhodně generované hádané číslo. Na řádcích 2 a 3 je uživatel vyzván k pokusu uhodnout číslo a v proměnné P se počítají provedené pokusy. Řádky 4 a 5 kontrolují úspěšnost pokusu a na zbývajících řádcích se vypisuje pokyn, kam směřovat další odhad nebo ukončení programu s výsledkem.

Příklad 26:

Zadání:
Vytvořte z TinyBasRV jednoduchý kódový zámek. Zadá-li uživatel na klávesnici správnou posloupnost 4 nebo 5 číslic, na výstupu D1 se na 3 sekundy objeví logická 1 a zazní zvukový signál.

Rada:
Použijte příkazy INKEY, LET, DOUT, BEEP, GOTO a IF-THEN. Počet kódových číslic si můžete zvolit sami. Tajnou číselnou kombinaci si uložte do pole @ a každá správně uhodnutá číslice posune ukazatel v poli na vyšší hodnotu. Při chybě se ukazatel vrátí na počátek pole.
Klávesy **0** až **9** vrací v příkazu INKEY hodnoty 48 až 57.

Příklad řešení:

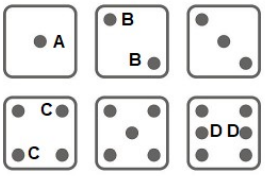
```
0 REM Kodovy zamek
1 LET @0=49,57,50,48,51
2 LET P=0
3 INKEY K
4 IF K=0 THEN GOTO 3
5 IF K<>@P THEN GOTO 2
6 LET P=P+1;
7 IF P<5 THEN GOTO 3
8 DOUT 1,1; BEEP 800,3000; DOUT 0,1
9 GOTO 2
```

Poznámky k řešení:
Na řádku 1 je do pole @ uložen kód zámku. Proměnná P ukazuje, kam se uživatel při zadávání kódu dostal. Řádek 5 rozhoduje, zda se postoupí na další pozici, nebo se vrátí opět na první číslici kódu. Na řádku 7 se při správném postupu a dosažení ukazatele P na hodnotu 5 umožní přejít na řádek 8, kde se kódový zámek na 3 sekundy otevře.

Příklad 27:

Zadání:
Napište program, který bude nahrazovat hrací kostku. Program bude na displeji zobrazovat body tak, jak se nacházejí na kostce.

Rada:
Pokud se podíváte na vzory nacházející se na stěnách kostky, zjistíte, že kromě střední tečky se tečky vždy vyskytují po dvojicích. Např. 3 je kombinace A+B, 5 je A+B+C.



Vzory si můžete zakódovat do pole např. pomocí binárních čísel a pak je podle náhodně taženého čísla zobrazit. Doporučené pozice kurzoru jsou pro A: 400, pro B: 235 a 565, pro C: 245 a 555 a pro D: 395 a 405. Použijte příkazy LET, TIME, INKEY, GOTO, CURSOR, PUTCH a IF-THEN.

Příklad řešení:

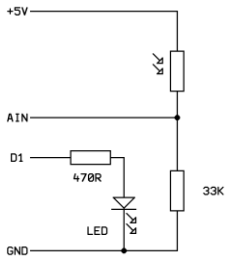
```
0 REM Hraci kostka
1 PUTCH 12; LET @0=1,2,3,6,7,14
2 INKEY K
3 IF K=0 THEN GOTO 2
4 TIME D; LET C=D/6; LET D=D-C*6
5 CURSOR 400; PUTCH .32; IF @D&1 THEN PUTCH 143
6 CURSOR 235; PUTCH 32; CURSOR 565
7 PUTCH .32; IF@D&2 THEN PUTCH 143
8 CURSOR 235; IF@D&2 THEN PUTCH 143
9 CURSOR 245; PUTCH 32; CURSOR 555
10 PUTCH .32; IF@D&4 THEN PUTCH 143
11 CURSOR 245; IF@D&4 THEN PUTCH 143
12 CURSOR 395; PUTCH 32; CURSOR 405
13 PUTCH .32; IF@D&8 THEN PUTCH 143
14 CURSOR 395; IF@D&8 THEN PUTCH 143
15 GOTO 2
```

Poznámky k řešení:
V poli @ je zakódováno zobrazení čísel 0 až 5 pomocí binárního kódu. Pokud se podíváte, jak se uvedená čísla zapisují ve dvojkové soustavě, zjistíte, že nejnižší bit představuje tečku A, vyšší bit obě tečky B, další C a čtvrtý bit jsou tečky D. Řádky 2 a 3 čekají na stisknutí klávesy. Na řádku 4 je vygenerováno do proměnné D náhodné číslo 0 až 5. Pak jsou procházeny všechny pozice jednotlivých teček. Např. u řádku 5 je kurzor nastaven na tečku A a ta je nejprve smazána. Pokud prvek pole s indexem D má nastavený nejnižší bit, je tečka na pozici vykreslena.

Příklad 28:

Zadání:
Doplňte TinyBasRV tak, aby fungoval jako řídicí panel nouzového osvětlení. Pomocí fotorezistoru se bude snímat stav okolního osvětlení a měřená hodnota se zobrazí na displeji. Uživatel zadá na začátku hodnoty při kterých se zapne a naopak vypne LED připojená na D1.

Doplňkový HW:
Na vstup AIN a výstup D1 připojte dělič s fotorezistorem a LED s rezistorem podle obrázku.



Rada:
Použijte příkazy INPUT, PRINT, DOUT, AINP, GOTO a IF-THEN. Měřená hodnota osvětlení bude pouze číslo z A/D převodníku. Uživatel si může nastavit 2 hranice, při kterých se světlo zapne nebo vypne, a tak si nastavit hysterezi obvodu. Ošetřete, aby zadané hranice nebyly mimo rozumný rozsah.
Jako fotorezistor můžete použít typ 5537 nebo podobný.

Příklad řešení:

```
0 REM Nouzove osvetleni
1 PRINT "\f0\n\n"
2 INPUT H
3 IF H<10 THEN GOTO 2
4 IF H>1000 THEN GOTO 2
5 INPUT L
6 IF L>=H THEN GOTO 5
7 AINP A; CURSOR 0; PRINT A,"   "
8 IF A<L THEN DOUT 1,1
9 IF A>H THEN DOUT 0,1
10 DELAY 200; GOTO 7
```

Poznámky k řešení:
Na řádcích 2 až 6 je vyžadováno zadání hranic, při kterých LED zhasne nebo se rozsvítí (i s kontrolou povolených hodnot). Vlastní smyčka je na řádcích 7 až 10. Po změření je hodnota vypsána. Příkaz PRINT je doplněn o vymazání delšího čísla, které by mohlo zůstat na displeji z předchozího měření. Pak je na řádcích 8 a 9 LED rozsvícena či zhasnuta.

Příklad 29:

Zadání:
Vytvořte z TinyBasRV jednoduchý kryptografický přístroj, který bude kódovat a dekódovat znaky abecedy **a** až **z** na jiný znak podle vámi zvolené šifrovací tabulky. Program po stisku příslušného znaku musí ukázat jak zakódovaný, tak odpovídající dekódovaný znak.

Rada:
V poli @ si pomocí příkazu LET vytvořte šifrovací tabulku. Po kontrole, zda zmáčknutá klávesa patří mezi povolené znaky, vyberte znak z tabulky, zobrazte ho a pak tabulku prohledejte pro dekódování. Doporučené příkazy PRINT, INKEY, CURSOR, GOTO a IF-THEN.

Příklad řešení:

```
0 REM Sifrovaci stroj
1 LET @0=6,18,11,5,20,14,22,2,9,15
2 LET @10=17,8,1,10,19,0,13,4,23,24
3 LET @20=25,7,16,12,21,3
4 PRINT "\fZnak   Koder   Dekoder"
5 INKEY K
6 IF K<97 THEN GOTO 5
7 IF K>122 THEN GOTO 5
8 LET Z=K-97; CURSOR 34; PUTCH K
9 CURSOR 41; PUTCH 65+@Z; LET I=0
10 IF @I=Z THEN GOTO 12
11 LET I=I+1; IF I<26 THEN GOTO 10
12 CURSOR 50; PUTCH 65+I; GOTO 5
```

Poznámky k řešení:
Základem programu je kódovací tabulka v poli @. Všech 26 znaků anglické abecedy má svůj náhodně zvolený kód. Znak „a" má kód na pozici 0, „b" na 1, atd. Čísla kódů mohou být od 0 do 25.
Uvažujeme zadávání malých písmen abecedy, jejich hodnoty budou 97 až 122. Na to čekají řádky 5 až 7. Ukazatel do kódovací tabulky v proměnné Z získáme odečtením hodnoty 97 od znaku. Vypíšeme zadaný znak, jeho zakódovanou podobu ve formě velkého písmene, které začínají na hodnotách 65 (řádky 8 a 9). Na řádcích 10 a 11 tabulku prohledáme pro opačnou funkci – dekódování. Řádek 12 dekódovaný znak vypíše.

Příklad 30:

Zadání:
Napište program pro jednoduché stopky. První zmáčknutí libovolné klávesy měřený čas odstartuje, další pak hodnotu zastaví. Zobrazovat se bude na minuty:sekundy,desetiny sekund.

Rada:
Použijte příkazy INKEY, LET, PUTCH, PRINT, GOTO a IF-THEN. Po spuštění programu nejprve počkejte na zmáčknutí klávesy. Uložte si aktuální čas startu. Zobrazujte rozdíl mezi aktuálním časem a startem dělení 3191 (počet tiků za desetinu sekundy). Nejprve hodnotu dělenou 600 – to budou minuty a pak sekundy a desetiny sekund. Nezapomeňte, zobrazit sekundy vždy na 2 desetinná místa. Znak **0** má hodnotu 48.
Další stisk počítání zastaví, aby se dal přečíst dosažený čas. Třetí stisk klávesy pak opět vrátí stopky na nulový čas a připraví ke startu.

Příklad řešení:

```
0 REM Stopky
1 PUTCH 12; PRINT “0:00,0“
2 INKEY K
3 IF K=0 THEN GOTO 2
4 TIME B
5 DELAY 80
6 TIME T; LET T=T-B; LET S=T/3191
7 LET A=S/600; LET S=S-A*600
8 CURSOR 0; PRINT A,“:“
9 LET A=S/10; LET S=S-A*10
10 IF A<10 THEN PUTCH 48
11 PRINT A,“,“,S
12 INKEY K; IF K=0 THEN GOTO 5
13 INKEY K
14 IF K=0 THEN GOTO 13
15 GOTO 1
```

Poznámky k řešení:
Řešení postupuje podle pokynů v odstavci **Rada**. Nejprve se čeká na start stopek (řádky 2 a 3). Do proměnné B se uloží čas startu. Lidské oko není příliš rychlé, tak má smyčka zobrazování zanesené zpoždění na řádku 5. Uplynulý čas v desetinách sekundy se vypočítá do proměnné S na řádku 6. Řádky 7 až 11 čas zobrazí na displej. Řádek 12 kontroluje zastavení stopek. Řádky 13 a 14 čekají na nový restart programu.

Příklad 31:

Zadání:
Napište program představující jednoduchou hru „Dostihy“. Hráč může s počátečním kapitálem vsadit částku na jednoho ze tří koní. Koně budou do cíle dobíhat s náhodným pořadím.

Rada:
Použijte např. příkazy INPUT, PRINT, PUTCH, LET, CURSOR, DELAY, TIME, GOTO a IF-THEN. Pozici, na které se nachází kůň, si uložte do pole @. Bude se pak lépe odkazovat na koně, který se bude náhodně posouvat pomocí příkazu TIME.
Při sázce na vítězného koně připočítejte hráči trojnásobek vsazené částky.

Příklad řešení:

```
0 REM Dostihy
1 LET M=1000
2 PRINT “\fVsadis? 1-“,M,“\n“
3 INPUT B; IF B<1 THEN GOTO 2
4 IF B>M THEN GOTO 2
5 PRINT “\nKun 1-3?“; INPUT K
6 LET @1=1; LET @2=1; LET @3=1
7 PRINT “\fZavod“; PUTCH p49,64
8 PUTCH p50,128; PUTCH p51,192
9 TIME T; LET V=T/3; LET V=T-V*3+1
10 DELAY 100; LET @V=@V+1; PUTCH p143,V*64+@V
11 IF @V<31 THEN GOTO 9
12 CURSOR 395; PRINT “Vyhral “,V; DELAY 1000
13 IF V=K THEN LET M=M+B*3; GOTO 2
14 LET M=M-B; IF M>0 THEN GOTO 2
15 PRINT “\nDosly prachy\n“
```

Poznámky k řešení:
Počáteční kapitál je nastaven na řádku 1. Pak je uživatel vyzván, aby určil hodnotu sázky s kontrolou, zda se nepokouší švindlovat (řádky 2 až 4). Řádek 5 vyzve k sázce na jednoho ze tří koňů. Zde se platnost volby nekontroluje, pokud hráč zadá nesmyslné číslo, peníze vyhodí oknem. Řádek 6 usadí koně na startovní pozici (pozice koní jsou v odpovídajícím prvku pole @). Řádky 7 a 8 připraví grafiku závodu. Na řádku 9 se vygeneruje, který kůň náhodně postoupí o jedno políčko. To je vykresleno na řádku 10 s tím, že pozice koně se uloží do jeho proměnné.
Řádek 11 řeší případné vítězství koně a ukončení závodu. Na řádcích 12 až 15 se vyplácí výhra, odebírají prohrané peníze a případně ukončuje hra při nedostatku peněz.

Příklad 32:

Zadání:
Napište pro TinyBasRV jednoduchou hru „Žabí skokani“. Na začátku budou 4 žáby jednoho druhu vlevo a 4 žáby druhého druhu vpravo. Uprostřed je jedna mezera. Hráč může skočit s libovolnou žábou do vedlejší mezery nebo jednu sousední žábu do mezery přeskočit. Cílem je na co nejmenší počet skoků přemístit žáby zleva na pravou stranu a naopak.

Rada:
Použijte příkazy INPUT, LET, PUTCH, PRINT, CURSOR, GOTO a IF-THEN. Počáteční a cílovou pozici si můžete uložit do pole @. Znaky pro žáby vlevo a vpravo si zvolte sami. Hráč bude muset vždy zadat „odkud“ skáče „kam“. Jednoduše to lze zadávat číslem např. 34 – skok z pozice 3 na 4. Budete muset ošetřit, aby hráč nezadal nesmyslné pozice skoku. Nesmí také skákat přes 2 a více žab naráz.

Příklad řešení:
```
0 REM Zabi skokani
1 LET P=0; LET @1=8,8,8,8,15,1,1,1,1
2 LET @10=1,1,1,1,15,8,8,8,8
3 PUTCH 12; LET I=1
4 CURSOR I*2; PUTCH v80+@I; PRINT I
5 LET I=I+1; IF I<10 THEN GOTO 4
6 PRINT “\n\n“; INPUT M; IF M>99 THEN GOTO 3
7 LET F=M/10; LET T=M-F*10; LET D=F-T
8 IF T<1 THEN GOTO 3
9 IF F<1 THEN GOTO 3
10 IF D<0 THEN LET D=-D
11 IF @T<>15 THEN GOTO 3
12 LET I=1; IF D>2 THEN GOTO 3
13 LET P=P+1; LET @T=@F; LET @F=15
14 LET K=I+9; IF @I<>@K THEN GOTO 3
15 LET I=I+1; IF I<10 THEN GOTO 14
16 PRINT “\nUspech na “,P,“ skoku“
```

Poznámky k řešení:
V proměnné P je uložen počet skoků. V poli jsou nejprve uloženy pozice žab při startu a pak i cílová pozice. Čísla jsou malá aby se snadno v programu zapsala. Pro zobrazení budou znaky zvýšeny o 80, takže levé žáby se objeví jako Q (81), pravé jako X (88) a volná pozice bude podtržítko (95). Na řádcích 3 až 5 je vykresleno hrací pole s žábami i s jejich pozicemi.
Na řádku 6 hráč zadá svůj skok. Ihned se kontroluje, zda číslo není moc velké. Na řádku 7 se rozdělí odkud (F) a kam (T) se skáče a jak dlouhý je to skok (D). Řádky 8 a 9 kontrolují, zda nebyla někde zadána 0 nebo záporné číslo. Řádek 10 opraví délku skoku na kladné číslo.
Řádek 11 kontroluje, jestli se skáče do volné pozice, řádek 12, není-li skok moc dlouhý. Na řádku 13 se provede platný skok. Řádky 14 a 15 zkontrolují případný konec hry při dosažení cíle.