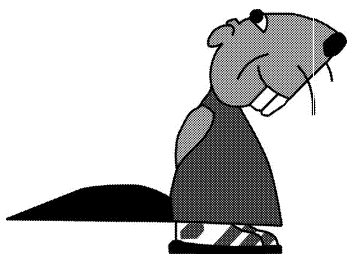


Bobřík informatiky – soutěž žáků a studentů v informatice

JIŘÍ VANÍČEK

Pedagogická fakulta JU, České Budějovice



V listopadu 2008 proběhl první ročník soutěže v informatice pro žáky základních a středních škol s názvem *Bobřík informatiky*. Soutěž si klade za cíl prohloubit a formovat zájem mládeže o informační a komunikační technologie a jejich využívání při vzdělávání. Tato soutěž není zaměřena jednostranně na programování ani na uživatelský přístup a není závislá na nějaké konkrétní platformě.

Článek vysvětluje důvody pro vznik nové soutěže, seznamuje čtenáře s jejím charakterem a uvádí některé typické soutěžní úlohy. Organizátoři věří, že svým zaměřením soutěž přinese (navrátí) do českých škol další podstatný směr, kterým by se měla ubírat výuka mládeže s počítači, a že úlohy osloví tvůrce učebnic, tvůrce vzdělávacích standardů i samotné učitele.

Důvody pro vznik nové soutěže

Informatika jako školní předmět ani informatika jako vědní obor nemá v českých základních a středních školách na různých ustláno. I když by se mohlo zdát, že investice do počítačů, programového vybavení a vzdělávání učitelů byly v minulých letech masivní, že zakotvením vzdělávací oblasti informační a komunikační technologie do rámcových vzdělávacích

programů a ustanovením informatiky jako povinného předmětu (a to i na 1. stupni ZŠ) postavilo tento vyučovací předmět na úroveň předmětů tradičních a že je tudíž důvod ke spokojenosti, pořád přetrvávají potíže, které výuku informačních technologií deformují:

- nízká hodinová dotace (1 hodina pro celý 2. stupeň ZŠ, to je 15krát méně než u matematiky nebo 11krát méně než u součtu dotace dějepisu a výchovy k občanství) [1], nebo srovnáme-li se Slovenskem, dvakrát méně hodin informatiky na 2. stupni a třikrát méně na 1. stupni ZŠ;
- nevyjasněnost cílů (stávající RVP neposkytují učitelé dostatečnou oporu pro to, aby mohl v tomto mladém a nezakotveném předmětu spolehlivě stanovit cíle výuky);
- nízká kvalifikovanost učitelů (málo učitelů s „aprobací“ informatika);
- zřídka využití počítačů v ostatních předmětech (používání textových editorů v mateřském jazyce, tabulkových procesorů a interaktivní geometrie v matematice, počítačových simulací v přírodovědných předmětech, grafických editorů ve výtvarné výchově a ve společenských vědách např. vyhledávání informací) v důsledku snižuje schopnosti žáků používat informační a komunikační technologie jako pracovní nástroj.

Jistá zanedbanost této vzdělávací oblasti s sebou přináší negativní důsledky. Nízké a nesoustavné využití počítačů v běžných školních předmětech, ať již způsobené důvody organizačními, finančními nebo nezájmem, koncentruje zátěž z výuky informační gramotnosti pouze do jednoho předmětu (zvaného informatika, informační technologie apod.). V tomto předmětu je pak patrná rezignace na výukové cíle z oblasti základů informatiky a jejich nahrazení výukou uživatelského přístupu k počítači (z našich zkušeností víme, že někteří učitelé si myslí, že obsahem informatiky je především používání textového editoru a Internetu, příp. tabulkového procesoru [2]). Jinými slovy, informatika jako předmět je izolována od běžného vzdělávání a informatika jako obor je vytlačována ze školního kurikula ve prospěch uživatelského přístupu, což je chyba.

Škola pak má v důsledku menší vliv na formování budoucích zájemců o IT profese, než by bylo vzhledem k jejich potřebnosti ve společnosti žádoucí. Mají-li se z některých nadaných žáků stát budoucí IT odborníci, na rozdíl např. od předmětů s přírodovědným základem (matematika, fyzika, chemie, přírodopis), tito žáci nedostávají na škole základ pro takovou specializaci; škola je v tomto směru vzdělává velice málo. Talentovaní žáci

svůj zájem o počítače často proměňují v zájem o „společenské dění“ na Internetu, o počítačové hry a o aktivity, v nichž není potřeba přemýšlet, řešit problémy, rozumět programování a algoritmizaci, znát principy práce s informacemi, být tvořivý a vynalézavý.

Bystří žáci, zaměření na počítače, nemají doposud takovou komplexní soutěž, kterou nabízejí jiné obory (předmětové olympiády, Matematický klokan, korespondenční semináře apod.) Stávající celostátně organizované soutěže zaměřené na počítače jsou orientovány buď na programování (Matematická olympiáda kategorie P, Baltík a Creative Baltie, Korespondenční seminář z programování), nebo jsou zaměřeny na určitou specifickou nebo aplikační oblast (úpravy digitální fotografie, tvorba webových stránek, zapojení do internetových projektů, soutěže v CAD tvorbě apod.). Některé z těchto soutěží jsou omezeny svým lokální dosahem, použitou platformou či jednorázovostí.

Nová soutěž má za úkol přitáhnout děti k informatickým problémům a na vybraných úlohách jim ukázat, jak pestrá a zajímavá je problematika práce s počítači. Jejich učitelům pak představuje oblasti, které jsou bohužel v současné výuce na školách upozaděny ve prospěch uživatelského ovládnutí kancelářských aplikací (např. algoritmizace, logické zdůvodňování a argumentace, formální popis řešení, reprezentace a struktury údajů, kódování apod.). Učitel si tak může udělat představu, jaké typy úloh jsou chápány jako úlohy z informatiky, které dovednosti a znalosti jsou s daným oborem spojeny. S tímto vědomím pak může modifikovat svoji výuku a upravit svůj vzdělávací program tak, aby jej vyvážal z jednostranného zaměření na uživatelský přístup. Učitel může soutěž využít také k podnícení zájmu žáků o svůj předmět a k jeho propagaci na škole.

Charakter soutěže

K názvu soutěže *Bobřík informatiky* vedly organizátory dva důvody. Za prvé, pod názvem Informatický bobr (nebo podobnými) se tato soutěž pořádá po několik let v řadě zemí, mj. u všech našich sousedů. Soutěž vznikla v Litvě v roce 2004 a nyní existuje mezinárodní spolupráce tvůrců národních soutěží zejména k tomu, aby se využily zkušenosti z jiných zemí a aby národní soutěže měly stejné zaměření. Česká soutěž se tak vlastně připojila k rodině podobných soutěží se stejným cílem a směřováním [3].

Za druhé, kdo četl knihu Jaroslava Foglara *Hoši od Bobří řeky*, ví, že bobřík představuje určitou zkoušku, prokázání dovednosti, umu a kvality. Kdo by neznal bobříka hladu, bobříka mlčení, bobříka odvahy! Ulovení

bobříka znamená překonání sebe sama, splnění nelehké zkoušky. V této soutěži, která takovou zkoušku představuje, lze ulovit bobříka dosažením určitého počtu bodů při testu na počítači. Nejde sice o prokázání mrštnosti nebo odvahy, ale jistě o prokázání důvtipu a schopnosti přemýšlet a rozumět světu počítačů a informací.

Kdo zná soutěž Matematický klokan, má přibližnou představu o tom, jakou soutěží je Bobřík informatiky. Soutěží mezi sebou jednotlivci, soutěž probíhá na školách u počítačů. Každý soutěžící v elektronickém online testu odpovídá na 15 otázek nejčastěji zaškrtnutím správných odpovědí z několika možností. Soutěž probíhá pro danou kategorii v jeden den. Online test umožňuje okamžitou opravu (soutěžící se ihned po skončení svého testu dozví, kolik bodů získal a zda se stal úspěšným řešitelem), rychlé zpracování výsledků a levný provoz (odpadá rozšiřování zadání, zasílání výsledků ze školy do centra, zátěž učitele z opravování apod.)

Soutěž je určena žákům 5. – 9. tříd základních škol a studentům středních škol. Soutěžící jsou rozděleni do tří kategorií podle věku:

Benjamin (5. – 8. r. ZŠ, prima – tercie osmiletého gymnázia)

Junior (9. r. ZŠ + 1. – 2. r. SŠ, kvarta – sexta osmiletého gymnázia)

Senior (3. – 4. r. SŠ, septima – oktáva osmiletého gymnázia)

Testové úlohy

Úlohy se liší svojí náročností, tematickým zaměřením a způsobem řešení [4]. Ve všech zúčastněných státech se používají stejná témata pro výběr testových úloh. Tematicky je možné rozdělit úlohy do sedmi kategorií [5]:

- Digitální gramotnost (základní znalosti o počítačích a informatice; počítačová gramotnost, práce se základními typy softwarových aplikací; etické a právní aspekty, bezpečnost)
- Programování (formální popis řešení, postupu, chování; pochopení, analýza, interpretace a vytváření takových popisů; algoritmy, algoritmické myšlení)
- Řešení problémů (logické zdůvodňování, argumentace; hlavolamy, hádanky, problémy; strategie řešení problémů)
- Zpracování údajů (reprezentace údajů, kódování, modely; matematické základy informatiky, kombinatorika; údaje a struktury údajů, informace a zpracování údajů)

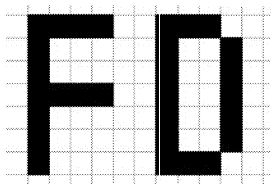
Tyto tematické okruhy nevypovídají nic o obtížnosti úloh. Úlohy jsou cíleny na běžného bystrého žáka, nikoliv na elitní studenty ve speciálních

studijních programech nebo s nadstandardní péčí. Na webu soutěže je ze soutěžních úloh připraven demo test, který si lze anonymně vyzkoušet.

Představíme několik úloh, použitých v ověřovacím Nultém kole, které proběhlo v říjnu 2008.

1. *Kódování písmen* (obr. 1)

Marek si vymyslel své vlastní kódování písmen. Např. písmeno F zakódoval kódem 4114111, písmeno D zakódoval 3222223. Které z následujících písmen je zakódováno kódem 1111114?



Obr. 1

Odpovědi:

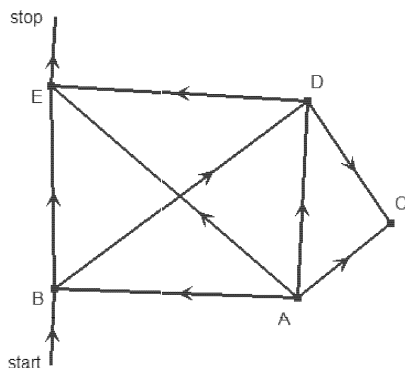
A. L

B. K

C. T

D. J

2. *Mapa skiareálu* (obr. 2)



Obr. 2

Na mapě sjezdovek skiareálu na Bobří hoře se vždy jezdí od bodu *Start* k bodu *Stop*. (*Stop* je nástupní místo sedačkové lanovky, *Start* výstup z této lanovky). Ve skiareálu lze projíždět pouze ve směru šipek (na sjezdovkách je zakázáno jezdit do kopce). Každá čára představuje jednu sjezdovku, každý z bodů A B C D E představuje jednu horskou restauraci. Sjezdovky z B do D a z A do E se nekřížují, je zde podjezd.

Je možné, že některými sjezdovkami není možné projet, tak aby bylo možno dojet opět k lanovce. Takové čáry můžeme z mapy odstranit a na fungování skiareálu se vlastně nic nezmění. Kolik čar lze z mapy odstranit?

Odpovědi:

A. 5

B. 4

C. 3

D. 6

3. Odsazení 1. řádku odstavce (obr. 3)

Ve kterém z orámovaných odstavců na obrázku je správně odsazen 1. řádek odstavce?

Vůbec· neměla· ponětí,· co· řekne,· až vstoupí· do· pokoje.· Najednou· stála· u· dveří· a držela· za· kliku.· Zmáčkla· ji· a· pomalu· vešla.¶
- Vůbec· neměla· ponětí,· co· řekne,· až vstoupí· do· pokoje.· Najednou· stála· u· dveří· a držela· za· kliku.· Zmáčkla· ji· a· pomalu· vešla.¶
.....Vůbec· neměla· ponětí,· co· řekne,· až vstoupí· do· pokoje.· Najednou· stála· u· dveří· a držela· za· kliku.· Zmáčkla· ji· a· pomalu· vešla.¶
- - Vůbec· neměla· ponětí,· co· řekne,· až vstoupí· do· pokoje.· Najednou· stála· u· dveří· a držela· za· kliku.· Zmáčkla· ji· a· pomalu· vešla.¶

Obr. 3

Odpovědi:

A. v 1. odstavci

B. ve 2. odstavci

C. ve 3. odstavci

D. ve 4. odstavci

4. Zašifrovaný směr

Bobří používají k orientaci podle světových stran zvláštní kompasu se zašifrovanými směry. Tak například jih představuje číslo 6, severovýchod

číslo 1:30, sever číslo 12. Jak má bobří kompas zašifrován směr na severozápad?

Odpovědi:

A. 10:30

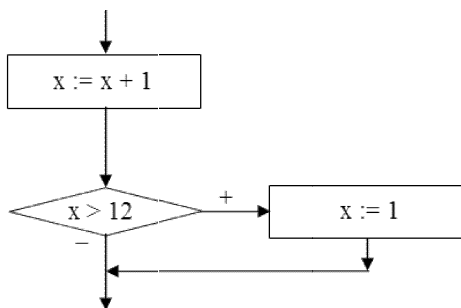
B. 9

C. 11

D. 3:50

5. Část algoritmu (obr. 4)

Na diagramu vidíte část algoritmu, který používá některé z elektronických zařízení. O které zařízení se jedná?



Obr. 4

Vysvětlivky: značka $:=$ znamená „přiřad, dosad“. Např. $A := 3$ znamená „proměnné A přiřad číslo 3“ neboli „za A dosad 3“.

Odpovědi:

A. hodiny

B. elektronický teploměr

C. kalkulačka, když počítá kořen kvadratické rovnice

D. automatická váha

6. Demoverze

Leoš si na svůj počítač nainstaloval demoverzi hry, kterou stáhl z Internetu. Neudělal přitom nějaký prohřešek proti zákonu o ochraně duchovního vlastnictví?

Odpovědi:

A. Neudělal, nainstalováním demoverze se neporuší žádný zákon.

B. Udělal, každý software je potřeba zakoupit.

- C. Udělal, demoverzi lze instalovat pouze se souhlasem autora software.
D. Udělal, software stažený z Internetu není povoleno zdarma nainstalovat.

7. *Barevná hloubka*

Nekomprimovaný bitmapový obrázek má rozměry 40×50 pixelů a zabírá v paměti přesně 1000 bajtů. Kolik nejvíce různých barev může používat?

Odpovědi:

- A. 16 B. 128 C. 256 D. více než 16 000

8. *Kolik bylo bobrů?*

Bobři vždy mluví pravdu, ondatri vždycky lžou. V tůni žije dohromady deset zvířat, samí bobři nebo ondatri. Slepý krtek jde kolem a chce vědět, kolik bobrů a kolik ondater zde žije. Proto se zeptá každého ze zvířat: „Kolik bobrů tu žije?“ Deset odpovědí znělo: 4, 4, 1, 5, 1, 1, 2, 4, 3, 2

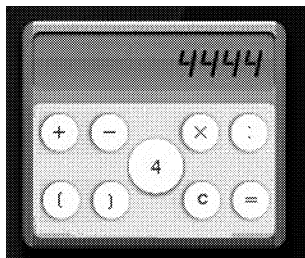
Nyní již byl slepý krtek spokojen. Ty také? Otázka: Kolik bobrů žije v tůni?

Odpovědi:

- A. 2 B. 1 C. 4 D. 3

9. *Podivná kalkulačka (obr. 5)*

Na kalkulačce, kterou mají bobři ve svém doupěti, se kromě tlačítek pro základní početní operace $+$ $-$ $\cdot$ $:$ a tlačítek $[]$ $[C]$ $[=]$ nachází již jen tlačítko pro číslici $[4]$. Tlačítko $[4]$ však smí bobr mezi dvěma zmáčknutími tlačítka $[C]$ stisknout nanejvýš čtyřikrát (každou svojí nohou pouze jedenkrát). Při pátém stisku $[4]$ se celá kalkulačka zablokuje.



Obr. 5

I s takovou podivnou kalkulačkou se dá počítat. Např. $44 : 4 + 4 =$ zobrazí na displeji číslo 15, nebo $4 \cdot (4 + 4) + 4 = 36$. Které z následujících čísel nelze na displeji této kalkulačky zobrazit?

Odpovědi:

A. 21

B. 2

C. 111

D. 256

Výsledky

Národní kolo 1. ročníku soutěže proběhlo v listopadu 2008, organizátorem byla katedra informatiky Pedagogické fakulty Jihočeské univerzity ve spolupráci s Jednotou školských informatiků. Soutěže se zúčastnilo 4069 žáků a studentů ze 79 škol ze všech krajů České republiky. 35 % ze zúčastněných se stalo úspěšnými řešiteli, celkem 38 soutěžících zodpovědělo správně všechny otázky a dosáhlo plného počtu bodů.

U všech úloh představených v tomto článku jsou správnými odpověďmi A. V následujícím textu přikládáme zdůvodnění správného řešení.

1. Kódování písmen

Marek kodoval svá písmena tak, že zapsal počet černých políček v jednotlivých řádcích shora dolů. Zkontrolujeme na obrázku u zadání úlohy: písmeno F (4114111) má v horním řádku čtyři černá políčka, ve druhém shora jedno černé políčko. Podobně písmeno D (3222223) má v prvním a posledním řádku po třech černých políčkách, v ostatních řádcích po dvou černých políčkách.

Kód 11111114 ukazuje, že ve spodním řádku písmena jsou čtyři černá políčka, v ostatních pouze jedno. Takovému popisu odpovídá pouze písmeno L. Písmeno J je podobné, ovšem jeho dolní oblouk by měl ve spodním řádku mít méně než čtyři černá políčka (podobně jako je tomu u písmene D).

2. Mapa skiareálu

Od chaty A vedou pouze šipky pryč. K chatě A se tak není možné dostat, můžeme tedy všechny 4 čáry, vedoucí od chaty A, odstranit. Od chaty C se zase nedá odejet, můžeme tedy odebrat obě čáry směřující k chatě C.

Celkem z mapy odebereme 5 čar (kdybychom pouze sečetli $4 + 2$, čáru AC bychom odebrali dvakrát).

3. Odsazení 1. řádku odstavce

V prvním odstavci je správně použito formátování odstavce (odsazení

1. řádku). V ostatních je jako náhrada formátování použito vkládání znaků do textu (tabulátorů nebo mezer).

4. *Zašifrovaný směr*

Pokud překryjeme směrovou růžici kompasu přes ciferník hodin, bude směr, do kterého míří velká ručička, odpovídat času, který by na hodinách zobrazovala (obr. 6). Severozápadu odpovídá čas 10:30.



Obr. 6

5. *Část algoritmu*

Algoritmus vlastně říká: přičti do proměnné X jedničku, a pokud je číslo v proměnné X větší než dvanáct, nahraď toto číslo jedničkou. Opakuje-li se tento algoritmus pořád dokola, čísla v proměnné X se mění z jedničky na dvojku, na trojku a tak dále až na dvanáctku a pak znovu na jedničku.

Přesně takto se střídají čísla, na která postupně ukazuje hodinová ručička na ciferníku hodin. Jedná-li se o hodiny elektronické s „americkým“ zobrazováním času pomocí zkratk *a.m.* a *p.m.*, čísla spočítaná algoritmem se zobrazují na displeji.

6. *Demoverze*

Demoverze programu je určena právě k instalaci na počítači zdarma. Slouží k tomu, aby se uživatel mohl přesvědčit o kvalitách a výhodách daného software, ale není to plnohodnotná verze. Leoš žádný prohrěšek proti zákonu neudělal.

7. Barevná hloubka

Obrázek má 2000 pixelů a 1000 bajtů, na každý pixel tak připadá půl bajtu. Každý pixel tedy zabírá 4 bity, což umožňuje $2^4 = 16$ barevných kombinací. Obrázek může mít nanejvýš 16 barev.

8. Kolik bylo bobrů?

Bobr mluví pravdu, takže všichni bobři musí na otázku „Kolik bobrů tu žije?“ říci stejné číslo. Toto číslo však nemůže říci žádná ondatra, protože lže. Toto číslo se tudíž vyskytuje mezi odpověďmi tolikrát, kolik bobrů jej vyslovilo (kdyby bylo v tůni pět bobrů, muselo by se číslo 5 vyskytovat pětkrát).

Podívejme se na řadu čísel a hledejme číslo, jehož hodnota se shoduje s četností jeho výskytu v této číselné řadě. Číslo 2 se zde vyskytuje dvakrát; žádné jiné číslo tuto vlastnost nemá. V tůni žili dva bobři.

9. Podivná kalkulačka

Ukážeme, která z nabízených čísel lze na kalkulačce zobrazit:

$$(4 + 4) : 4 = 2$$

$$444 : 4 = 111$$

$$4 \cdot 4 \cdot 4 \cdot 4 = 16 \cdot 16 = 256$$

Číslo 21 pomocí nanejvýš 4 čtyřek zobrazit na displeji nelze:

$$21 - 4 = 17, \text{ které třemi čtyřkami nelze dosáhnout } (4 \cdot 4 = 16, 44 : 4 = 22)$$

$$21 + 4 = 25, \text{ které také nelze dosáhnout třemi čtyřkami } (44 : 4 = 22)$$

$$21 \cdot 4 = 84 - \text{opět nelze } (44 + 4 \text{ ani } 44 \cdot 4 \text{ a taktéž } 4 \cdot 4 \cdot 4 \text{ nevede k úspěchu})$$

$$21 \text{ není dělitelné } 4.$$

Číslo 21 je možné dosáhnout použitím pěti čtyřek (např. $4 \cdot 4 + 4 + 4 : 4$) nebo pomocí dalších tlačítek pro matematické operace (např. $4! - 4 + 4 : 4$), případně použitím tlačítka pro převod z hexadecimální do desítkové soustavy (např. $4 \cdot 4 - 4 : 4 = [HEX \rightarrow DEC]$, neboť $15_{16} = 21_{10}$).

Závěrem

Veškeré informace o soutěži *Bobřík informatiky* lze získat na webu soutěže <http://www.ibobr.cz>. Tam zájemce najde podrobná pravidla, informace pro zorganizování soutěže na škole, test nanečisto, podrobné výsledky letošního ročníku i diskusní skupinu. Učitel, který nechce propásnout příští ročník, zde může školu registrovat, aby dostával zprávy od pořadatelů na svůj e-mail. Věříme, že soutěž zakotví mezi tradičními znalostními soutěžemi mládeže a v důsledku přispěje ke zkvalitnění výuky informatiky na školách. Bobru zdar!

Literatura

- [1] Rámcový vzdělávací program pro základní vzdělávání (se změnami provedenými k 1. 9. 2007). Praha: VÚP, 2007.
- [2] Neumajer, O.: Sedm mýtů o informatice a ICT ve vzdělávání [online]. Metodický portál RVP pro základní vzdělávání, 4. 11. 2008. Dostupné z [www](http://www.rvp.cz/clanek/2747) [http://www.rvp.cz/clanek/2747].
- [3] Informacinių technologijų konkursas „BEBRAS“ [online]. Vilnius: Matematikos ir informatikos insitutas [cit. 20. 11. 2008]. Dostupné z [www](http://www.bebbras.org) [http://www.bebbras.org].
- [4] Hrušecká, A. – Pekárová, A. – Tomcsányi, P. – Tomcsányiová, M.: Informatický bobor – nová súťaž v informačných technológiách pre žiakov základných a stredných škôl [online]. Portál Česká škola, 21. 4. 2008. Dostupné z [www](http://www.ceskaskola.cz/ICTveskole/Ar.asp?ARI=104994&CAI=2129) [http://www.ceskaskola.cz/ICTveskole/Ar.asp?ARI=104994&CAI=2129].
- [5] Tomcsányi, P.: Náročnosť úloh v súťaži Informatický bobor. Sborník konferencie Didinfo 2009, Banská Bystrica.

Lomené čáry jako grafy funkcí s absolutními hodnotami

STANISLAV TRÁVNÍČEK

Přírodovědecká fakulta UP, Olomouc

(Dokončení)

Když se učitel při výběru vhodných příkladů probírá různými sbírkami úloh, setkává se i s příklady na grafy funkcí s absolutními hodnotami, jejichž výsledky nejsou zobrazeny, a v tomto případě mu může pomoci náš druhý program, jehož úkolem je:

Úloha 2. Na základě zadání rovnice funkce s absolutními hodnotami zobrazit její graf.

Sestavíme program GFAH (Graf funkce s absolutními hodnotami), do něhož budeme vkládat rovnice funkcí s absolutními hodnotami zapsané zcela přirozeně, např.

$$y=3/10*x+6-|x+12/5|+2*|x|-5/2*|x-2|.$$

(1)

Program GFAH vytvoříme za těchto omezujících podmínek: „uvnitř“ čísel nebudou zadávány mezery, souřadnice x_i vrcholů lomené čáry budou zadávány v pořadí od nejmenší po největší (v našem příkladu to jsou: 12/5, 0, 2), jejich rozsah bude omezen na interval $-9, 9$ a jejich maximální počet je 9; více napovídá příklad (1). Tento rozsah se pro školní potřeby považuje za dostatečný. Vzhledem k návaznosti problémů, s nimiž jsme se setkali již v předchozím programu *LomCara*, se budeme na části tohoto programu často odvolávat.

Nejprve si uvedme deklarace; po záhlaví

```
program GFAH;
uses Crt, Graph;
```

se opět definuje typ *Zlomek*, a konstanty *Nula*, *Levy*, *Pravy*, J_x , J_y , O_x , O_y a přidá se deklarace

```
const
  Jedna: Zlomek = (Ci: 1; Jm: 1);
```

Převezmeme i deklarace proměnných X , Y , L , Q , M , N , I , J , X_R , Y_R , řádek grafových proměnných a přidáme

```
var
  Ind: Integer;
  St: string;
```

Z programu *LomCara* dále použijeme funkci *NSD* (největší společný dělitel), proceduru *Zkrat* (zkrat zlomek), proceduru *Secti* (sčítání a odčítání zlomků), proceduru *Nasob* (násobení zlomků), proceduru *CtiCislo* (načtení čísla z řetězce znaků) a dále pak proceduru *SestrojOsy* (sestrojení čtvercového jednotkového pole a souřadnicových os) a pak závěrečnou část programu *Vypocty pro graf* a *Vystup grafu*.

Procedura *CtiZlomek1* je mírně upravena na tvar

```
procedure CtiZlomek2(var Z: Zlomek);
var IC, Cis, Sig: Integer;
```

Je do ní vložena procedura *CtiCis* (jak bylo shora uvedeno) a dále je pak

```
begin {CtiZlomek2}
  CtiCis(Z.Ci);
  if St[Ind] = '/' then
    begin
      Inc(Ind);
      CtiCis(Z.Jm)
    end else Z.Jm := 1
end; {CtiZlomek2}
```

Těžištěm programu je část programu (označená jako procedura) *AnalyzujRovnici*, která čte ze zadaného řetězce *St znak po znaku* (do proměnné *Ch*) a nastavuje hodnoty souřadnic x_i vrcholů a hodnoty všech ostatních koeficientů či parametrů. Délka řetězce *St* je v proměnné *Delka* a znaky se testují od *Ind* = 1 do *Ind* = *Delka* (*Ind* = index: poloha znaku v řetězci *St*).

```
procedure AnalyzujRovnici;
var
  Ch: Char;
  Prvni, Err: Boolean; {První = true: ceká se 1. z dvojice
                        znaku absolutni hodnoty}
  Co, Delka: Integer;
  {kod pripustnosti znaku 1 .. byl zlomek
                        2 .. bylo znamenko +,-
                        3 .. bylo znamenko *
                        4 .. byla pocatecni |
                        5 .. byla koncova |
                        6 .. bylo x
                        9 .. byl zacatek
                        .. mezery se toleruji,
                        ale ne v cislech}
```

Základem analýzy je parametr *Co* (*Co* bylo předtím za znak?), který je na počátku nastaven na hodnotu 9; při načtení každého znaku se tak ví, který znak předcházel a je-li načtený znak přípustný, a nastaví se nová

hodnota *Co*.

```
begin
  Err := false; {syntakticka chyba ukoncuje program}
  Ind := 1; N := 1; Co := 9;
  K := Nula; Q := Nula; Z := Jedna; Sig := 1;
  Prvni := true; Delka := Length(St);
```

Cyklus analýzy začíná takto:

```
repeat
  Ch := St[Po];
  case Ch of
```

a v příkazu case se vyšetřují jednotlivé případy (z technických důvodů pokračujeme od začátku řádku)

```
'0'..'9': if Co in [2,9] then
  begin
    CtiZlomek(Z);
    Z.Ci := Z.Ci*Sig;
    Sig := 1; Co := 1
  end else Err := true;
'+', '-': begin
  case Co of
    1: Q := Z;
    5,6,9:
  else Err := true;
  end;
  if Ch = '-' then Sig := -1
  else Sig := 1;
  Inc(Ind); Co := 2
end;
'*'      : if Co <> 1 then Err := true
           else begin
             Inc(Ind); Co := 3
           end;
'|'      : begin
```

```

if Prvni then
begin
  case Co of
    2,9: begin
      L[N] := Jedna;
      L[N].ci := L[N].ci*Sig
    end;
    3 : begin L[N] := Z end;
    else Err := true
  end;
  Inc(Ind); Co := 4
end else begin
  case Co of
    6: Z := Nula;
    1:
  else Err := true
  end;
  X[N] := Z;
  X[N].ci := - X[N].ci;
  Inc(N); Inc(Ind); Co := 5
end;
Prvni := not Prvni
end;
'x','X': if (Co in [2,3,4,9]) then
begin
  if Prvni then
    if Co in [2,9] then
      begin
        K := Jedna;
        K.ci := K.ci*Sig;
      end else K := Z;
      Inc(Ind); Co := 6
    end
  else Err := true;
' ' : begin Inc(Ind) end
  else Err := true
end
until (Ind > Delka) or Err;

```

```

if Co = 1 then Q := Z;
if Err then
begin
  WriteLn('Syntax error ',Co,' in ',Ind);
  ReadLn;
  Halt
end
end; {AnalyzujRovnici}

```

Vlastní program už je pak docela jednoduchý:

```

begin {program}
  ClrScr;
  WriteLn('Graf funkce s absolutními hodnotami');
  WriteLn;
  Write('Zadej rovnici ve tvaru: ');
  WriteLn('y = 3/10*x+6-|x+12/5|+2*|x|-5/2*|x-2|');
  WriteLn('max 9 absolutních hodnot s usporadаныmi vrcholy');
  WriteLn; Write('Vloz rovnici (|=ALT 179): y = ');
  ReadLn(St);
  {for I := 0 to 10 do X[I] := Nula; }
  Q := Nula; N:=1;
  AnalyzujRovnici;

```

Dále následují úseky programu *Vypočty pro graf* a *Vystup grafu* uvedené ve 2. části článku. Programem GFAH je tak Úloha 2 vyřešena.

Literatura je uvedena za 1. částí článku.