Technická univerzita v LIBERCI
FAKULTA PEDAGOGICKÁ

Katedra : matematiky a didaktiky matematiky

Kombinace oborů : matematika - informatika

Soustavy lineárních rovnic na ZŠ

Systems of Linear Equations at Primary Schools

Diplomová práce 97-FP-KMD-001

Autor : Radek Vaňura
Adresa : Štefánikova 1945
544 01 Dvůr Králové nad Labem

Vedoucí práce : RNDr. Daniela Bittnerová, CSc.
Konzultant DP :

Počet	stran	Tabulek	Příloh	Grafů
	38	1	27	1

V Liberci dne : 20. 3 ..... 19 99

Technická univerzita v Liberci
FAKULTA PEDAGOGICKÁ
461 17 LIBEREC 1, Hálkova 6 Tel./Fax: 420.48.5227 332

Katedra: matematiky a didaktiky matematiky

ZADÁNÍ DIPLOMOVÉ PRÁCE

(závěrečného projektu)

diplomant:	Radek VAŇURA
adresa:	Štefánikova 1945, 544 01 DVŮR KRÁLOVÉ n.L.
obor:	matematika - informatika pro 2. stupeň škol
Název DP:	**SOUSTAVY LINEÁRNÍCH ROVNIC NA ZŠ**
Vedoucí práce:	RNDr. Daniela Bittnerová, CSc.
Termín odevzdání:	květen 1998

Pozn.: Podmínky pro zadání práce jsou k nahlédnutí na katedrách. Katedry rovněž specifikují zadání: východiska, cíle, předpoklady, metody zpracování, základní literaturu (zpravidla na rub tohoto formuláře). Zásady pro zpracování DP lze zakoupit v Edičním středisku TU a jsou též k dispozici v UK TUL, na katedrách a na Děkanátě Fakulty pedagogické.

V Liberci dne 29. 10. 1997

J. Wild
.....
vedoucí katedry

V. Kárda
.....
proděkan

Převzal (diplomant):

Datum:

Podpis: Radek Vaňura

KMD/M
38 s. + 27 s. příl.
1 ml. příl. + DISKETA

TECHNICKÁ UNIVERZITA V LIBERCI
Univerzitní knihovna
Voronežská 1329, Liberec 1
V 217/99 P+DISKETA

SOUSTAVY LINEÁRNÍCH ROVNIC NA ZŠ

Úvod:
Již na základních školách se žáci setkávají s pojmem soustava lineárních rovnic, i když ve velmi jednoduché podobě.

Cíl:
Popis a rozbor současného stavu metodiky zavedení tohoto problému. Navržení vhodného učebního textu nebo doplňující pomůcky pro ZŠ týkajícího se dané partie. Použití textových struktur. Ověření navrženého textu ve výuce.

Požadavky:
Schopnost studovat učebnici psanou anglicky, ovládat matematický textový editor.

Literatura:

Učebnice matematiky pro ZŠ.
Sbírky úloh pro ZŠ.

Zadavatel: RNDr. Daniela Bittnerová, CSc.

Prohlášení o původnosti práce:

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a že jsem uvedl veškerou použitou literaturu.

Liberec 20.3.1999

Radek Vaňura

Poděkování

Děkuji RNDr. D. Bittnerové za odborné vedení diplomové práce a za podnětné rady. Základním školám Strž ve Dvoře Králové nad Labem a Vesec v Liberci děkuji za dovolení vyzkoušet výukový program v jejich počítačových učebnách a žákům osmých a devátých tříd děkuji za svědomitou spolupráci.

Prohlášení k využívání výsledků DP:

Jsem si vědom těchto skutečností:

- diplomová práce je majetkem Fakulty pedagogické TU v Liberci,
- s diplomovou prací nelze bez svolení školy disponovat,
- diplomová práce může být zapůjčena či objednána (kopie) za účelem využití jejího obsahu.

Beru na vědomí, že po pěti letech si mohu diplomovou práci vyžádat v Univerzitní knihovně TU v Liberci, kde bude uložena.

Radek Vaňura

Štefánikova 1945

54401 Dvůr Králové nad Labem

Anotace

Tato práce se zabývá rozborem současného stavu výuky soustav lineárních rovnic na základních školách. Jako doplňující prvek do výuky této látky je zde navržen počítačový program, který umožňuje procvičit řešení libovolného počtu příkladů vybranou metodou v určitém čase. Praktickou části je jeho vyzkoušení na dvou základních školách a posouzení výhod či nevýhod této alternativní výuky. Je vypracován v grafickém prostředí programovacího jazyka Turbo Pascal, verze 7.0.

Summary

This work concentrates on analysing the current situation in the teaching of linear equations system in primary schools. As a complementary item to be incorporated into the teaching of this matter, the work puts forward a computer programme enabling to practise solutions-finding for any number of examples by a selected method within a pre-set time. The practical part is formed by its testing at two primary schools and by appreciating this alternative teaching's advantages or disadvantages. It has been worked out in the graphic environment of the 7.0 version of the Turbo Pascal programming language.

Annotation

Diese Arbeit befaßt sich mit der Analyse des gegenwärtigen Standes bei dem Grundschulunterricht im System der Linearen Gleichungen. Als Ergänzung des Unterrichts wird hier ein Rechnerprogramm vorgeschlagen, das ermöglicht, die Lösung einer beliebigen Anzahl von Beispielen durch die gewählte Methode binnen der gegebenen Zeitspanne durchzuüben. Der praktische Teil besteht in seiner Erprobung in zwei Grundschulen und in der Beurteilung der Vor- und Nachteile dieser alternativen Unterrichtsmethode. Das Programm wurde im graphischen Milieu der Programmierungssprache Turbo Pascal, Version 7.0 ausarbeitet.

Obsah

1. ÚVOD **1**

2. MATEMATICI HISTORIE **3**

2.1 DIOFANTOS Z ALEXANDRIE 3
2.2 FRANÇOIS VIÈTE (1540-1603) 4
2.3 RENÉ DESCARTES (1596-1650) 5

3. DEFINICE A POJMY **7**

3.1 ROZSAH UČIVA STŘEDNÍ ŠKOLY 7
3.1.1 Rovnice a jejich početní řešení *7*
3.1.2 Rovnice a jejich grafické řešení *10*
3.1.3 Soustavy rovnic s jednou neznámou *11*
3.1.4 Lineární rovnice s jednou neznámou *11*
3.1.5 Lineární rovnice o dvou neznámých *11*
3.1.6 Soustavy rovnic o více neznámých *12*

4. SOUČASNÝ ZPŮSOB VÝUKY NA ZŠ **14**

4.1 UČEBNICE 7. ROČNÍKU ZŠ 14
4.1.1 Rovnost *14*
4.1.2 Ekvivalentní úpravy rovnic *15*
4.1.3 Lineární rovnice *16*
4.1.4 Slovní úlohy *17*
4.2 UČEBNICE OSMÉHO ROČNÍKU 17
4.2.1 Řešení lineárních rovnic s jednou neznámou *17*
4.2.2 Řešení lineární rovnice se dvěma neznámými *18*
4.2.3 Soustava dvou lineárních rovnic se dvěma neznámými *19*
4.3 UČEBNICE DEVÁTÉHO ROČNÍKU 22
4.3.1 Řešení soustavy rovnic s dvěma neznámými *22*

5. VÝUKOVÉ PROGRAMY **24**

5.1 VÝUKOVÝ PROGRAM „MATIK“ 25
5.2 VÝUKOVÝ PROGRAM „SOUSTAVY LINEÁRNÍCH ROVNIC“ 26
5.3 ZKOUŠENÍ VÝUKOVÉHO PROGRAMU NA ZŠ 28
5.3.1 ZŠ Strž - Dvůr Králové nad Labem *28*
5.3.1.1 První návštěva 28
5.3.1.2 Druhá návštěva 29
5.3.1.3 Hodnocení 30
5.3.2 ZŠ Vesec - Liberec *30*
5.3.2.1 První návštěva 30
5.3.2.2 Druhá návštěva 32
5.3.2.3 Desetiminutovka 33
5.3.2.4 Hodnocení 34
5.4 CELKOVÉ ZHODNOCENÍ 34

6. ZÁVĚR **37**

7. SEZNAM PRAMENŮ **38**

1. Úvod

Na základní škole se žáci poprvé setkávají s termínem soustava lineárních rovnic. S touto látkou jsou seznamováni postupně s využitím základů, které jsou jim předkládány od sedmého ročníku, kde se poprvé seznamují s rovností a s rovnicemi o jedné neznámé, až po ročník osmý, kde se vyučují lineární rovnice o dvou neznámých, a konečně i vlastní soustavy lineárních rovnic, resp. metody, které umožňují jejich vyřešení.

Na všech školách seznamuje žáky s každou novou látkou učitel, který se při svém výkladu snaží co nejvíce používat názorných pomůcek tak, aby byla daná látka více přístupná chápání jednotlivých žáků.

Je samozřejmé, že u každé látky je nutné ji i dostatečně procvičit na příkladech. K tomu slouží na mnoha školách sbírka úloh, která obsahuje velké množství vhodných příkladů.

Naší snahou bylo použít v době počítačů zcela odlišný způsob procvičení příkladů při počítání nebo konstrukci řešení soustavy lineárních rovnic o dvou neznámých. Takovou možnost poskytuje vhodný počítačový program, který sám bez pomoci učitele předkládá příklady k vyřešení každému žáku. Přitom kontroluje čas a správnost odpovědí, s jejichž pomocí stanoví výslednou známku. Největší důraz byl kladen na grafickou metodu, která při zobrazení na monitoru počítače poskytuje velice názornou představu o řešení dané soustavy lineárních rovnic. Grafické řešení na tabuli ve třídě je velice náročné, protože k narýsování jednoho příkladu potřebujeme mnoho cenného času. Vlastní procvičení je pak takřka nemožné právě z důvodu velké časové náročnosti. Tuto situaci je výhodné řešit právě pomocí počítačů.

Samozřejmě nemůžeme počítačem nahradit celou výuku grafické metody na tabuli. Žáky je nutno naučit rýsovat. V případě programu přímky zobrazuje

počítač sám. Proto je tento program vytvořen jako doplňující pomůcka při výuce a procvičování soustav lineárních rovnic.

Cílem této diplomové práce je tedy vyzkoušení programu jako nového didaktického postupu k lepšímu procvičení počítání soustav lineárních rovnic.

K tvorbě programu v grafickém prostředí byl použit programovací jazyk „Turbo Pascal 7.0“, který nám byl zapůjčen ZŠ Lesní LIBEREC.

Vlastní diplomová práce je napsána v textovém editoru Word 7.0.

2. Matematici historie

2.1 Diofantos z Alexandrie

Byl významným řeckým matematikem, který se celý život věnoval řešení rovnic. Při jejich zápisu se ještě nepoužívaly číslice, ale celý zápis byl slovní. Číslice se zapisovaly písmeny, a to pomocí řeckého způsobu psaní, protože tento systém byl v této době nejlepší.

Asi od 5. století před naším letopočtem se k zápisu čísel používala písmena a několik pomocných značek převzatých z abeced jiných jazyků. Číslo jedna se zapsalo jako alfa s jednou čárkou, dvě jako beta s čárkou, tři jako gama s čárkou atd. Deset jako iota s čárkou, dvacet jako kappa s čárkou, třicet jako lambda s čárkou atd. Touto čárkou se v řečtině označoval přízvuk a tím se vlastně odlišovalo příslušné písmeno od číslice, kterou také vyjadřovalo. Vícemístná čísla se zapisovala součtově, přičemž se řadila vedle sebe sestupně odleva doprava. Přízvuk se vynechával a nad číslem se psala vodorovná čárka. Tisíce se zapisovaly stejným písmenem, ale čárka se psala před písmenem, tedy vlevo dole. Tím se vlastně poprvé objevil zárodek psaní různých jednotek, který je již v dnešní době obvyklý. Řekové k tomu ovšem potřebovali celou svoji abecedu a ještě některé pomocné znaky. Zato ovšem nepotřebovali nulu, protože každé číslo bylo jednoznačně určeno příslušným písmenem abecedy a umístěním čárky. Při tomto způsobu však bylo velice těžké provádět jakékoliv výpočty. Pro všechny úlohy se používalo slovní řešení. Neexistoval obecný zápis.

Tím se dostáváme k zápisu rovnic. Rovnice se zpočátku zapisovaly slovně. Na druhém stupni vývoje se často používané znaky či symboly nahrazovaly zkratkami, které se stále odvozovaly ze slov. Diofantos vymyslel nový, zjednodušený způsob zápisu.

U Diofantova objevu nejde jen o zjednodušení zápisu algebraických výrazů. Kromě větší přehlednosti a srozumitelnosti je možné si osvojit i určitý mechanismus, který nesmírně zjednodušuje výpočty. Bylo by těžké si představit vzorce či výpočty bez nějakého systému jejich zápisu, při kterém bychom se museli vyjadřovat pouze slovně. Diofantos tím zahájil éru, která nakonec vedla k našemu nesmírně ekonomickému způsobu zapisování.

Tento způsob zápisu umožňuje vytvoření mnoha algoritmů. Diofantos však tak daleko nedošel. Uvědomil si některé výhody zápisu matematických operací, objevil matematickou notaci, ale na objev algoritmu se muselo čekat ještě více než půl tisíciletí.

2.2 François Viète (1540-1603)

Tento geniální francouzský aritmetik a algebraik působil v 16. století. Měl velké zásluhy i jako geometr. A to přesto, že byl vynikajícím právníkem a advokátem. V době, kdy Francouzi válčili se Španěly, se mu podařilo rozluštit tajný kódovací klíč, který umožnil Francii číst tajné zprávy nepřátel.

Avšak jeho hlavní zásluhou je pozvednutí algebry na třetí, ryze symbolický stupeň. Roku 1591 vydal svůj „Úvod do analytického umění“, v němž jasně vyslovuje svůj princip homogennosti. Tento princip ve skutečnosti znamená, že můžeme navzájem srovnávat jen veličiny stejného druhu, tedy jen úsečky s úsečkami, jablka s jablky. Naproti tomu je nepřípustné dávat do jakéhokoliv vztahu úsečky, plochy a tělesa. Tuto myšlenku Viète rozvedl a začal počítat pomocí velkých písmen. Přitom samohlásky byly symboly pro neznámé a souhlásky pro známé veličiny. Všechny tyto „veličiny“ jsou však u něho stále ještě útvary geometrické.

Viète jako první přísně oddělil aritmetiku, tj. konkrétní počet jakožto počítání s čísly, od počítání pomocí písmen. Tam, kde to šlo, přenesl strojovost a mechaničnost počítání s čísly na počítání s písmennými symboly. Užíval již

znamének plus a minus, ale neužíval znamének pro složitější matematické úkony a operace. Jedinou výjimkou je zlomková čára, kterou používal v dnešním smyslu. Tím tedy Viète jako první převedl „matematiku“ na „vzorce“ a spojil je operačními symboly.

Z jeho dalších zásluh jmenujme alespoň zavedení označení „koeficientů“ tak, jak jsme zvyklí říkat v dnešní době konstantám u jednotlivých mocnin neznámé v rovnici.

François Viète vyrovnal náskok geometrie před aritmetikou a algebrou.

2.3 René Descartes (1596-1650)

Stal se zakladatelem tzv. „moderní matematiky“. Udělal poslední krok k objevení dnes běžně používaného souřadnicového systému. U Descarta aritmetika a algebra nejen předchází geometrii (což je ostrý protiklad vzhledem k jeho předchůdcům), nýbrž je jí věcně nadřazena tím, že představuje mnohem obecnější nauku o veličinách, nauku, která se dá - mimo jiné - aplikovat i na geometrii. A to je onen epochální krok, který bychom nejlépe vyjádřili slovy, že na místo matematiky geometrizované nastupuje od dob Descartových matematika algebraizovaná.

Descartes objevil to, čemu dnes říkáme „kartézské souřadnice“, což je odvozeno od latinské podoby Descartova jména „Cartesius“. Tyto souřadnice představují systém dvou navzájem kolmých os; je to osa x a osa y. Každý bod v rovině je pak jednoznačně určen dvěma čísly, udávajícími jeho vzdálenost od obou těchto os. Oněm číslům se pak říká kartézské souřadnice daného bodu. Tím jsme bodu (útvaru ryze geometrickému) přiřadili dvojici čísel (tedy útvar ryze algebraický). Přímka je potom množina bodů, které splňují určitou podmínku. Dvojice libovolných bodů přímky splňují rovnici o dvou neznámých (lépe řečeno, splňují rovnici o dvou proměnných x a y). Tato rovnice je lineární, to znamená, že se obě proměnné vyskytují v rovnici nejvýše v první mocnině.

Právě tady vznikl vlastní název „lineární rovnice“, protože tato rovnice je obrazem geometrické linie, přímky.

Od této doby se používá označení lineárních rovnic ve tvaru $ax + b = 0;\ a \neq 0$.

3. Definice a pojmy

3.1 Rozsah učiva střední školy

Před vlastním rozborem soustav lineárních rovnic uvedeme postupně několik definic a pojmů, které musejí nutně předcházet.

3.1.1 Rovnice a jejich početní řešení

Při řešení většiny fyzikálních, technických a jiných úloh potřebujeme matematicky formulovat zadání problému. Určujeme všechna čísla z daného číselného oboru **M**, pro něž máme definované funkce *f*, *g* proměnné *x*, které nabývají týchž funkčních hodnot. Za množinu **M** se zpravidla volí podmnožina množiny **R** nebo **C.** Funkční hodnoty jsou také z těchto množin. Jedná se tedy o reálné funkce reálné proměnné nebo obecněji komplexní funkce komplexní proměnné.

Zapisujeme :

$$f(x) = g(x),\ kde\ x \in \mathrm{M}.$$

Tento výraz nazýváme ***rovnicí s neznámou*** *x;* $f(x)$ se nazývá ***levá strana rovnice***, $g(x)$ ***pravá strana rovnice***. Je-li $g(x) = 0$, říkáme, že rovnice je v ***anulovaném tvaru***.

Oborem řešení rovnice je číselný obor **M**. Čísla $x \in \mathbf{M}$, splňující podmínky úlohy, kterou rovnice vyjadřuje, nazýváme ***kořeny*** nebo ***řešení rovnice***. Nutné je uvědomit si rozdíl mezi pojmy *rovnost* a *rovnice*. (Rovnost $r = s$ je zápis vyjadřující, že dvě čísla r, s jsou si rovna, a naopak rovnice $f(x) = g(x)$ představuje úlohu, v které je třeba zjistit, kdy uvedená rovnost nastane.)

Podle typu funkcí f, g v rovnicích rozdělujeme na střední škole rovnice do několika typů :

- algebraické (lineární, kvadratické atd.);
- iracionální;
- exponenciální a logaritmické;
- goniometrické.

K určení rovnice je třeba :

- obor řešení **M**;
- označení proměnné (zpravidla pomocí písmen konce abecedy $r, s, t, u, v, x, y, \ldots$);
- funkce f, g, jejichž proměnnou je neznámá a jejichž funkční hodnoty si mají být rovny.

Rozdělení postupu při řešení rovnice

1. část - ***Rozbor*** (analýza zadané úlohy).

Předpokládáme, že daná rovnice má alespoň jeden kořen, který vypočteme pomocí ***úprav*** původní rovnice. Vznikne rovnice ve tvaru, z něhož již umíme všechna řešení určit. Úpravy musíme volit pouze takové, aby se žádný kořen „***neztratil***". Takovým úpravám říkáme ***důsledkové*** a rovnicím, které získáme tímto postupem, říkáme ***důsledkové rovnice.*** Jestliže ***každý*** kořen důsledkové rovnice je kořenem i rovnice původní, nazýváme příslušnou důsledkovou úpravu ***ekvivalentní úpravou*** a obě rovnice ***ekvivalentními rovnicemi***;

Výměna obou stran rovnice je ekvivalentní úprava.	$f(x)=g(x) \Leftrightarrow g(x)=f(x)$	
Přičtení téhož čísla nebo téže funkce, která má smysl pro celý obor řešení rovnice, k oběma stranám rovnice. Tato úprava je ekvivalentní.	$f(x)=g(x) \Leftrightarrow f(x)+c=g(x)+c$ $f(x)=g(x) \Leftrightarrow f(x)+h(x)=g(x)+h(x)$	-kde c je libovolné -pro $h(x)$ definované pro každé $x \in \mathbf{M}$
Násobení obou stran rovnice týmž číslem nebo touž funkcí, která má smysl pro celý obor řešení rovnice. Tato úprava je ekvivalentní, pokud číslo, kterým se rovnice násobí, je různé od nuly, resp. funkce, kterou se násobí, nenabývá v oboru řešení funkce nulové hodnoty.	$f(x)=g(x) \Rightarrow f(x)\cdot c=g(x)\cdot c$ $f(x)=g(x) \Rightarrow f(x)\cdot h(x)=g(x)\cdot h(x)$ $f(x)=g(x) \Leftrightarrow f(x)\cdot c=g(x)\cdot c$ $f(x)=g(x) \Leftrightarrow f(x)\cdot h(x)=g(x)\cdot h(x)$	-kde c je libovolné -pro $h(x)$ definované pro každé $x \in \mathbf{M}$ -kde $c \neq 0$ -pro $h(x)$ definované pro každé $x \in \mathbf{M}$ a $h(x) \neq 0$
Umocnění obou stran rovnice týmž přirozeným mocnitelem. Tato úprava není obecně ekvivalentní.	$f(x)=g(x) \Rightarrow [f(x)]^n=[g(x)]^n$	-kde $n \in \mathbf{N}$
Logaritmování obou stran rovnice. Můžeme je provést jen za předpokladu, že obě strany rovnice jsou kladné. Tato úprava je ekvivalentní.	$f(x)=g(x) \Leftrightarrow \log_a f(x)=\log_a g(x)$	-kde $f(x)>0,\ g(x)>0,\ a>0,\ a\neq 1.$

Tabulka 1 - Přehled důsledkových úprav rovnice $f(x)=g(x), x \in \mathbf{M}$ v [POL-83:159 / Tab. 10.1.]

2. část - *Závěr rozboru*

Určíme množinu $\mathbf{M}' \subset \mathbf{M}$ kořenů rovnice pomocí ekvivalentních úprav. Množina $\mathbf{M}'$ představuje všechna možná řešení dané rovnice;

3. část – ***Zkouška***

Kontroluje, zda všechny naše získané prvky množiny **M'** jsou kořeny naší rovnice. Provádí se dosazením do levé strany rovnice, čímž dostaneme číslo L a do pravé strany rovnice, čímž dostaneme číslo P. Je-li L=P, je dotazované číslo kořenem dané rovnice. Zkouška je nutnou součástí řešení, pokud některé námi použité důsledkové úpravy nebyly ekvivalentní. Zkouška je vhodná i k odhalení numerických chyb během výpočtu, ať už všechny naše úpravy byly ekvivalentní či nikoliv. Výsledkem zkoušky je tedy ověření správnosti výpočtu a stanovení množiny všech kořenů, kterou značíme $\mathbf{M}_k$;

4. část - ***Diskuse řešení***

Provádí se pouze v případě, že rovnice obsahuje nějaké parametry (proměnné koeficienty). Stanoví se, pro které hodnoty těchto parametrů má rovnice řešení a jaký je jejich počet. A samozřejmě, pro které hodnoty řešení nemá.

3.1.2 Rovnice a jejich grafické řešení

Naším úkolem je graficky vyřešit rovnici $f(x) = g(x)$ v oboru $\mathbf{M} \subset \mathbf{R}$. V pravoúhlé soustavě souřadnic $(0; x, y)$ sestrojíme grafy reálných funkcí f, g reálné proměnné x. Potom x-ové souřadnice společných bodů (průniků) obou grafů představují ***reálné kořeny*** rovnice $f(x) = g(x)$. Této metody lze užít převážně tehdy, pokud nedovedeme rovnici vyřešit početně nebo chceme-li před vlastním početním řešením získat přibližnou informaci o reálných kořenech rovnice.

3.1.3 Soustavy rovnic s jednou neznámou

Představme si množinu rovnic s neznámou x

$$f_1(x) = g_1(x),\ f_2(x) = g_2(x),\ \dots,\ f_n(x) = g_n(x).$$

Jednotlivé rovnice pak spojujeme pomocí znaků „$\vee$" (tj. platí aspoň jedna z nich) nebo „$\wedge$" (tj. platí zároveň). V prvním případě je řešení takové soustavy rovnic sjednocením řešení jednotlivých rovnic a v druhém případě jde o průnik řešení obou množin $\mathbf{M}_k$.

3.1.4 Lineární rovnice s jednou neznámou

Rovnice $f(x) = g(x)$ s neznámou $x \in \mathbf{R}$ se nazývá ***lineární rovnice*** neboli ***rovnice 1. stupně,*** jestliže ji lze vyjádřit (ekvivalentními úpravami) ve tvaru

$$ax = b,\ kde\ a \neq 0,\ a \in \mathbf{R},\ b \in \mathbf{R}.$$

Je zřejmé, že potom má vždy jeden kořen

$$x = \frac{b}{a}.$$

Pokud $a \in \mathbf{R},\ b \in \mathbf{R}\ pak\ i\ x = \frac{b}{a} \in \mathbf{R}$.

3.1.5 Lineární rovnice o dvou neznámých

Nyní se zaměříme na funkce o dvou proměnných. $x,\ y \in \mathrm{M}$:

$$f(x,y) = g(x,y),\ x \in \mathbf{M},\ y \in \mathbf{M}.$$

Například lineární rovnice o dvou neznámých $x,\ y \in \mathbf{M}$:

$a_1x + a_2y + a_3 = b_1x + b_2y + b_3$ s koeficienty $a_i \in \mathbf{M},\ b_i \in \mathbf{M}\ (i \in 1, 2, 3)$, kde $a_1 \neq b_1 \vee a_2 \neq b_2$

Při řešení takové rovnice o dvou neznámých hledáme všechny uspořádané dvojice $[x, y]$ čísel z oboru **M,** pro něž jsou dané funkce f, g proměnných

x, y definovány a nabývají týchž funkčních hodnot. Každá tato uspořádaná dvojice $[x, y]$ se nazývá řešením rovnice o dvou neznámých. Toto řešení samozřejmě závisí na oboru **M**, v němž se rovnice řeší (např. v oboru reálných čísel má rovnice jedno řešení, ale v oboru celých čísel nemá žádné řešení). Rovnicím, kde se za neznámé připouštějí pouze celá čísla, říkáme ***diofantovské*** (někdy též neurčité rovnice).

Zpravidla se neuvažují jednotlivé rovnice o více neznámých, ale více rovnic o více neznámých. Pak se snažíme je vyřešit zároveň. Mluvíme o soustavě rovnic o dvou, resp. více neznámých. Řešením se pak rozumí společné řešení, které vyhovuje každé rovnici (tj. průnik množin řešení jednotlivých rovnic).

3.1.6 Soustavy rovnic o více neznámých

Početní řešení soustavy rovnic se provádí různými metodami, při nichž používáme důsledkové úpravy. Podstatou těchto metod je ***eliminace*** (vyloučení) neznámých kromě jediné. Podle způsobu tohoto vylučování známe dvě základní metody:

- Metoda ***sčítací*** (adiční) :
  Spočívá ve vhodném vynásobení soustavy tak, že po sečtení takto vynásobených rovnic se jedna neznámá eliminuje.
- Metoda ***dosazovací*** (substituční) :
  Spočívá ve vyjádření jedné neznámé z jedné rovnice a její dosazení do ostatních rovnic, čímž se v těchto rovnicích tato neznámá eliminuje.

Soustavu rovnic o dvou neznámých $x, y \in \mathbf{R}$ lze rovněž vyřešit i ***graficky***. Takovou soustavu vyřešíme jako průnik množiny bodů, jejichž souřadnice jsou řešením jednotlivých rovnic soustavy. Pak souřadnice bodů společného průniku představují řešení dané soustavy rovnic.

Řešením každé lineární rovnice je přímka. Řešení určité soustavy lineárních rovnic se pak stanoví podle vzájemné polohy dvou přímek. A to podle následujícího rozdělení :

- **Soustava má pouze jedno řešení.** Obě přímky se protínají právě v jednom bodě, jehož souřadnice jsou řešením dané soustavy lineárních rovnic.
- **Soustava má nekonečně mnoho řešení.** Obě přímky splývají, a proto počet bodů společného průniku je nekonečně.
- **Soustava nemá žádné řešení.** Obě přímky jsou rovnoběžné. Nemají žádný společný bod.

4. Současný způsob výuky na ZŠ

Většina učitelů na základních školách používá při výuce soustav lineárních rovnic učebnice a sbírky příkladů, které mají všichni žáci. V učebnici je uveden rozbor a řešení jednotlivých soustav rovnic a ve sbírkách je velké množství příkladů k procvičení. Pomocí těchto učebnic může každý žák doma samostatně studovat.

4.1 Učebnice 7. ročníku ZŠ

4.1.1 Rovnost

Žáci sedmého ročníku jsou schopni používat vzorce pro výpočty, ale zatím o nich nemluví jako o ***rovnicích*** . Například znají vzorec pro výpočet obsahu trojúhelníka. V učebnici pro sedmý ročník (II.díl) je tento vzorec použit a názorně se na něm ukazuje, co je to rovnost. Obsah je zde vyjádřen pomocí dvou zápisů stejného vzorce:

$$S = \frac{z \cdot v}{2} \quad a \quad S = (z \cdot v) : 2;$$ *kde v je výška a z je velikost základny trojúhelníka.*

Je patrné, že v obou případech je obsah stejný. Potom mohou mluvit o ***rovnosti*** . Výraz na ***levé*** straně rovnosti označují L a výraz na straně ***pravé*** označují P. Pokud se výraz L nerovná výrazu P, dostávají ***nerovnost***.

Žáci byli dosud zvyklí řešit úlohy deduktivně. Například je v učebnici použita následující úloha :

Příklad :

Na školní zahradě je vysázeno 20 keřů šeříku a forzýtií. Forzýtií je 12. Kolik keřů šeříku je vysázeno na školní zahradě?

Řešení:

celkem .....20 keřů

forzýtií.....12 keřů

šeříků..... x keřů

$$x + 12 = 20$$

Žáci snadno dosadí za x číslo 8, aby platila rovnost. Žádné jiné číslo rovnost nesplní.

Zápis rovností dvou výrazů, ve kterém máme určit neznámé číslo tak, aby daná rovnost platila, nazýváme ***rovnice***.

Číslo ***x*** v zápisu rovnice je ***neznámá***. Neznámé v rovnici označujeme zpravidla písmeny z konce abecedy: *x, y, z* apod.

Řešit rovnici znamená najít takové číslo, aby po dosazení tohoto čísla za neznámou se rovnice změnila v rovnost. Nalezené číslo se nazývá ***řešení*** nebo ***kořen*** dané rovnice.

Rovnice, tak jako rovnost, má levou a pravou stranu. Značíme je L a P.

O správnosti řešení se žáci přesvědčují pomocí ***zkoušky***.

4.1.2 Ekvivalentní úpravy rovnic

Ekvivalentní úpravy jsou demonstrovány v učebnici pomocí tzv. laboratorních vah, kde při položení stejného závaží na obě strany vah se rovnováha nezmění.

Řešení rovnice se nezmění, jestliže přičteme (nebo odečteme) k oběma stranám (od obou stran) rovnice tentýž výraz. Této úpravě říkáme ***ekvivalentní úprava*** rovnice.

Tato látka je procvičena na příkladech, kde se žáci snaží použít nějakou ekvivalentní úpravu, která povede k vyřešení rovnice.

Příklad

$$2x + 2 = x - 1$$

Řešení :

$$\begin{aligned} 2x + 2 &= x - 1 \\ 2x + 2 - x &= x - 1 - x \quad \left[-x\ (\textit{od obou stran odečteme } x)\right] \\ x + 2 &= -1 \\ x + 2 - 2 &= -1 - 2 \quad \left[-2\ (\textit{od obou stran odečteme číslo } -2)\right] \\ x &= -3 \end{aligned}$$

Řešením rovnice je tedy číslo -3.

Protože se jednalo o ekvivalentní úpravy, musí být číslo -3 řešením rovnice. Správnost výpočtu se opět ověřuje zkouškou.

V této úloze se jednalo pouze o operace sčítání a odčítání, ale stejné pravidlo platí i pro násobení a dělení.

Řešení rovnice se nezmění, vynásobíme-li (dělíme-li) obě strany rovnice týmž číslem různým od nuly. Této úpravě opět říkáme ***ekvivalentní úprava*** rovnice.

4.1.3 Lineární rovnice

V učebnici pro sedmý ročník se také zavádí termín ***lineární*** rovnice s jednou neznámou. Nazývají se tak všechny rovnice, které lze upravit na tvar $ax - b = 0$, kde a, b jsou reálná čísla, pro která platí, že a je různé od nuly $(a \neq 0)$. Dále se zde říká, že každá ***lineární*** rovnice má vždy jedno řešení $x = \frac{b}{a}$.

4.1.4 Slovní úlohy

V druhém dílu učebnice pro sedmou třídu jsou uvedeny příklady jednoduchých slovních úloh na procvičení a je zde uvedeno několik pravidel pro vyřešení slovní úlohy.

- Pozorně přečteme text slovní úlohy, abychom pochopili, co je dáno (podmínky úlohy) a co máme vypočítat (otázka úlohy). Označíme neznámou.
- Všechny podmínky úlohy vyjádříme pomocí neznámé. Dostaneme výrazy a zapíšeme podmínku pro jejich rovnost. Získáme rovnici.
- Rovnici vyřešíme.
- Správnost výsledku ověříme zkouškou tak, že zjistíme, zda řešení vyhovuje podmínkám slovní úlohy.
- Výsledek vyjádříme pomocí slovní odpovědi.

Slovní úlohy jsou zde procvičeny na spoustě příkladů od jednoduchých až po složitější úlohy o pohybu.

4.2 Učebnice osmého ročníku

4.2.1 Řešení lineárních rovnic s jednou neznámou

Zde se jedná většinou o opakování na příkladech převážně ze sedmého ročníku. Opakují se zde definice rovnic, řešení, ekvivalentní úpravy atd. Uvedené příklady jsou od jednodušších až po rovnice s lomenými výrazy.

U příkladů s lomenými výrazy je zdůrazněno :

Při řešení rovnic s ***neznámou ve jmenovateli*** postupujeme stejně jako při řešení lineárních rovnic. Navíc určujeme podmínky pro neznámou tak, aby rovnice měla smysl.

4.2.2 Řešení lineární rovnice se dvěma neznámými

Žáci umí řešit lineární rovnice o jedné neznámé ze sedmého ročníku. V osmém ročníku se začínají řešit rovnice s dvěma neznámými. Je zde zdůrazněno použití ekvivalentních úprav stejně jako u rovnic s jednou neznámou.

Řešení rovnice se dvěma neznámými se nezmění, jestliže přičteme (odečteme) k oběma stranám (od obou stran) rovnice stejný výraz. (***První ekvivalentní úprava.***)

Řešení rovnice se dvěma neznámými se nezmění, jestliže násobíme (dělíme) obě strany rovnice stejným výrazem různým od nuly. (***Druhá ekvivalentní úprava.***)

Žáci se sami přesvědčují na vzorových příkladech, že jedna rovnice se dvěma neznámými má nekonečně mnoho řešení. Řešení je zde znázorněno s použitím tabulky, kde za jednu neznámou žáci dosazují libovolné číslo a druhou neznámou spočítají. Po abecedním uspořádání neznámých v rovnici získávají ***uspořádané dvojice čísel,*** které jsou řešením příslušné rovnice.

Rovnice, které je možné upravit na tvar $ax + by = c$, kde $a \neq 0$ a současně $b \neq 0$ (a, b, c jsou čísla; x, y jsou neznámé), se nazývají ***lineární rovnice se dvěma neznámými.***

4.2.3 Soustava dvou lineárních rovnic se dvěma neznámými

V této kapitole se žáci poprvé setkávají s vlastním termínem ***soustava dvou lineárních rovnic o dvou neznámých.*** Nejdříve si zvolí dvě rovnice o dvou neznámých. Těmto dvěma rovnicím právě říkáme ***soustava dvou lineárních rovnic a dvou neznámých.*** O každé z nich vědí, že má nekonečně mnoho řešení. K vyřešení této soustavy potřebujeme najít takové uspořádané dvojice, které vyhovují oběma rovnicím ***současně.*** Mluvíme o tzv. ***řešení soustavy dvou lineárních rovnic se dvěma neznámými.***

V učebnici je předveden příklad soustavy :

$$x - 2y = -3$$
$$3x + 2y = 7$$

Řešení :

Je ukázáno pomocí vyjádření jedné neznáme pomocí druhé z jedné libovolné rovnice. V našem případě např. $x = -3 + 2y$. *Po dosazení do druhé rovnice soustavy dostaneme*

$$3(-3 + 2y) + 2y = 7,$$

což je rovnice s jednou neznámou, kterou snadno vyřešíme:

$$-9 + 6y + 2y = 7$$
$$8y = 16$$
$$y = 2$$

Nyní dosadíme do upravené rovnice soustavy vypočítané číslo za neznámou y:

$$x = -3 + 2y$$
$$x = -3 + 2 \cdot 2$$
$$x = 1$$

Správnost výsledku ověříme zkouškou. Levá a pravá strana první rovnice:

$$L_1 = 1 - 2 \cdot 2 = -3$$
$$P_1 = -3$$

$$L_1 = P_1$$

Levá a pravá strana druhé rovnice:

$$L_2 = 3 \cdot 1 + 2 \cdot 2 = 7$$
$$P_2 = 7$$

$$L_2 = P_2$$

Když platí $L_1 = P_1$ *a* $L_2 = P_2$, *je* $x = 1, y = 2$ *řešením naší soustavy. Zapíšeme je jako uspořádanou dvojici čísel, tj.* [1, 2].

Tento postup nazýváme ***dosazovací (substituční) metodou*** řešení soustavy dvou lineárních rovnic se dvěma neznámými a můžeme jím řešit každou soustavu.

Při řešení soustav dvou lineárních rovnic se dvěma neznámými nastane právě jeden ze tří možných případů:

- Soustava má právě jedno řešení. (Řešením je jediná uspořádaná dvojice čísel.)
- Soustava nemá žádné řešení.
- Soustava má nekonečně mnoho řešení.

Kromě ***dosazovací*** metody existují i další způsoby řešení soustav rovnic.

V učebnici je uvedena tzv. ***sčítací metoda*** :

$$2u + v = 4$$
$$\underline{4u + 3v = 6}$$

Při řešení se postupuje např. takto :

- *První rovnici vynásobíme (-2), druhou opíšeme:*

$$-4u - 2v = -8$$
$$\underline{4u + 3v = 6}$$

Rovnice sečteme a dostaneme:

$$\underline{v = -2}$$

- *První rovnici vynásobíme (-3), druhou rovnici opíšeme:*

$$-6u - 3v = -12$$
$$\underline{4u + 3v = 6}$$

Rovnice sečteme a dostaneme :

$$-2u = -6 \quad / : (-2)$$
$$u = 3$$

Zkouška:

$$L_1 = 2 \cdot 3 + (-2) = 4, \quad P_1 = 4, \quad L_1 = P_1$$
$$L_2 = 4 \cdot 3 + 3(-2) = 6, \quad P_2 = 6, \quad L_2 = P_2$$

Řešením soustavy lineárních rovnic je uspořádaná dvojice $[3, -2]$.

4.3 Učebnice devátého ročníku

Látka navazuje na kapitolu o lineárních funkcích a grafech. Opět začíná opakováním všech pojmů a definic jako v učebnici předešlé.

Lineární rovnice s jednou neznámou včetně rovnic s neznámou ve jmenovateli jsme upravovali pomocí ekvivalentních úprav (viz učebnice osmého ročníku).

Lineární rovnice s neznámou x je každá rovnice tvaru $ax - b = 0$, kde a je libovolné reálné číslo různé od nuly a b je libovolné reálné číslo. Každá lineární rovnice s jednou neznámou má jedno řešení $x = \frac{b}{a}$.

První ekvivalentní úprava v sobě zahrnuje i odečtení stejného výrazu od obou stran rovnice (odečtení nahradíme přičtením výrazu k němu opačného).

Druhá ekvivalentní úprava obsahuje i dělení obou stran rovnice číslem nebo výrazem různým od nuly. Dělení číslem (výrazem) nahradíme násobením číslem (výrazem) k němu převráceným.

Pokud se jedná o rovnici s lomeným výrazem, je opět nutné stanovit podmínky řešení.

4.3.1 Řešení soustavy rovnic s dvěma neznámými

Opět je zde zdůrazněn vlastní postup hledání uspořádané dvojice, která je řešením první i druhé rovnice. K tomuto účelu opět slouží ekvivalentní úpravy, které jsme již zopakovali. Soustavu lineárních rovnic řešíme metodou ***substituční (dosazovací)*** nebo ***adiční (sčítací)***. Soustava pak může mít právě jedno řešení (řešením je jedna uspořádaná dvojice čísel) nebo nemá žádné řešení nebo má nekonečně mnoho řešení.

Při grafickém řešení soustavy lineárních rovnic postupujeme tak, že sestrojíme grafy lineárních funkcí příslušných k daným rovnicím a určíme souřadnice jejich společných bodů.

- Soustava má jedno řešení. Grafy odpovídajících lineárních funkcí jsou různoběžné přímky.
- Soustava nemá žádné řešení. Grafy odpovídajících lineárních funkcí jsou různé rovnoběžné přímky.
- Soustava má nekonečně mnoho řešení. Grafy odpovídajících lineárních funkcí jsou splývající přímky.

Žáci jsou upozorněni, že pokud má soustava nekonečně mnoho řešení, neznamená to, že řešením je libovolná uspořádaná dvojice čísel.

5. Výukové programy

Mnoho základních škol vůbec nepoužívá osobní počítače, vybavené vhodným softwarem, při výuce jakýchkoliv jiných předmětů, než je informatika. Důvodů může být hned několik.

- Chybí dostatek finančních prostředků na zařízení počítačové učebny.
- Chybí kvalifikovaní učitelé, kteří by byli schopni již zařízenou učebnu i s vhodným softwarem použít k výuce svého předmětu.
- Chybí prostory. Výuka informatiky probíhá mnohdy nepřetržitě, takže k využití učebny pro jiné předměty třeba vůbec nemůže dojít.

Přesto již existuje několik základních škol, kde s osobními počítači mají již dost zkušeností a nebojí se je používat pouze k výuce informatiky. Většinou se jedná právě o hodiny informatiky, které se využijí pro vyzkoušení nějakého programu z oblastí jiných předmětů, např. z fyziky, z chemie, z dějepisu, z biologie, z matematiky, z českého jazyka apod. Žáci se účastní hodiny informatiky, pracují na počítačích, ale náplň látky se týká zcela jiného předmětu.

Programů, které se zabývají výukou a procvičením nějaké látky ze základní školy, stále přibývá. Díky stále se rozšiřujícím schopnostem počítače dochází k simulování různých dějů, které bychom museli obtížně předvádět v laboratořích, dochází k předvádění různých chemických reakcí, složení lidského těla atd. Těchto možností je nekonečně mnoho. Mnoho softwarových firem již zareagovalo a jejich programátoři začali tvořit mnoho skvěle propracovaných názorných programů. Stále se však jedná o programy, které simulují děje, reakce nebo obsahují hodně informací. Například v matematice pro základní školy jich mnoho není. Jedním z nejlepších je výukový program „MATIK“.

5.1 Výukový program „MATIK“

Jedná se o výukový program pro ZŠ z roku 1993, který obsahuje látku matematiky od třetího do osmého ročníku. Byl vytvořen ještě dříve, než byla zavedena povinná devítiletá školní docházka. Jedná se o jeden z nejlepších výukových programů pro ZŠ. Program je velmi dobře graficky proveden a doplněn obrázky. Ty jsou vhodné zejména při výuce v nižších ročnících.

Matik byl vzorem k vytvoření našeho programu „Soustavy lineárních rovnic“. Zkoušení výpočtu lineárních rovnic je zde zařazeno v sedmém a osmém ročníku. V úvodu zkoušení si mohou žáci zvolit libovolný počet příkladů, které jsou jim pak postupně předkládány. Po této volbě následuje upozornění na způsob ovládání a zadávání jednotlivých čísel a znamének. Přesně se zde požaduje vložení čísla nebo znaménka do předem vysvíceného rámečku. Pak už jen následuje vlastní řešení jednotlivých příkladů. Všechny mají v hlavičce uvedené zadání soustavy s doprovodným textem, tj. název metody použité při řešení, radu k výpočtu apod. Během celého zkoušení je třeba, aby každý žák vkládal do předsvícených rámečků již čísla, která si musel spočítat pomocí tužky a poznámkového bloku. Při první chybné odpovědi dostává žák možnost k opravě a pokud ji nevyužije, je mu předloženo správné vyřešení celého příkladu. V pravém spodním rohu je neustále zobrazen počet již správně vyřešených příkladů, celkový počet a pořadové číslo příkladu. Po spočítání všech příkladů je na základě počtu správných a špatných odpovědí stanoveno výsledné známkové hodnocení. Každý žák je nejdříve sám vyzván k svému vlastnímu ohodnocení. Program buď jeho volbu potvrdí, nebo naopak opraví.

Nevýhodou tohoto jinak určitě výborného programu je ukončení příkladu po první chybě. Pokud při jakékoliv odpovědi nevyužijí žáci dvě možnosti, které jsou jim nabídnuty, program příklad předloží celý dořešen. U grafické metody se při zadání souřadnic bodu, které jsou větší než nějaký vnitřní rozsah

programu, vůbec bod nezobrazí, protože se měřítko grafu nepřepočítává. Dále není možné do programu vložit vlastní příklady nebo nějaké upravit.

5.2 Výukový program „SOUSTAVY LINEÁRNÍCH ROVNIC“

Program byl vytvořen pomocí programového jazyka TURBOPASCAL, ver. 7.0, v grafickém prostředí. Úzce se zaměřuje na procvičování příkladů soustav lineárních rovnic o dvou neznámých početní a grafickou metodou.

Databáze příkladů je uvedena v textovém souboru, který je volně přístupný v libovolném textovém editoru. Je tedy možno příklady měnit, či přidávat další soustavy rovnic při zachování určitých pravidel syntaxe pro zápis každé soustavy do souboru. Počet řešení a vlastní hodnoty si program řeší sám, takže není nutné se řešením mnoho zabývat. Při vlastním zkoušení si program z tohoto souboru náhodně vybírá určitý počet příkladů k vyřešení. Počet příkladů, které budou předloženy žákům k vypracování, se zadá parametricky při volání programu z prostředí MS-DOS nebo WINDOWS. Prvním parametrem je počet příkladů a druhým čas v minutách, který bude k výpočtu vymezen.

Žáci mají možnost výběru mezi početními metodami (sčítací, dosazovací) nebo grafickou metodou, pokud nebude učitelem určena konkrétní metoda k procvičení.

V celém programu se první chyba v zadání nějakého čísla bere jako nepozornost a program nabízí druhou možnost k odpovědi. Pokud i tato odpověď bude špatně, je brána jako chyba a program sám zobrazí správnou odpověď. Oproti programu „Matik“ neskončí procvičení tohoto příkladu po první chybě, ale pokračuje až do konce příkladu. Tento postup má bezesporu výchovnou hodnotu, protože povzbudí a zaujme i méně nadané žáky, a neodradí je, když nejsou úspěšní v řešení.

Do předsvícených rámečků je třeba přesně zadat cifry, případně i znaménko toho konkrétního čísla, které zadáváme. U některých těžších příkladů je možné, že žáci nedokáží vše počítat zpaměti, a je tedy opět nutné používat poznámkový blok k pomocným výpočtům. Po ukončení celého procvičování všech příkladů se známka stanoví na základě počtu správných a špatných odpovědí. Při tvorbě známky zde zatím není vzat ohled na rychlost odpovědí, což program již umožňuje, protože během celého zkoušení žákům ubývá čas v pravé části monitoru. Časový limit stejně jako počet příkladů stanoví učitel. Žáci se díky tomuto času musí věnovat programu nepřetržitě a tím se úspěšně zabraňuje vyrušování v hodině. Program tedy vede žáky k plné aktivitě a soustředěnosti, nedovolí odvádět pozornost jiným směrem. V neposlední řadě se v žácích podporuje zájem o matematiku i vztah k moderní technice.

Hlavní výhodou programu je rychlé a přehledné znázorňování při grafickém řešení příkladů, tj. pomocí grafické metody. Ta není při normálním školním vyučování příliš používána a žáci ji snadno zapomínají. V tomto programu se přehledně zobrazují přímky každé rovnice a výsledné řešení je okamžitě vidět na monitoru. Snadno je patrné, kolik řešení má každá soustava rovnic pomocí vzájemné polohy obou přímek. Řešení každé rovnice se na obrazovce zobrazí po zadání dvou libovolných bodů, které této přímce vyhovují. Je na žácích, zda budou zadávat souřadnice průsečíků s osami nebo jakékoliv jiné dva body. Měřítko celého grafu se mění automaticky - vždy tak, aby v něm byly zobrazeny všechny použité body. Po zobrazení obou přímek na monitoru je každý žák vyzván k odpovědi, kolik má daná soustava řešení. Pak je samozřejmě zobrazena odpověď.

Pokud při počítání příkladů uběhne čas stanovený učitelem k vypracování, mají žáci možnost ještě dokončit poslední započatý příklad. Teprve potom je procvičování ukončeno a je stanovena známka programem.

Po stanovení známky je třeba stisknout kombinaci kláves „**ALT-Q**“. Ta je zde proto, aby známka byla vidět až do příchodu učitele, který by tím měl přehled, kdo jak pracuje, případně známky použil ke své klasifikaci. Není doporučeno kombinaci žákům sdělovat, i když je zřejmé, že ji brzy zjistí.

5.3 Zkoušení výukového programu na ZŠ

5.3.1 ZŠ Strž - Dvůr Králové nad Labem

I tato škola nemá učitele, kteří by chtěli ve svých hodinách používat počítačovou učebnu pro jiné předměty, než je informatika. Učebna je na této škole výborně zařízená. Probíhá zde jen výuka informatiky, převážně v odpoledních hodinách. Třídy jsou většinou půlené na chlapce a dívky, protože v počítačové učebně je pouze čtrnáct pracovišť.

5.3.1.1 První návštěva

Na této základní škole nám bylo umožněno navštívit žáky devátých ročníků v hodinách informatiky a s jejich pomocí vyzkoušet výukový program „Soustavy lineárních rovnic“.

Jednalo se o žáky devátých ročníků, kteří již látku soustav rovnic o dvou neznámých probírali v osmém ročníku. Znali tedy všechny metody sloužící k výpočtu soustav lineárních rovnic. V době, kdy jsme výukový program testovali, ještě látku soustav rovnic v devátém ročníku neopakovali, což se samozřejmě projevilo v úvodu zkoušení tohoto programu. Bylo nutné všechny metody připomenout a zopakovat. Největší problémy nastaly s grafickou metodou. Bylo jasně patrné, že grafická metoda není tak hodně používanou metodou při řešení soustav lineárních rovnic na ZŠ.

Před vlastním cvičením bylo nutné žáky seznámit s ovládáním a prostředím výukového programu. Žáci si mohli sami zkoušet jakékoliv metody při řešení. U několika málo žáků se projevil nedostatek dovedností při ovládání

klávesnice počítače. To samozřejmě vedlo ke ztrátám času při vkládání čísel do programu.

V posledních patnácti minutách vyučovací hodiny byli všichni rozděleni do tří skupin tak, že každá měla za úkol řešit soustavy lineárních rovnic pomocí jiné metody. Na každém počítači se měly spočítat dva příklady přidělenou metodou. V závěru program jednotlivé práce oznámkoval, ale toto hodnocení bylo spíše orientační. Celá tato první návštěva byla brána jako seznámení s výukovým programem.

Po zadání úloh bylo nutné chodit po celé učebně a věnovat se každému žáku individuálně při řešení jeho konkrétního příkladu pomocí jemu přidělené metody.

5.3.1.2 Druhá návštěva

Po týdenní přestávce jsme žákům stejné třídy předložili opět při hodině informatiky program soustavy lineárních rovnic. Program i metody jsme jen připomněli a opět jsme vytvořili tři skupiny, které měly za úkol vypočítat za dvacet minut čtyři příklady námi zvolenou metodou pro každou skupinu. Prvnímu žáku, který správně vypočítá náhodně vybrané čtyři soustavy lineárních rovnic bez vážnějších chyb a který bude programem ohodnocen na ***výbornou***, byl slíben zápis do žákovské knížky od jeho učitele matematiky. Tato skutečnost a ubývající čas v pravé straně plochy monitoru vedla žáky bez jakéhokoliv napomínání rychle kupředu.

Až na pár výjimek se podařilo spočítat náhodně vybrané příklady předepsanou metodou ve stanoveném čase. Po dokončení těchto čtyř příkladů program stanovil známku a žák dostal za úkol počítat další čtyři příklady jinou metodou dle svého výběru.

Během celého zkoušení bylo jasně vidět, že druhá práce na tomto programu je daleko rychlejší než první seznamování, které proběhlo v minulém

týdnu. Bystřejší žáci, kteří příklady vyřešili v rekordním čase, si po stanovení známky zkoušeli jinou metodu - v mnoha případech i grafickou, přestože minulý týden tvrdili, že si ji už nepamatují.

5.3.1.3 Hodnocení

Skupina první, tj. chlapci devátého ročníku, pracovala daleko rychleji a nebylo třeba jim příliš názorně vysvětlovat obsluhu programu. Většina totiž navštěvovala hodiny nepovinné informatiky i v odpoledním vyučování, a proto byla v práci s počítačem rychlejší než skupina dívek, kterým bylo třeba velice důkladně vysvětlit ovládání a způsob vkládání čísel do výukového programu.

Ukázalo se, že dívky na poznámkový papír nepočítaly jen pomocné výpočty, ale některé dokonce celé soustavy, které se pak snažily vkládat do počítače. Těmto žákyním bylo nutné vše ukázat ještě jednou, a to přímo na jejich příkladech.

5.3.2 ZŠ Vesec - Liberec

Hlavním problémem výuky jiných předmětů v počítačové učebně, která je také na této škole výborně zařízena, a dokonce disponuje i přístupem do světové sítě Internet, je zcela zaplněný rozvrh učebny, a to výukou a kroužky informatiky. Několik učitelů, převážně fyziky a matematiky, přesto stále doufá, že se situace časem vyřeší.

Třídy jsou i zde při výuce informatiky půlené, a to podle jazyků, které žáci navštěvují. V počítačové učebně je šestnáct stanic, na kterých se tedy střídají po sobě angličtináři a němčináři.

5.3.2.1 První návštěva

Během první návštěvy nám bylo umožněno navštívit žáky osmých ročníků v hodinách matematiky a s jejich pomocí také vyzkoušet výukový program

„Soustavy lineárních rovnic“ v učebně informatiky, která byla volná zcela výjimečně z důvodu nemoci učitele.

Jednalo se tedy o žáky osmých ročníků, kteří látku soustav rovnic o dvou neznámých právě probírali, takže ji měli v čerstvé paměti. Znali ale pouze dvě početní metody sloužící k výpočtu soustav lineárních rovnic. V době, kdy jsme výukový program testovali, ještě neuměli grafickou metodu, což se samozřejmě projevilo v dělení pouze na dvě skupiny. Protože se jednalo o celou třídu matematiky, seděli u jednoho počítače vždy asi dva žáci. Bylo nutné žáky rozsadit tak, aby vždy alespoň u každého počítače seděl jeden žák, který navštěvuje hodiny informatiky, a tak byl schopen počítač bez vážnějších problémů ovládat. Ne všichni žáci zdejší školy totiž navštěvují hodiny informatiky.

V úvodu jsme početní metody pouze připomněli. Stejně jako na ZŠ ve Dvoře Králové, i tady bylo nutné před vlastním cvičením žáky seznámit s ovládáním a prostředím výukového programu. Žáci si mohli sami vyzkoušet jakoukoliv metodu při řešení. Tak si mohli program dobře prohlédnout.

Díky skutečnosti, že ani u jednoho počítače nebyl jen jeden žák, mohli dvojice pracovat kolektivně, a to tak, že jeden žák obsluhoval klávesnici a druhý počítal na pomocný papír dílčí výpočty. To v mnoha případech vedlo k daleko rychlejším odpovědím než na ZŠ ve Dvoře Králové, kde u každého počítače byl pouze jeden žák.

V posledních patnácti minutách vyučovací hodiny byli všichni rozděleni do dvou skupin tak, že každá měla za úkol řešit soustavy lineárních rovnic pomocí jiné početní metody. Na každém počítači se měly spočítat dva příklady přidělenou metodou. V závěru program jednotlivé práce vyhodnotil, ale tyto známky byly i zde spíše orientační. Celá první návštěva byla brána jako seznámení s výukovým programem.

Práce na této první návštěvě neprobíhala s vážnějšími problémy. U jednotlivých počítačů se i při prvním setkání s programem dvojice dokázala dohodnout a spolupracovat. Nebylo příliš nutné individuálně vysvětlovat úplně vše, jako tomu bylo na ZŠ Strž ve Dvoře Králové u skupiny dívek.

5.3.2.2 Druhá návštěva

Po dvoudenní přestávce jsme žákům stejné třídy předložili opět při hodině matematiky program „Soustavy lineárních rovnic". Program i metody jsme jen připomněli a opět jsme vytvořili dvě skupiny, které měly za úkol vypočítat za dvacet pět minut pět příkladů námi zvolenou metodou pro každou skupinu. První dvojici žáků, která správně vypočítá náhodně vybraných pět soustav lineárních rovnic bez vážnějších chyb a která bude programem ohodnocena na ***výbornou***, byl opět slíben zápis do žákovské knížky od jejího učitele matematiky. Mezi žáky tohoto osmého ročníku bez jakéhokoliv přispění učitele vznikla soutěž o nejlepší známku přidělenou počítačem. Každá dvojice se chlubila sousedům svou známkou po dokončení výpočtů.

Až na pár výjimek se podařilo spočítat náhodně vybrané příklady předepsanou metodou ve stanoveném čase. Po dokončení těchto pěti příkladů program stanovil známku a žáci dostali za úkol počítat dalších pět příkladů druhou metodou. Bylo jasné, že celkové množství už nestihnou. Ti rychlejší stihli dokonce tři z pěti, ale problémoví žáci nestihli dopočítat ani prvních pět příkladů prvního zadání. Těm bylo nutné s počítáním pomáhat. Pomoc se netýkala obsluhy programu, ale vlastního způsobu počítání soustav lineárních rovnic. Tato látka ještě nebyla zcela zažitá a někteří měli s počítáním rovnic jisté problémy.

Během celého zkoušení bylo jasně vidět, že i zde práce třídy na tomto programu je daleko úspěšnější než první seznamování, které proběhlo před dvěma dny.

5.3.2.3 Desetiminutovka

Od třídy 8.B nám bylo umožněno použít výsledky desetiminutové práce ze tří příkladů a porovnat je s našimi výsledky při práci na počítači s podobnými příklady. Toto porovnání jsme chtěli provést hlavně pro posouzení výhod a nevýhod našeho výukového programu.

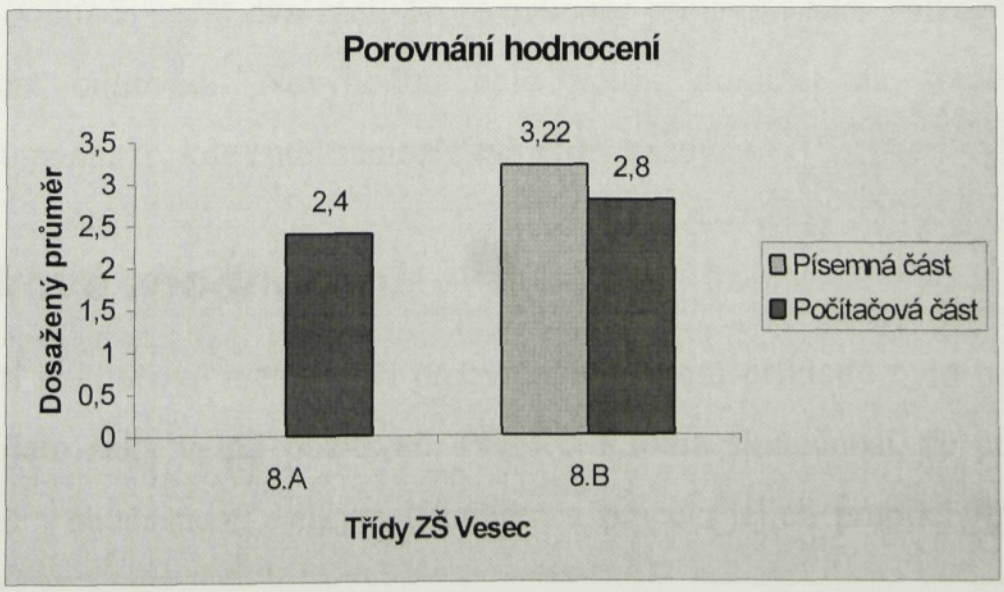
V této osmé třídě se písemné práce zúčastnilo devatenáct žáků. Zadány byly tři soustavy rovnic o dvou neznámých, ze kterých mohl každý žák získat patnáct bodů. Za každý příklad bylo maximum pět bodů.

Známky dle bodového hodnocení :

1	15 b	- 13 b
2	12 b	- 11 b
3	10 b	- 6 b
4	5 b	- 3 b
5	2 b	- 0 b

Výsledný známkový průměr v této desetiminutovce činil Ø 3.22 (několik přehledných prací je uvedeno v příloze č. 24-27).

Při práci na počítačích bylo dosaženo při druhých návštěvách lepších průměrů než v této písemné práci, a to průměru Ø 2.4 u 8.A a Ø 2.8 u 8.B (viz Graf č.1).



Graf č.1

Známkové hodnocení bylo lepší při použití počítače, ale rozdíl nebyl tak výrazný. To svědčilo o poctivém přístupu žáků k práci s naším programem, kdy se snažili dosáhnout nejlepších známek přidělených počítačem, i když neměly vliv na jejich konečnou známku z matematiky. K vyrovnanosti přispěl také fakt, že se procvičovaly pouze početní metody, protože grafická metoda nebyla ještě probrána. U grafické metody by byly patrnější rozdíly, které by odpovídaly větší náročnosti při rýsování na papír před automatickým zobrazováním na monitoru. Oproti tomu by pomocí programu bylo sice možné zvládnou více příkladů, ale zase by se neprocvičilo vlastní rýsování, které se žáci samozřejmě naučit musí.

Při práci pomocí výukového programu se žák nemůže dostat do tzv. „slepé uličky" tím, že by použil špatný postup nebo pokračoval ve výpočtech s nějakou početní či postupovou chybou. Oproti tomu musí dodržovat postup, který mu program předkládá a s použitím pomocných výpočtů musí vyplňovat označené rámečky.

5.3.2.4 Hodnocení

V obou skupinách osmých tříd žáci pracovali přibližně stejně. Nebyly zde vidět tak velké rozdíly, jako při dělení na chlapce a dívky na ZŠ ve Dvoře Králové. Vzhledem k tomu, že se jednalo o celou třídu matematiky, museli vždy u jednoho počítače sedět dva žáci. To se nakonec projevilo jako velká výhoda pro rychlost odpovědí. Nevýhodou bylo pouze dohlížet na spravedlivé vystřídání u počítače, kde chtěl samozřejmě sedět každý.

5.4 Celkové zhodnocení

Použití počítačové metody při procvičování řešení příkladů bylo na obou školách přijato žáky velmi pozitivně. Přispěla k tomu skutečnost, že počítače jsou pro ně v současnosti velkým lákadlem a práce s jejich pomocí je velmi baví.

U početních metod se ukázalo, že i při velice jednoduchých soustavách lineárních rovnic většina žáků byla nucena používat na všechny pomocné výpočty poznámkový blok. To mělo za následek velké zpomalení jejich odpovědí. Skutečnost, že v některých případech odpovídali až po celkovém vyřešení soustavy na papíře, vedla k zamyšlení ke zlepšení programu do budoucna. Žáci by měli mít větší volnost při výpočtech. Místo poznámkového bloku by měli všechny své výpočty uvádět přímo na obrazovce počítače. Program by pouze upozorňoval na početní chyby nebo by v případě nutnosti napověděl další krok výpočtu. Tím by mohl být libovolný postup výpočtu a záleželo by pouze na žáku, jaký zvolí.

Při porovnávání mezi klasickou metodou počítání na papíře a odpovídáním pomocí počítače je zřejmé, že při současném stavu programu je použití tužky a papíru daleko vhodnější.

U ***početní*** metody žáky zdržují výpočty na pomocný papír a u ***grafické*** metody sice přehledně vidí řešení soustavy lineárních rovnic, ale konstrukci přímek za ně rýsuje počítač. Tím se neprocvičuje vlastní rýsování, ale jen správnost postupu řešení grafickou metodou. V době testování našeho programu nebyla ještě grafická metoda v osmých ročnících probrána, a tak její procvičení nebylo možné. Při použití grafické metody mají žáci větší možnost projít více příkladů, protože práce na počítači je daleko rychlejší a přesnější. Grafy lineárních rovnic sestrojuje program sám na základě dvou libovolných zadaných bodů a na žácích je stanovit jejich souřadnice a nakonec i vlastní řešení a jejich počet. Navíc je zde okamžitě a velice názorně vidět vazba mezi vzájemnou polohou přímek a počtem řešení. Grafy obou přímek jsou přehledně zobrazeny na monitoru v různých barvách společně se všemi body, které byly použity. Je zde samozřejmě i celá přesně formulovaná odpověď.

Program umožňuje žákům pracovat samostatně podle svých předpokladů s minimální pomocí učitele. To by mělo přispět k většímu procvičení a

prohloubení látky u každého jednotlivce. Procvičení pomocí programu je tedy vhodné zavést do výuky této látky, ale samozřejmě pouze jako doplněk. Žáci hlavně potřebují umět počítat a rýsovat pomocí tužky a papíru. Zde se nejvíce uplatní úloha učitele, který vše vysvětlí a názorně předvede přesně v závislosti na stupni nálady a chápání svých žáků. Toho program nemůže být nikdy schopen.

6. Závěr

Přidání tohoto zcela odlišného metodického postupu do výuky v matematice by v budoucnu určitě vedlo k lepšímu využití času při procvičování početních řešení soustav lineárních rovnic o dvou neznámých a u grafické metody k lepšímu pochopení grafického řešení společně s názorným stanovením počtu řešení s použitím grafu. V současnosti se zavádění počítačů na základních školách velice rozšiřuje, a proto je tato možnost snadno řešitelná.

Program je samozřejmě nutné doladit podle konkrétních požadavků, které vyplývají při jeho zkoušení. U početních metod by bylo například vhodné zavést již zmíněné úpravy, které by vedly k libovolnému postupu řešení místo používání poznámkového bloku. Dále by bylo vhodné přidat do programu grafickou část, která by ukazovala, jak se mění pozice přímky v závislosti na tvaru rovnice. Žáci by měli možnost sami přímkou otáčet či posouvat v souřadném systému pomocí počítačové myši. Tím by program mohl sloužit i jako názorná pomůcka při výuce nové látky.

Tato diplomová práce vedla k vyzkoušení používání počítačů jako nového doplňujícího prvku ve výuce. Zatím není možné tento druh procvičování zařadit, ale vzhledem k rychlému nástupu veškeré počítačové techniky lze toto zařazení v dohledné době předpokládat.

Je samozřejmé, že výukový program je nutné řádně vyzkoušet ve více hodinách a pomocí většího počtu žáků, aby se dostatečně projevily jeho výhody či naopak nevýhody.

7. Seznam pramenů

[BKM] Bušek, I. - Kubínová, M. - Novotná, J.: Matematika 9 (I.díl). Praha, Prometheus 1994.

[BKS] Bušek, I. - Kubínová, M. - Novotná J.: Sbírka úloh z matematiky 9. Praha, Prometheus 1995.

[BMS] Bušek, I. - Macháček, V. - Kotlík, B. - Tichá, M.: Sbírka úloh z matematiky 8. Praha, Prometheus 1996.

[Kin] Kindl, K.: Matematika (přehled učiva ZŠ). Praha, SPN 1972.

[Kow] Kowal, S. - překlad Jarník, J.: Matematika pro volné chvíle. Praha, SNTL 1975.

[Mül] Mülerová, J. - Čižmár, J. - Divíšek, J. - Macháček, V.: Matematika 7 (II. díl). Praha, Prometheus 1994.

[Mra] Mrázek, J.: Taje matematiky. Praha, Práce 1986.

[Peš] Pešková, E. - Mulačová, J.: Matematika (přehled středoškolského učiva). Praha, Orfeus 1992.

[Pol] Polák, J.: Přehled středoškolské matematiky. Praha, SPN 1983.

[Šed] Šedivý, O. - Križalkovičová, M. - Macháček, V. - Židek, S.: Matematika 8 (I.díl). Praha, SPN 1991.

[Vil] Vild, J.: Metodické pokyny k diplomovým a závěrečným pracím na Pedagogické fakultě TUL. Liberec, TUL 1995.

```
program vyuka;
uses dos,graph,crt,gmouse,obj_unit;
type text80  = string[80];
     rec_rov = record
        A:real;
        B:real;
        C:real;
     end;
     typ_rov   = array[1..2] of rec_rov;
     typ_reseni                = array[1..3] of real;
const
    gr_cesta='';
    konfigurak='.\config.kfg';
    data='.\priklady.dat';
    stin=7;
    b_sti=darkgray;
    b_tex=lightcyan;
    b_nad=lightgreen;
    b_zak_tex=lightgray;
    b_ota=red;
    b_hot=black;
    b_ram=white;
    b_men=blue;
    b_pap=lightgray;
    b_err=lightred;
    b_war=red;
    b_mes=yellow;
    ram_podklad=1;
    scr_podklad : fillpatterntype=($22,$66,$cc,$9c,$b9,$b9,$b1,$23);
    scr_menu    : FillPatternType=($00,$00,$00,$00,$00,$00,$00,$00);
    font1=1;
    font2=0;
    font3=2;
    pocet_soustav_m=100;
    pocet_objektu=10;
var
  mys_ok:boolean;
  tlacitko,pocet_soustav_a:integer;
  x_max,y_max:integer;
  vyska_pisma,sirka_pisma:integer;
  sirka_koef:byte;
  Podklad: array [0..5] of Pointer;     {Podklad menu}
  rov:array[1..pocet_soustav_m] of typ_rov;
  posloup:array[1..pocet_soustav_m] of integer;
  cis_prik:array[0..1] of integer;
  reseni:array[1..pocet_soustav_m] of typ_reseni;
  p_objekt: array[1..pocet_objektu] of frame;
  oldsetup: TextSettingsType;
  oldcolor,oldbkcolor:word;
  oldFill : FillSettingsType;
  chyby:real;
  konec,konec_odp:boolean;
  code,pocet_prikl,delka_odpov:integer;
  start_sec,konec_sec:longint;
procedure ramecek(x1,y1,x2,y2,b_p,druh:integer);forward;
procedure disp(x,y:integer;what:text80;b_t:byte);forward;
procedure CEKANI;forward;
procedure Warning(text:text80);forward;
function MENU(p_radku:byte;x1,y1,x2,y2:integer):byte;forward;
{*************************************************************************
******************** P R O C E D U R Y   A  F U N K C E ********************
*************************************************************************}
procedure SAVE_SETUP;
begin
  GetFillSettings(oldFill);                          { Uložení vzorku a barvy }
  GetTextSettings(Oldsetup);
  oldcolor:=getcolor;
  oldbkcolor:=getbkcolor;
end;
procedure LOAD_SETUP;
begin
  with Oldsetup do
  begin                                                                { Obnovení starého
stylu textu }
    SetTextJustify(Horiz, Vert);
    SetTextStyle(Font, Direction, CharSize);
  end;
  with oldFill do
              SetFillStyle(Pattern, Color);   { Obnovení starého vyplňovacího stylu }
  setcolor(oldcolor);
  setbkcolor(oldbkcolor);
end;
procedure VYBER;
var i,ii,cislo:integer;
    znova:boolean;
begin
  randomize;
  for i:=1 to pocet_prikl do begin
```

```
    repeat
      znova:=false;
            cislo:=random(pocet_soustav_a)+1;
            for ii:=1 to pocet_prikl do
                                  if posloup[ii]=cislo then znova:=true;
    until (NOT znova);
    posloup[i]:=cislo;
  end;
end;
procedure KONEC_CASU;
var  h, m, s, hu : Word;
     psetup: TextSettingsType;
     pcolor,pbkcolor:word;
     pFill : FillSettingsType;
begin
  if konec_odp then exit
  else begin
  GetFillSettings(pFill);                                  { Uložení vzorku a barvy }
  GetTextSettings(psetup);
  pcolor:=getcolor;
  pbkcolor:=getbkcolor;
  GetTime(h,m,s,hu);
  konec_sec:=(s+m*60+h*3600);
  setcolor(7);
  SetFillStyle(1,0);
{     Bar3D(0,0,7,trunc((480/(delka_odpov*60))*(konec_sec-start_sec)),0,Topoff);
}     Bar3D(631,0,639,trunc((480/(delka_odpov*60))*(konec_sec-start_sec)),0,Topoff);
  if (konec_sec>(start_sec+delka_odpov*60)) AND (NOT konec_odp) then begin
    KONEC_ODP:=true;
  end;
  with psetup do
  begin                                                                    { Obnovení starého
stylu textu }
    SetTextJustify(Horiz, Vert);
    SetTextStyle(Font, Direction, CharSize);
  end;
  with pFill do
  SetFillStyle(Pattern, Color);   { Obnovení starého vyplňovacího stylu }
  setcolor(pcolor);
  setbkcolor(pbkcolor);
  end;
end;
procedure START_CASU;
var  h, m, s, hu : Word;
begin
  GetTime(h,m,s,hu);
  start_sec:=(s+m*60+h*3600);
  SAVE_SETUP;
  SetFillStyle(1,4);
  setcolor(b_ram);
  Bar3D(631,0,639,479,0,Topoff);
  LOAD_SETUP;
  konec_odp:=false;
end;
function START_OK:boolean;
var v:byte;
begin
  ramecek(x_max div 7,y_max div 4,(x_max div 7)*6,(y_max div 4)*3,b_men,2);
  p_objekt[1].init(x_max div 2,y_max div 2-30,250,20,'Start dalšího příkladu',1,14,b_men);
  p_objekt[2].init(x_max div 2,y_max div 2,250,20,'KONEC - Zobrazit výsledky',1,14,b_men);
  v:=MENU(2,195,200,445,280);
  if v=1 then START_OK:=true
  else        START_OK:=false;
end;
function PREVOD(vstup:real;druh:boolean):string;
var znam:string;
    p1,pom:byte;
    code:integer;
    pom2:longint;
    s_pr,s_p:string;
begin
  str(abs(trunc(int(vstup))):sirka_koef,s_pr);
  str(frac(vstup),s_p);
  p1:=pos('.',s_p);
  s_p:=copy(s_p,1,p1-1)+copy(s_p,p1+1,length(s_p));
  while pos(' ',s_p)<>0 do delete(s_p,pos(' ',s_p),1);
  if vstup<0 then begin
     s_p:=copy(s_p,2,length(s_p));
     znam:='-';
  end
  else if druh then znam:='+' else znam:=' ';
  p1:=pos('-',s_p);
  val(copy(s_p,p1+1,2),pom,code);
  s_p:=copy(s_p,1,5);
  val(s_p,pom2,code);
  pom2:=round(pom2/1000);
  str(pom2,s_p);
  while pom>1 do begin
```

```
      s_p:='0'+s_p;
      dec(pom);
    end;
    if length(s_p)>1 then
         s_p:='.'+s_p;
    while copy(s_p,length(s_p),1)='0' do delete(s_p,length(s_p),1);
    s_pr:=znam+s_pr+s_p;
    while pos(' ',s_pr)<>0 do delete(s_pr,pos(' ',s_pr),1);
    prevod:=s_pr;
  end;
  function ZAO(vstup:real):real;
  begin
    vstup:=vstup*100;
    vstup:=round(vstup);
    ZAO:=(vstup)/100;
  end;
  procedure VYSLEDKY(kolik_prikl:integer);
  var znamka:real;
      ch:char;
  begin
    SAVE_SETUP;
    ramecek(x_max div 7,y_max div 4,(x_max div 7)*6,(y_max div 4)*3,b_men,2);
    settextstyle(font2,horizdir,2);
    disp(140,150,'Výsledky všech zadání',b_ram);
    settextstyle(font2,horizdir,1);
    if konec_odp then
         disp(110,200,'Došel povolený čas k vypracování .....',b_ram);
    konec_odp:=true;
    disp(110,230,'Počet použitých příkladů                    z',b_ram);
    disp(110,260,'Celkový dosažený čas [v min]   ',b_ram);
    disp(110,290,'Trestné body za chyby ',b_ram);
    disp(110,320,'Návrh známky za tento výkon    ',b_ram);
    znamka:=(5/pocet_prikl*(pocet_prikl-kolik_prikl+chyby));
    if znamka=0 then znamka:=znamka+1;
    znamka:=round(znamka);
    settextstyle(font2,horizdir,1);
    disp(420,230,prevod(kolik_prikl,false),b_err);
    disp(490,230,prevod(pocet_prikl,false),b_err);
    disp(460,260,prevod((konec_sec-start_sec) div 60,false),b_err);
    disp(460,290,prevod(zao(chyby),false),b_err);
    settextstyle(font2,horizdir,4);
    disp(430,350,prevod(znamka,false),b_err);
    LOAD_SETUP;
    repeat
       ch:=readkey;
       if ch=#0 then begin
          ch:=readkey;
          if ord(ch)=16 then exit;
       end;;
    until false;
  end;
  procedure PR_INIT;
  var x:integer;
  begin
    for x:=1 to pocet_soustav_m do begin
      rov[x,1].a:=-0;
      rov[x,1].b:=-0;
      rov[x,1].c:=-0;
      rov[x,2].a:=-0;
      rov[x,2].b:=-0;
      rov[x,2].c:=-0;
      reseni[x,1]:=-0;
      reseni[x,2]:=-0;
      reseni[x,3]:=-0;
      inc(x);
    end;
    for x:=1 to pocet_objektu do begin
          p_objekt[x].init(0,0,0,0,'',0,0,0);
    end;
    sirka_koef:=3;
    konec_odp:=true;
  end;
  procedure GR_INIT;
  var gd,gm:integer;
    code,x1,y1,x2,y2:integer;
  begin
    gD :=DETECT;
    InitGraph(Gd, Gm,gr_cesta);
    if GraphResult <> grOk then Halt(1);
    x_max:=getmaxx;
    y_max:=getmaxy;
    if (x_max<>639) AND (y_max<>479) then      write('Chyba rozliseni .....');
  end;
  PROCEDURE CLEARSCREEN(druh:byte);
  begin;
    if druh=1 then       setfillpattern(scr_podklad,b_pap)
    else                                        setfillstyle(1,b_men);
    setcolor(b_ram);
```

```
{       bar3d(8,0,x_max,y_max,0,topoff);
}       bar3d(0,0,x_max-11,y_max,0,topoff);
end;
procedure SMAZ(x1,x2,y1,y2,druh:integer);
begin
  SAVE_SETUP;
  if druh=1 then       setfillpattern(scr_podklad,b_pap)
  else                                          setfillstyle(1,b_men);
  setcolor(b_men);
       bar3d(X1,X2,Y1,Y2,0,topoff);
  LOAD_SETUP;
end;
procedure HLAVICKA;
var ch:char;
begin
  clearscreen(1);
  settextjustify(centertext,centertext);
  ramecek(x_max div 7,y_max div 4,(x_max div 7)*6,(y_max div 4)*3,b_men,1);
  settextstyle(font1,horizdir,4);
  disp(x_max div 2,30+y_max div 3,'Soustavy lineárních rovnic',b_nad);
  settextstyle(font2,horizdir,2);
  disp(x_max div 2,90+y_max div 3,'výukový program pro ZŠ',b_nad);
  settextstyle(font3,horizdir,4);
  disp((y_max div 6)*3,(y_max div 7)*5 ,'TU LIBEREC (c) 1997 verze 1.1',b_tex);
  mys_ok:=initmouse;
  if not(mys_ok) then begin
     warning('  Nenašel jsem připojenou myš !   ');end;
  CEKANI;
       settextstyle(0,0,0);
       settextjustify(lefttext,bottomtext);
       setwritemode(normalput);
       vyska_pisma:=textheight('X');
       sirka_pisma:=textwidth('X');
end;
procedure PRAC_PLOCHA(druh:byte);
begin
  SAVE_SETUP;
  clearscreen(1);
  settextjustify(centertext,centertext);
  ramecek(20,20,x_max-20,y_max-20,b_men,1);
  settextstyle(font1,horizdir,4);
  disp(x_max div 2,35,' Soustavy lineárních rovnic ',b_nad);
       settextstyle(font2,horizdir,1);
  case druh of
       1: disp(x_max div 2,65,' Metodou DOSAZOVACÍ vyřešte následující soustavu',b_nad);
       2: disp(x_max div 2,65,' Metodou SČÍTACÍ vyřešte následující soustavu',b_nad);
       3: disp(x_max div 2,65,' GRAFICKOU metodou vyřešte následující soustavu',b_nad);
  end;
  LOAD_SETUP;
end;
procedure RAMECEK(x1,y1,x2,y2,b_p,druh:integer);
begin;
  if druh=1 then setfillpattern(scr_podklad,b_sti)
  else                                setfillpattern(scr_menu,b_sti);
  bar(x1+stin,y1+stin,x2+stin,y2+stin);
  setfillstyle(ram_podklad,b_p);
  setlinestyle(solidln,0,1);
  setcolor(b_ram);
  Bar3d(x1,y1,x2,y2,0,topoff);
  Bar3d(x1+2,y1+2,x2-2,y2-2,0,topoff);
end;
procedure DISP(x,y:integer;what:text80;b_t:byte);
begin
 setcolor(b_t);
 outtextxy(x,y,what);
end;
procedure CEKANI;
var ch:char;
begin
  place(0,0);
  curson;
  repeat
  tlacitko:=mBUtton;
  KONEC_CASU;
  until keypressed or (tlacitko<>0);
  cursoff;
  if keypressed then begin
     ch:=readkey;
     while keypressed do ch:=readkey;
  end
  else tlacitko:=0;
end;
procedure BEEP(druh:integer);
begin;
 case druh of
   1: begin;sound(280);delay(200);nosound;end;
   2: begin;sound(800);delay(200);nosound;end;
  end;
```

```
end;

procedure WARNING(text:text80);
var vyska,delka:integer;
    pom1,pom2:integer;
    size:word;
begin;
  SAVE_SETUP;
  settextjustify(centertext,centertext);
        settextstyle(font2,horizdir,1);
        delka:=length(text)*textwidth('X')+2;
        vyska:=textheight('X')+2;
        size:=ImageSize((x_max-delka)div 2,(x_max div 2)-vyska,((x_max-delka)div
2)+delka+stin,((y_max div 2)+vyska)+stin);
  GetMem(Podklad[1],size);
        pom1:=(x_max-delka) div 2;
        pom2:=(y_max div 2) - vyska;
        GetImage(pom1,pom2,pom1+delka+stin,((y_max div 2)+vyska)+stin,podklad[1]^);
        ramecek((x_max-delka)div 2,(y_max div 2)-vyska,((x_max-delka)div 2)+delka,(y_max div
2)+vyska,b_war,2);
        disp (getmaxx div 2,getmaxy div 2,text,b_ram);
  CEKANI;
        settextstyle(0,0,0);
        PutImage((getmaxx-delka)div 2,((getmaxy div 2)-vyska), Podklad[1]^,
NormalPut);release(podklad[1]);
        LOAD_SETUP;
end;
procedure ZOBR_ROVNICI(xx,yy,priklad,poradi:integer);
var pstr:string[5];
    text:text80;
    znamenko:string[2];
    i:real;
begin
  text:='';
  if (rov[priklad,poradi].a<0) then znamenko:='-'
  else
                znamenko:=' ';
  i:=abs(rov[priklad,poradi].a);
  if int(i)=i then
          str(i:sirka_koef:0,pstr)
  else
          str(i:sirka_koef:2,pstr);
  text:=text+znamenko+pstr+'x ';
  if (rov[priklad,poradi].b<0) then znamenko:='- '
  else
znamenko:='+ ';
  i:=abs(rov[priklad,poradi].b);
  if int(i)=i then
          str(i:sirka_koef:0,pstr)
  else
          str(i:sirka_koef:2,pstr);
  text:=text+znamenko+pstr+'y =';
  if (rov[priklad,poradi].c<0) then znamenko:=' -'
  else
znamenko:='  ';
  i:=abs(rov[priklad,poradi].c);
  if int(i)=i then
          str(i:sirka_koef:0,pstr)
  else
          str(i:sirka_koef:2,pstr);
  text:=text+znamenko+pstr;
        disp(xx,yy,text,b_ram);
end;
procedure ZOBR_VYSLEDEK(xx,yy,priklad:integer);
var ps:string;
begin
  str(trunc(reseni[priklad,2]):sirka_koef,ps);
        disp(xx,yy,ps,b_ram);
  str(trunc(reseni[priklad,3]):sirka_koef,ps);
        disp(xx+50,yy,ps,b_ram);
end;
function NACTENI_ROVNIC:boolean;
var F: text;
  radek:string;
  pos1,pos2,pos3,pos4,pos5,pos6:byte;
  kod:integer;
  koef:string;
        function FileExists(FileName: String): Boolean;
        begin
          {$I-}
          Assign(F, FileName);
          FileMode := 0;
          Reset(F);
          {$I+}
          FileExists := (IOResult = 0) and (FileName <> '');
        end;  { FileExists }

begin
```

```
  if (NOT FileExists(data)) then begin
          warning(' Soubor příkladů nebyl nenalezen  ');
     nacteni_rovnic:=FALSE;
     exit;
  end;
  pocet_soustav_a:=0;
       while not Eof(F) do   begin
         Readln(F,radek);
           pos1:=pos('x',radek);
           pos2:=pos('y',radek);              { cislo rovnice do pole 1..soustav dve rovnice}
           pos3:=pos('=',radek);
           pos4:=pos('*',radek);
           pos5:=pos(':',radek);
           pos6:=pos('R',radek);
               while Pos(' ',radek) > 0 do  radek[Pos(' ',radek)] := '0';
               if pos4=1 then begin
                       val(copy(radek,pos4+1,length(radek)),cis_prik[0],kod);
      inc(pocet_soustav_a);
      cis_prik[1]:=0;
    end
    else
      if (pos1<>0) AND (pos2<>0) AND (pos3<>0) then begin
         cis_prik[1]:=cis_prik[1]+1;
         koef:=copy(radek,1,pos1-1);
         if length(koef)>sirka_koef then
                                              sirka_koef:=length(koef);
         val(koef,rov[cis_prik[0],cis_prik[1]].a,kod);
         koef:=copy(radek,pos1+1,pos2-1-pos1);
         if length(koef)>sirka_koef then
                                              sirka_koef:=length(koef);
         val(koef,rov[cis_prik[0],cis_prik[1]].b,kod);
         koef:=copy(radek,pos3+1,length(radek));
         if length(koef)>sirka_koef then
                                              sirka_koef:=length(koef);
         val(koef,rov[cis_prik[0],cis_prik[1]].c,kod);
      end;
        end;
  Close(F);
  nacteni_rovnic:=TRUE;
end;

function RESENI_ROV(c:integer):byte;
       function DET(a11,a12,a21,a22:real):real;
       begin
         DET:=a11*a22-a12*a21;
       end;
begin
  if DET(rov[c,1].a,rov[c,1].b,rov[c,2].a,rov[c,2].b)<>0 then begin
    if (det(rov[c,1].c,rov[c,1].b,rov[c,2].c,rov[c,2].b)=0) AND
(det(rov[c,1].a,rov[c,1].c,rov[c,2].a,rov[c,2].c)=0) then begin
                 reseni[c,1]:=9;
                 reseni[c,2]:=0;
                 reseni[c,3]:=0;          {NEKONECNE RESENI}
    end else begin
                 reseni[c,1]:=1;

reseni[c,2]:=ZAO(det(rov[c,1].c,rov[c,1].b,rov[c,2].c,rov[c,2].b)/DET(rov[c,1].a,rov[c,1].b,rov
[c,2].a,rov[c,2].b));

       reseni[c,3]:=ZAO(det(rov[c,1].a,rov[c,1].c,rov[c,2].a,rov[c,2].c)/DET(rov[c,1].a,rov[c,
1].b,rov[c,2].a,rov[c,2].b));
    end;
                            {JEDNO RESENI}
  end else
    if (det(rov[c,1].c,rov[c,1].b,rov[c,2].c,rov[c,2].b)=0) AND
(det(rov[c,1].a,rov[c,1].c,rov[c,2].a,rov[c,2].c)=0) then begin
                 reseni[c,1]:=9;
                 reseni[c,2]:=0;
                       reseni[c,3]:=0;          {NEKONECNE RESENI}
    end
    else begin
          reseni[c,1]:=0;
                 reseni[c,1]:=0;          {ZADNE RESENI}
                       reseni[c,2]:=0;
    end;
end;
procedure MYS_PLOCHA(x1,y1,x2,y2:integer);
var x,y:integer;
begin
 where(x,y);
 if (x<x1) then place(x1,y);
 if (x>x2) then place(x2,y);
 if (y<y1) then place(x,y1);
 if (y>y2) then place(x,y2);
end;
procedure ZOBR_SOUSTAVY(x,y:integer);
var i:integer;
begin
```

```
  i:=1;
  while i<=pocet_soustav_a do begin
          ZOBR_ROVNICI(x,y+2*(i-1)*10,i,1);
          ZOBR_ROVNICI(x,y+2*(i-1)*10+10,i,2);
          RESENI_ROV(i);
          ZOBR_VYSLEDEK(x+350,y+2*(i-1)*10+10,i);
    inc(i);
  end;
end;
function MENU(p_radku:byte;x1,y1,x2,y2:integer):byte;
var znak:char;
    pocet,citac,ocislo,cislo:byte;
    konec:boolean;
begin
  for citac:=1 to p_radku do p_objekt[citac].show;
  cislo:=1;
  ocislo:=1;
  p_objekt[cislo].aktiv:=true;
  p_objekt[cislo].zvyraz;
  place(p_objekt[cislo].x,p_objekt[cislo].y);
  konec:=false;
  curson;
  repeat
     MYS_PLOCHA(x1,y1,x2,y2);
     KONEC_CASU;
       if keypressed then begin
         cursoff;
         znak:=readkey;
        case znak of
            #27 : begin
                      konec:=true;
                      menu:=0;
               end;
            #13 : begin
                      konec:=true;
                      menu:=cislo;
               end;
             #0 : begin
                      znak:=readkey;
                      case znak of
                       'P' : begin;
                             inc(cislo);
                             if cislo>p_radku then cislo:=p_radku;
                              end;            {dolu}
                       'H' : begin;
                             dec(cislo);
                             if cislo<1 then cislo:=1;
                              end;            {nahoru}
                  end;
                  end;
         end;
         curson;
       end;
     for citac:=1 to p_radku do
        if (p_objekt[citac].ismouse) then begin
                      menu:=citac;
                      konec:=true;
       end;
     if cislo<>ocislo then begin
              p_objekt[cislo].aktiv:=true;
              p_objekt[cislo].zvyraz;
              p_objekt[ocislo].aktiv:=false;
              p_objekt[ocislo].show;
              ocislo:=cislo;
              place(p_objekt[cislo].x,p_objekt[cislo].y);
        end;
        repeat
              pocet:=0;
              for citac:=1 to p_radku do
                      if p_objekt[citac].aktiv then begin
                              inc(pocet);
                              cislo:=citac;
                      end;
               if pocet>1 then begin
                  p_objekt[ocislo].show;
             p_objekt[ocislo].aktiv:=false;
          end
        until pocet=1;
       until konec;
  cursoff;
  for citac:=1 to p_radku do p_objekt[citac].hide;
end;
function ZADANI(x,y:integer;sirka:byte;s:string):real;
var ch:char;
    des,i,Code1,code2: Integer;
    zaklad,I1,I2,r:real;
    pos1:byte;
begin
```

```
  SAVE_SETUP;
  setcolor(b_ram);
  settextjustify(lefttext,toptext);
  setfillstyle(1,b_pap);
  bar3d(x,y,x+sirka_pisma*(sirka+2),y+3*vyska_pisma,0,topoff);
  if s='0' then s:=copy(s,1,length(s)-1);
  setcolor(b_hot);
  outtextxy(x+sirka_pisma+2,y+vyska_pisma+2,s);
  if (length(s)>0) AND (copy(s,1,1)='-') then zaklad:=-1
  else  zaklad:=1;
  repeat
    repeat
      if (NOT keypressed) then KONEC_CASU;
      if (NOT keypressed) then delay(50);
                        setcolor(0);
            outtextxy(x+((length(s)+1)*sirka_pisma),y+vyska_pisma+2,'_');
      if NOT keypressed then KONEC_CASU;
      if (NOT keypressed) then delay(50);
                        setcolor(15);
            outtextxy(x+((length(s)+1)*sirka_pisma),y+vyska_pisma+2,'_');
    until keypressed;
    ch:=readkey;
    if (((ord(ch)=13) OR (ord(ch)=27))AND (length(s)=0)) then s:='0';
    if ((ord(ch)>47) AND (ord(ch)<58) OR (ord(ch)=46)) AND (length(s)<sirka) then  s:=s+ch;
    if (ch='-') AND (length(s)=0) then begin
              s:=s+ch;
              zaklad:=-1;
    end;
    if (ord(ch)=8) then s:=copy(s,1,length(s)-1);
          setcolor(b_ram);
          bar3d(x,y,x+sirka_pisma*(sirka+2),y+3*vyska_pisma,0,topoff);
          setcolor(b_hot);
    outtextxy(x+sirka_pisma+2,y+vyska_pisma+2,s);
    pos1:=pos('.',s);
    des:=length(s)-pos1;
    code1:=0;
    code2:=0;
    if pos1=0 then Val(copy(s,1,des),I1,Code1)
    else begin
       Val(copy(s,1,pos1-1),I1,Code1);
       if des<>0 then  Val(copy(s,pos1+1,des),I2,Code2);
      end;
    if ((code1<>0) OR (code2<>0)) AND (ord(ch)=13) then warning(' Špatně vložené číslo  ');
  until ((ord(ch)=13) AND (code1=0) AND (code2=0))OR (ord(ch)=27);
  setcolor(b_ram);
       bar3d(x,y,x+sirka_pisma*(sirka+2),y+3*vyska_pisma,0,topoff);
       setcolor(b_hot);
  outtextxy(x+sirka_pisma+2,y+vyska_pisma+2,s);
  LOAD_SETUP;
  for i:=1 to des do zaklad:=zaklad*10;
  if pos1=0 then zadani:=i1
  else           zadani:=i1+(i2/zaklad);
end;
function HL_MENU:byte;
begin
  p_objekt[1].init(x_max div 2,y_max div 2-30,250,20,'Početní rešení',1,14,b_men);
  p_objekt[2].init(x_max div 2,y_max div 2,250,20,'Grafické řešení',1,14,b_men);
  p_objekt[3].init(x_max div 2,y_max div 2+30,250,20,'Ukončení programu',1,14,b_men);
  HL_MENU:=MENU(3,195,200,445,280);
end;
procedure GRAFICKY;
var cyklus:integer;
  function GRAFIKA(cis_prikl:integer):boolean;
  type typ_bod = record
              x:real;
              y:real;
              e:boolean;
       end;
       typ_body        = array[1..4] of typ_bod;
  var i,ii,cis_rov,cis_bodu:integer;
      vstup1,vstup2:real;
      odpov1,odpov2:real;
      odpov3,odpov4:real;
      odpov5,odpov6:real;
      odpov7,odpov8:real;
      odpov0:real;
      bod :typ_body;
      volba1,volba2:byte;
      s1,s2:string;
      posuv:integer;
      krok1,krok2:integer;
      pocetx,pocety:integer;
      fce_OK:boolean;
     procedure GRAF;
     var i,ii:integer;
         krokx,kroky:integer;
         ps1:string;
     begin
```

```
      krok1:=400 div pocetx;
      krok2:=300 div pocety;
      SAVE_SETUP;
      setcolor(b_ram);
      SetTextStyle(2, HorizDir, 3);
      setfillstyle(0,black);
      bar3d(10,179,410,479,0,topoff);
      krokx:=1;
      kroky:=1;
      str(pocetx,ps1);
      while (((pocetx div 2)div krokx)*((length(ps1)+5)*6)>200) do inc(krokx);
      while (((pocety div 2)div kroky)*(11+4)>150) do inc(kroky);
      i:=210;
      ii:=1;
      repeat
        i:=i+krok1;
        if ((ii mod krokx)=0) then begin
           setcolor(b_men);
           line(i,180,i,478);
           setcolor(b_zak_tex);
              line(i,327,i,331);
           if i<400 then disp(i+3,335,prevod(ii,false),b_zak_tex);
        end;
        inc(ii);
      until (i+krok1>=410);
      i:=210;
      ii:=-1;
      repeat
        i:=i-krok1;
        if (ii mod krokx)=0 then begin
           setcolor(b_men);
           line(i,180,i,478);
           setcolor(b_zak_tex);
              line(i,327,i,331);
       if i<400 then disp(i+3,335,prevod(ii,false),b_zak_tex);
        end;
        dec(ii);
      until (i-krok1<=10);
      i:=329;
      ii:=-1;
      repeat
        i:=i+krok2;
        if (ii mod kroky)=0 then begin
           setcolor(b_men);
           line(11,i,409,i);
           setcolor(b_zak_tex);
              line(208,i,212,i);
              disp(212,i-2,prevod(ii,false),b_zak_tex);
        end;
        dec(ii);
      until (i+krok2>=479);
      i:=329;
      ii:=1;
      repeat
        i:=i-krok2;
        if (ii mod kroky)=0 then begin
           setcolor(b_men);
              line(11,i,409,i);
              setcolor(b_zak_tex);
              line(208,i,212,i);
              disp(212,i+7,prevod(ii,false),b_zak_tex);
        end;
        inc(ii);
      until (i-krok2<=179);
      setcolor(b_zak_tex);
      line(210,180,210,478);
      outtextxy(402,326,'X');
      line(11,329,409,329);
      outtextxy(212,185,'Y');
      outtextxy(204,337,'0');
      LOAD_SETUP;
    end;
    procedure PRIMKA(ax,ay,bx,by:real;b:byte);
    begin
       setcolor(b);
              line(trunc(210+ax*krok1),trunc(329-ay*krok2),trunc(210+bx*krok1),trunc(329-
by*krok2));
    end;
    procedure FCE(cis,barva:byte);
    var x,y:real;
        ax,ay,bx,by:real;
        OK:boolean;
        od1,od2:real;
              vyska,delka:integer;
        pom1,pom2:integer;
        size:word;
        text:string;
    begin
```

```
      text:=' Probíhá výpočet přímek ';
      if cis=1 then begin
                                  od1:=odpov1;
       od2:=odpov2;
      end else begin
                                  od1:=odpov3;
       od2:=odpov4;
      end;
      x:=0;
      ok:=true;
      SAVE_SETUP;
      settextjustify(centertext,centertext);
            settextstyle(font2,horizdir,1);
            delka:=length(text)*textwidth('X')+2;
            vyska:=textheight('X')+2;
            size:=ImageSize((x_max-delka)div 2,(x_max div 2)-vyska,((x_max-delka)div
2)+delka+stin,((y_max div 2)+vyska)+stin);
      GetMem(Podklad[1],size);
            pom1:=(x_max-delka) div 2;
            pom2:=(y_max div 2) - vyska;
            GetImage(pom1,pom2,pom1+delka+stin,((y_max div 2)+vyska)+stin,podklad[1]^);
            ramecek((x_max-delka)div 2,(y_max div 2)-vyska,((x_max-delka)div 2)+delka,(y_max
div 2)+vyska,b_war,2);
            disp (getmaxx div 2,getmaxy div 2,text,b_ram);
      if fce_ok then begin
         while (329-krok2*(od1*x+od2)<478) AND (329-krok2*(od1*x+od2)>182) AND (OK) do begin
                                        x:=x+0.001;
            if (210+x*krok1>408) OR (210+x*krok1<12) then OK:=false;
         end;
         ax:=x-0.001;
         ay:=(od1*x+od2);
         x:=0;
         ok:=true;
         while (329-krok2*(od1*x+od2)<478) AND (329-krok2*(od1*x+od2)>182) AND (OK) do begin
                                        x:=x-0.001;
            if (210+x*krok1>408) OR (210+x*krok1<12) then OK:=false;
         end;
         bx:=x+0.001;
         by:=(od1*x+od2);
      end else begin
         ax:=od2;
         bx:=od2;
         ay:=149/krok2;
         by:=-149/krok2;
      end;
            settextstyle(0,0,0);
            PutImage((getmaxx-delka)div 2,((getmaxy div 2)-vyska), Podklad[1]^,
NormalPut);release(podklad[1]);
            LOAD_SETUP;
      PRIMKA(ax,ay,bx,by,barva);
      case cis of
                                  1:disp(12,463,'První rovnice',barva);
        2:disp(12,470,'Druhá rovnice',barva);
      end;
    end;
    procedure GRAF_BOD(poradi:integer;soux,souy:real;b:byte);
    begin
      SAVE_SETUP;
      setfillstyle(1,0);
      setcolor(0);
       bar3d(trunc(215+soux*krok1),trunc(323-souy*krok2),trunc(221+soux*krok1),trunc(316-
souy*krok2),0,topoff);
      setcolor(b);
      setfillstyle(1,b);
      pieslice(trunc(210+soux*krok1),trunc(329-souy*krok2),0,360,2);
      if poradi=5 then outtextxy(trunc(215+soux*krok1),trunc(325-souy*krok2),'P')
      else                  outtextxy(trunc(215+soux*krok1),trunc(325-
souy*krok2),chr(64+poradi));
      LOAD_SETUP;
    end;
    procedure ZOBR_XY(co:byte;x,y:integer);
    begin
      if co=1 then
                        disp(x,y,'x =',b_ram)
      else
                        disp(x,y,'y =',b_ram);
      bar3d(x+4*sirka_pisma,y+vyska_pisma,x+(13)*sirka_pisma,y-2*vyska_pisma,0,topoff);
         if vstup1<>0 then disp(x+5*sirka_pisma,y,prevod(vstup1,false),b_ram);
      if co=1 then
            disp(x+14*sirka_pisma,y,'y ',b_ram)
      else
            disp(x+14*sirka_pisma,y,'x ',b_ram);
      bar3d(x+(16)*sirka_pisma,y+vyska_pisma,x+(25)*sirka_pisma,y-2*vyska_pisma,0,topoff);
         if vstup2<>0 then disp(x+17*sirka_pisma,y,prevod(vstup2,false),b_ram);
    end;
    procedure ZOBR_XY3(x,y:integer);
    begin
      y:=y+16;
```

```
      bar3d(x,y+vyska_pisma,x+(8)*sirka_pisma,y-2*vyska_pisma,0,topoff);
         if vstup1<>0 then disp(x+1*sirka_pisma,y,prevod(vstup1,false),b_ram);
      x:=x+100;
      bar3d(x,y+vyska_pisma,x+(8)*sirka_pisma,y-2*vyska_pisma,0,topoff);
         if vstup2<>0 then disp(x+1*sirka_pisma,y,prevod(vstup2,false),b_ram);
    end;
  begin
        RESENI_rov(cis_prikl);
    PRAC_PLOCHA(3);
        ZOBR_ROVNICI(50,100,cis_prikl,1);
        ZOBR_ROVNICI(50,110,cis_prikl,2);
    disp(480,470,prevod(cis_prikl,false),b_ram);
    line(70,115,180+sirka_koef*10,115);
    vstup1:=0;
    vstup2:=0;
    posuv:=0;
    for cis_rov:=1 to 2 do begin
       i:=1;
       if rov[cis_prikl,cis_rov].b=0 then fce_OK:=false
       else
                                fce_ok:=true;
       if fce_ok then
              if cis_rov=1 then disp(250,120+posuv,'Z první rovnice vyjádřete Y',b_mes)
              else              disp(250,120+posuv,'Z druhé rovnice vyjádřete Y',b_mes)
       else
              if cis_rov=1 then disp(250,120+posuv,'Z první rovnice vyjádřete X',b_mes)
              else              disp(250,120+posuv,'Z druhé rovnice vyjádřete X',b_mes);
       while i<3 do begin
          if fce_ok then ZOBR_XY(2,50,140+posuv)
          else                          ZOBR_XY(1,50,140+posuv);
               vstup1:=ZADANI(50+4*sirka_pisma,140+posuv-
(2*vyska_pisma),7,prevod(vstup1,false));
          vstup1:=ZAO(vstup1);
               vstup2:=ZADANI(50+16*sirka_pisma,140+posuv-
(2*vyska_pisma),7,prevod(vstup2,false));
          vstup2:=ZAO(vstup2);
          s1:=prevod(vstup1,false);
          s2:=prevod(vstup2,true);
          if fce_ok then begin
            if cis_rov=1 then begin
               odpov1:=ZAO(-1*rov[cis_prikl,cis_rov].a/rov[cis_prikl,cis_rov].b);
               odpov2:=ZAO(rov[cis_prikl,cis_rov].c/rov[cis_prikl,cis_rov].b);
            end else begin
               odpov3:=ZAO(-1*rov[cis_prikl,cis_rov].a/rov[cis_prikl,cis_rov].b);
              odpov4:=ZAO(rov[cis_prikl,cis_rov].c/rov[cis_prikl,cis_rov].b);
            end;
            s1:='y = '+s1+'x ';
          end else begin
            if cis_rov=1 then begin
               odpov1:=0;
              odpov2:=ZAO(rov[cis_prikl,cis_rov].c/rov[cis_prikl,cis_rov].a);
            end else begin
               odpov3:=0;
              odpov4:=ZAO(rov[cis_prikl,cis_rov].c/rov[cis_prikl,cis_rov].a);
            end;
            s1:='x = '+s1+'y ';
          end;
          if vstup2>0 then   s1:=s1+s2
          else               s1:=s1+s2;
          SMAZ(25,140+posuv+vyska_pisma,50+35*sirka_pisma,140+posuv-(2*vyska_pisma),0);
                    disp(50,140+posuv,s1,b_ram);
          if (((vstup1<>odpov1) OR (vstup2<>odpov2)) AND (cis_rov=1))
               OR (((vstup1<>odpov3) OR (vstup2<>odpov4)) AND (cis_rov=2)) then begin
             if i=1 then       begin
               warning(' Špatně ... Zkuste to znovu ');
               SMAZ(25,140+posuv+vyska_pisma,50+35*sirka_pisma,140+posuv-(2*vyska_pisma),0);
             end else begin
               warning(' Zase špatně ... ');
               if fce_ok then
                  if cis_rov=1 then s1:='y = '+prevod(odpov1,false)+'x '+prevod(odpov2,true)
                  else          s1:='y = '+prevod(odpov3,false)+'x '+prevod(odpov4,true)
               else
                  if cis_rov=1 then s1:='x = '+prevod(odpov1,false)+'y '+prevod(odpov2,true)
                  else        s1:='x = '+prevod(odpov3,false)+'y '+prevod(odpov4,true);
               disp(400,130+posuv,'Správné vyjádření je :',b_err);
                          disp(400,140+posuv,s1,b_tex);
               chyby:=chyby+0.15;
               cekani;
             end;
          end else i:=3;
          inc(i);
       end;
       SMAZ(25,140+posuv+vyska_pisma,50+35*sirka_pisma,140+posuv-(2*vyska_pisma),0);
       disp(50,140+posuv,s1,b_ram);
       posuv:=posuv+35;
       vstup1:=0;
       vstup2:=0;
    end;
```

```
    cis_rov:=1;
    pocety:=6;
    pocetx:=6;
    posuv:=0;
    cis_bodu:=0;
    GRAF;
    repeat
      if rov[cis_prikl,cis_rov].b=0 then fce_OK:=false
      else                               fce_ok:=true;
      repeat
        inc(cis_bodu);
         i:=1;
         vstup1:=0;
         vstup2:=0;
         while i<3 do begin
               SMAZ(420,190+posuv,615,450,0);
                      disp(420,200+posuv,'Zadejte souřadnice',b_mes);
            if (cis_bodu mod 2)<>0 then disp(420,210+posuv,'libovolného PRVNÍHO bodu,',b_mes)
            else                        disp(420,210+posuv,'libovolného DRUHÉHO bodu,',b_mes);
                      if cis_rov=1 then disp(420,220+posuv,'vyhovující PRVNÍ rovnici',b_mes)
            else                disp(420,220+posuv,'vyhovující DRUHÉ rovnici',b_mes);
                      disp(430,240+posuv,'x-ová        y-ová',b_mes);
            zobr_xy3(420,250+posuv);
            vstup1:=zadani(420,250+posuv,6,prevod(vstup1,false));
            vstup2:=zadani(520,250+posuv,6,prevod(vstup2,false));
               SMAZ(420,190+posuv,615,450,0);
            case cis_bodu of
                    1: s1:='A['+prevod(ZAO(vstup1),false)+','+prevod(ZAO(vstup2),false)+']';
                    2: s1:='B['+prevod(ZAO(vstup1),false)+','+prevod(ZAO(vstup2),false)+']';
                    3: s1:='C['+prevod(ZAO(vstup1),false)+','+prevod(ZAO(vstup2),false)+']';
                    4: s1:='D['+prevod(ZAO(vstup1),false)+','+prevod(ZAO(vstup2),false)+']';
            end;
            if ((cis_bodu mod 2)<>0) AND (cis_rov=1) then disp(420,200+posuv,'PRVNÍ bod PRVNÍ
rovnice',b_mes);
            if ((cis_bodu mod 2)=0)  AND (cis_rov=1) then disp(420,200+posuv,'DRUHÝ bod PRVNÍ
rovnice',b_mes);
            if ((cis_bodu mod 2)<>0) AND (cis_rov=2) then disp(420,200+posuv,'PRVNÍ bod DRUHÉ
rovnice',b_mes);
            if ((cis_bodu mod 2)=0)  AND (cis_rov=2) then disp(420,200+posuv,'DRUHÝ bod DRUHÉ
rovnice',b_mes);
            disp(420,210+posuv,s1,b_ram);
            if ((cis_bodu mod 2)=0) AND (bod[cis_rov*2-1].x=vstup1) AND (bod[cis_rov*2-
1].y=vstup2) then
                    warning(' Tento bod jste již zadali, zkuste jiný ')
            else begin
               if ((fce_ok) AND(((ZAO(vstup2)<>ZAO(odpov1*vstup1+odpov2)) AND (cis_rov=1) OR
               (ZAO(vstup2)<>ZAO(odpov3*vstup1+odpov4)) AND (cis_rov=2))) OR ((NOT fce_ok) AND
               (((ZAO(vstup1)<>ZAO(odpov1*vstup2+odpov2)) AND (cis_rov=1) OR
               (ZAO(vstup1)<>ZAO(odpov3*vstup2+odpov4)) AND (cis_rov=2))) )) then begin
                     if i=1 then      begin
                          warning(' Špatně ... Zkuste to znovu ');
                     end else begin
                       if ((cis_bodu mod 2)<>0) AND (cis_rov=1) then begin
                          if fce_ok then begin
                                 vstup1:=0;
                                 vstup2:=zao(odpov1*vstup1+odpov2);
                          end else begin
                                 vstup1:=odpov2;
                                 vstup2:=0;
                          end;
                       end;
                       if ((cis_bodu mod 2)=0)  AND (cis_rov=1) then begin
                          if fce_ok then begin
                                 vstup1:=1;
                               vstup2:=ZAO(odpov1*vstup1+odpov2);
                          end else begin
                                 vstup1:=odpov2;
                               vstup2:=5;
                          end;
                       end;
                       if ((cis_bodu mod 2)<>0) AND (cis_rov=2) then begin
                          if fce_ok then begin
                              vstup1:=0;
                            vstup2:=ZAO(odpov3*vstup1+odpov4);
                          end else begin
                              vstup1:=odpov4;
                            vstup2:=0;
                          end;
                       end;
                       if ((cis_bodu mod 2)=0)  AND (cis_rov=2) then begin
                          if fce_ok then begin
                                 vstup1:=1;
                               vstup2:=ZAO(odpov3*vstup1+odpov4);
                          end else begin
                                 vstup1:=odpov4;
                               vstup2:=5;
                          end;
                       end;
```

```
                    warning(' Zase špatně ... ');
                  case cis_bodu of
                          1: s1:='A['+prevod(vstup1,false)+','+prevod(vstup2,false)+']';
                          2: s1:='B['+prevod(vstup1,false)+','+prevod(vstup2,false)+']';
                          3: s1:='C['+prevod(vstup1,false)+','+prevod(vstup2,false)+']';
                          4: s1:='D['+prevod(vstup1,false)+','+prevod(vstup2,false)+']';
                  end;
                              disp(420,220+posuv,s1,b_tex);
                  disp(570,220+posuv,'CHYBA',b_err);
                  chyby:=chyby+0.15;
                end;
            end else i:=3;
            inc(i);
        end;
    end;
          bod[cis_bodu].x:=vstup1;
          bod[cis_bodu].y:=vstup2;
          bod[cis_bodu].e:=true;
    if (abs(vstup1*2)+2>pocetx) then begin
                          pocetx:=2*abs(trunc(vstup1))+3;
     GRAF;
     if cis_rov=2 then FCE(1,4);
    end;
     if (abs(vstup2*2)+2>pocety) then begin
            pocety:=2*abs(trunc(vstup2))+3;
     GRAF;
     if cis_rov=2 then FCE(1,4);
    end;
    if ((cis_bodu mod 2)=0)  AND (cis_rov=1) then FCE(1,4);
    if ((cis_bodu mod 2)=0)  AND (cis_rov=2) then FCE(2,2);
    for i:=1 to cis_bodu do if bod[i].e then GRAF_BOD(i,bod[i].x,bod[i].y,b_ram);
    posuv:=posuv+40;
       until((cis_bodu mod 2)=0);
  inc(cis_rov);
  SMAZ(415,200+posuv,615,450,0);
until cis_rov>2;
vstup1:=reseni[cis_prikl,2];
vstup2:=reseni[cis_prikl,3];
if (abs(vstup1*2)+2>pocetx) then begin
              pocetx:=2*abs(trunc(vstup1))+3;
   GRAF;
   FCE(1,4);
   FCE(2,2);
   for i:=1 to cis_bodu do if bod[i].e then GRAF_BOD(i,bod[i].x,bod[i].y,b_ram);
end;
           if (abs(vstup2*2)+2>pocety) then begin
              pocety:=2*abs(trunc(vstup2))+3;
   GRAF;
   FCE(1,4);
   FCE(2,2);
   for i:=1 to cis_bodu do if bod[i].e then GRAF_BOD(i,bod[i].x,bod[i].y,b_ram);
end;
disp(420,380,'Určete počet řešení',b_mes);
p_objekt[1].init(515,395,180,20,'Žádné řešení',1,14,b_men);
     p_objekt[2].init(515,420,180,20,'Jedno řešení',1,14,b_men);
     p_objekt[3].init(515,445,180,20,'Nekonečně řešení',1,14,b_men);
     i:=MENU(3,0,0,639,479);
case i of
     1: i:=0;
     2: i:=1;
     3: i:=9;
end;
if i<>reseni[cis_prikl,1] then begin
    case trunc(reseni[cis_prikl,1]) of
      1: warning(' Soustava má jedno řešení ');
      0: warning(' Soustava nemá žádné řešení ');
      9: warning(' Soustava má nekonečně mnoho řešení ');
    end;
    chyby:=chyby+0.1;
end;
SMAZ(415,360,600,450,0);
if reseni[cis_prikl,1]=1 then begin
                    GRAF_BOD(5,vstup1,vstup2,b_mes);
     disp(12,478,'Řešení soustavy',b_mes);
     disp(420,400,'Odpověd :',b_tex);
     disp(500,400,'P['+prevod(vstup1,false)+','+prevod(vstup2,false)+']',b_ram);
     disp(420,420,'Tato soustava lineárních',b_mes);
                    disp(420,430,'rovnic má přesně jedno  ',b_mes);
     disp(420,440,'řešení pro hodnoty :',b_mes);
     s1:=prevod(reseni[cis_prikl,2],false);
     s2:=prevod(reseni[cis_prikl,3],false);
     disp(420,450,'x = '+s1+'  y = '+s2,b_mes);
end else
  if reseni[cis_prikl,1]=0 then begin
     disp(415,400,'Odpověd :',b_tex);
     disp(415,420,'Tato soustava lineárních',b_mes);
                    disp(415,430,'rovnic nemá žádné řešení',b_mes);
  end else begin
```

```
        FCE(3,b_mes);
        disp(415,400,'Odpověd :',b_tex);
        disp(415,420,'Tato soustava lineárních',b_mes);
                        disp(415,430,'rovnic má nekonečně mnoho',b_mes);
                        disp(415,440,'řešení',b_mes);
        disp(12,478,'Řešení soustavy',b_mes);
      end;
    disp(510,470,'CHYBY : '+prevod(zao(chyby),true),b_ram);
    cekani;
    grafika:=false;
  end;
begin
  START_CASU;
  for cyklus:=1 to pocet_prikl do begin            {`}
      if (NOT konec_odp) then begin
              if (GRAFIKA(posloup[cyklus])) then exit;
              if (NOT (START_OK)) then begin
          VYSLEDKY(cyklus);
          exit;
        end;
              end else begin
          VYSLEDKY(cyklus-1);
          exit;
      end;
  end;
  VYSLEDKY(pocet_prikl);
end;
procedure POCETNI;
var s1,s2,s3,s4,s5,s6:string;
    cyklus:integer;
    znak:char;

      function POC_SCITAC(cis_prikl:integer):boolean;
  var i,cis_rov,cis_prom:integer;
      vstup1,vstup2:real;
      vstup3,vstup4:real;
      vstup5,vstup6:real;
      nas1,nas2:real;
      odpov1,odpov2:real;
      odpov3,odpov4:real;
      odpov5,odpov6:real;
      pom:real;
      fce_ok:boolean;
      function NSN(a1,a2,a3:real):real;
      var pom1,pom2:real;
          citac:longint;
      begin
        citac:=0;
           if (a1*a2)=0 then
              nsn:=0
        else
              if (a1=-1*a2) then
                  nsn:=1
              else begin
                   pom1:=abs(a1);
                   pom2:=abs(a2);
                   repeat begin
               if pom1<pom2 then pom1:=pom1+abs(a1)
               else                                                    pom2:=pom2+abs(a2);
               inc(citac);
             end;
             if citac>2147483646 then begin
                warning(' Chyba ... NSN nebyl nalezen. Zavolejte učitele ! ');
                halt(1);
             end;
           until pom1=pom2;
           nsn:=pom1/a3;
         end;
      end;
      procedure ZOBR_XY(co:byte;x,y:integer);
      begin
        bar3d(x,y+vyska_pisma,x+(9)*sirka_pisma,y-2*vyska_pisma,0,topoff);
           if vstup1<>0 then disp(x+1*sirka_pisma,y,prevod(vstup1,false),b_ram);
        if co=1 then
                          disp(x+10*sirka_pisma,y,'x =',b_ram)
        else
                          disp(x+10*sirka_pisma,y,'y =',b_ram);
        bar3d(x+(14)*sirka_pisma,y+vyska_pisma,x+(23)*sirka_pisma,y-2*vyska_pisma,0,topoff);
           if vstup2<>0 then disp(x+15*sirka_pisma,y,prevod(vstup2,false),b_ram);
      end;
      procedure ZOBR_D(co:byte;x,y:integer);
      begin
        y:=y+2*vyska_pisma;
        bar3d(x,y+vyska_pisma,x+(9)*sirka_pisma,y-2*vyska_pisma,0,topoff);
           if vstup1<>0 then disp(x+1*sirka_pisma,y,prevod(vstup1,false),b_ram);
        disp(x+10*sirka_pisma,y,'x',b_ram);
        bar3d(x+12*sirka_pisma,y+vyska_pisma,x+21*sirka_pisma,y-2*vyska_pisma,0,topoff);
           if vstup2<>0 then disp(x+13*sirka_pisma,y,prevod(vstup2,false),b_ram);
```

```
        disp(x+22*sirka_pisma,y,'y =',b_ram);
        bar3d(x+27*sirka_pisma,y+vyska_pisma,x+36*sirka_pisma,y-2*vyska_pisma,0,topoff);
            if vstup3<>0 then disp(x+28*sirka_pisma,y,prevod(vstup3,false),b_ram);
        if co=1 then begin
           y:=y+30;
           bar3d(x,y+vyska_pisma,x+(9)*sirka_pisma,y-2*vyska_pisma,0,topoff);
              if vstup4<>0 then disp(x+1*sirka_pisma,y,prevod(vstup4,false),b_ram);
           disp(x+10*sirka_pisma,y,'x',b_ram);
           bar3d(x+12*sirka_pisma,y+vyska_pisma,x+21*sirka_pisma,y-2*vyska_pisma,0,topoff);
              if vstup5<>0 then disp(x+13*sirka_pisma,y,prevod(vstup5,false),b_ram);
           disp(x+22*sirka_pisma,y,'y =',b_ram);
           bar3d(x+27*sirka_pisma,y+vyska_pisma,x+36*sirka_pisma,y-2*vyska_pisma,0,topoff);
              if vstup6<>0 then disp(x+28*sirka_pisma,y,prevod(vstup6,false),b_ram);
        end;
      end;
      procedure ZOBR_XYs3(co:byte;x,y:integer);
      var s:string;
      begin
        if co=1 then s:='y ='
        else         s:='x =';
                   disp(x-3*sirka_pisma,y,s,b_ram);
        bar3d(x,y+vyska_pisma,x+9*sirka_pisma,y-2*vyska_pisma,0,topoff);
        if vstup1<>0 then disp(x,y,prevod(vstup1,false),b_ram);
      end;

  begin
    i:=1;
    cis_rov:=1;
    cis_prom:=1;
    vstup1:=0;
    vstup2:=0;
    vstup3:=0;
    vstup4:=0;
    vstup5:=0;
    vstup6:=0;
         RESENI_rov(cis_prikl);
    PRAC_PLOCHA(2);
         ZOBR_ROVNICI(50,120,cis_prikl,1);
         ZOBR_ROVNICI(50,130,cis_prikl,2);
    line(70,135,190+sirka_koef*10,135);
    if (rov[cis_prikl,cis_rov].b=0) OR (rov[cis_prikl,cis_rov].a=0) then fce_OK:=false
    else
                                                                    fce_ok:=true;
    if fce_ok then begin
                p_objekt[1].init(420,150,350,20,'Snažíme se o vyloučení proměnné
X',1,14,b_men);
                p_objekt[2].init(420,180,350,20,'Snažíme se o vyloučení proměnné
Y',1,14,b_men);
                cis_prom:=MENU(2,0,0,639,479);
    end else
              if rov[cis_prikl,cis_rov].b=0 then cis_prom:=1
    else                              cis_prom:=2;
    if cis_prom=0 then begin
       poc_scitac:=true;
       exit;
    end;
       if cis_prom=1 then disp(360,100,'Vyloučíme proměnnou X',b_mes)
       else               disp(360,100,'Vyloučíme proměnnou Y',b_mes);
    while i<3 do begin
          if cis_prom=1 then begin
          if (rov[cis_prikl,1].a+rov[cis_prikl,2].a=0) then begin
               nas1:=1;
               nas2:=1;
          end else
             if (rov[cis_prikl,1].a=rov[cis_prikl,2].a) then begin
               nas1:=1;
               nas2:=-1;
             end else begin
               nas1:=nsn(rov[cis_prikl,1].a,rov[cis_prikl,2].a,rov[cis_prikl,1].a);
               nas2:=(-1)*nsn(rov[cis_prikl,1].a,rov[cis_prikl,2].a,rov[cis_prikl,2].a);
           end
       end else begin
          if (rov[cis_prikl,1].b+rov[cis_prikl,2].b=0) then begin
               nas1:=1;
               nas2:=1;
          end else
             if (rov[cis_prikl,1].b=rov[cis_prikl,2].b) then begin
               nas1:=1;
               nas2:=-1;
             end else begin
               nas1:=nsn(rov[cis_prikl,1].b,rov[cis_prikl,2].b,rov[cis_prikl,1].b);
               nas2:=(-1)*nsn(rov[cis_prikl,1].b,rov[cis_prikl,2].b,rov[cis_prikl,2].b);
           end
       end;
       if ((nas1<-36654) OR (nas1>36654) OR (nas2<-36654) OR (nas2>36654)) then begin
          warning(' Neexistuje společný násobek v našem rozsahu ! ');
          warning(' Zvolte jinou metodu, nebo zavolejte učitele ! ');
```

```
      exit;
      end;
      disp(250,120,'První rovnici vynásobíme  :',b_mes);
            disp(250,140,'Druhou rovnici vynásobíme :',b_mes);
            bar3d(500,120+vyska_pisma,500+9*sirka_pisma,120-2*vyska_pisma,0,topoff);
            vstup1:=ZADANI(500,120-(2*vyska_pisma),7,prevod(vstup1,false));
      SMAZ(500,120+vyska_pisma,500+13*sirka_pisma,120-(2*vyska_pisma),0);
      disp(500,120,prevod(vstup1,true),b_ram);
            bar3d(500,140+vyska_pisma,500+9*sirka_pisma,140-2*vyska_pisma,0,topoff);
            vstup2:=ZADANI(500,140-(2*vyska_pisma),7,prevod(vstup2,false));
      SMAZ(500,140+vyska_pisma,500+13*sirka_pisma,140-(2*vyska_pisma),0);
      disp(500,140,prevod(vstup2,true),b_ram);
      if (vstup1<0) AND (nas1>0) then begin
         nas1:=(-1*nas1);
         nas2:=(-1*nas2);
      end;
      if (vstup1<>nas1) OR (vstup2<>nas2) then begin
         if i=1 then begin
               warning(' Špatně ... Zkuste to znovu ');
               SMAZ(500,140+vyska_pisma,500+13*sirka_pisma,120-(2*vyska_pisma),0);
         end else begin
               warning(' Zase špatně ... ');
               SMAZ(500,140+vyska_pisma,500+13*sirka_pisma,120-(2*vyska_pisma),0);
               disp(490,110,'Opraveno !',b_err);
               disp(500,120,prevod(nas1,true),b_ram);
               disp(500,140,prevod(nas2,true),b_ram);
            chyby:=chyby+0.2;
            cekani;
         end;
      end else i:=3;
      inc(i);
   end;
   vstup1:=0;
   vstup2:=0;
   odpov1:=zao(rov[cis_prikl,1].a*nas1);
   odpov2:=zao(rov[cis_prikl,1].b*nas1);
   odpov3:=zao(rov[cis_prikl,1].c*nas1);
         odpov4:=zao(rov[cis_prikl,2].a*nas2);
         odpov5:=zao(rov[cis_prikl,2].b*nas2);
               odpov6:=zao(rov[cis_prikl,2].c*nas2);
   i:=1;
   disp(250,170,'Vynásobíme obě rovnice a dostaneme :',b_mes);
   while i<3 do begin
       SMAZ(50,190+40+2*vyska_pisma,50+50*sirka_pisma,190-(2*vyska_pisma),0);
       ZOBR_D(1,50,190);
       vstup1:=zadani(50,190,7,prevod(vstup1,false));
       vstup2:=zadani(50+12*sirka_pisma,190,7,prevod(vstup2,false));
       vstup3:=zadani(50+27*sirka_pisma,190,7,prevod(vstup3,false));
       vstup4:=zadani(50,190+30,7,prevod(vstup4,false));
       vstup5:=zadani(50+12*sirka_pisma,190+30,7,prevod(vstup5,false));
       vstup6:=zadani(50+27*sirka_pisma,190+30,7,prevod(vstup6,false));
       if ((vstup1<>odpov1) OR (vstup2<>odpov2) OR (vstup3<>odpov3)
          OR (vstup4<>odpov4) OR (vstup5<>odpov5) OR (vstup6<>odpov6)) then begin
          if i<>1 then        begin
               warning(' Zase špatně ... ');
            if cis_prom=1 then s1:='x = '+prevod(odpov1,false)+'y '+prevod(odpov2,true)
            else               s1:='y = '+prevod(odpov1,false)+'x '+prevod(odpov2,true);
            SMAZ(50,190+40+2*vyska_pisma,50+50*sirka_pisma,190-(2*vyska_pisma),0);
            disp(430,190,'Chyba po vynásobení !',b_err);
            vstup1:=ZAO(rov[cis_prikl,1].a*nas1);
            vstup2:=ZAO(rov[cis_prikl,1].b*nas1);
            vstup3:=ZAO(rov[cis_prikl,1].c*nas1);
                  vstup4:=ZAO(rov[cis_prikl,2].a*nas2);
                  vstup5:=ZAO(rov[cis_prikl,2].b*nas2);
                        vstup6:=ZAO(rov[cis_prikl,2].c*nas2);
            chyby:=chyby+0.2;
               end;
       end else i:=3;   {Spravne}
       s1:=prevod(vstup1,false);
       while length(s1)<sirka_koef+1 do s1:=' '+s1;
       s2:=prevod(abs(vstup2),false);
       while length(s2)<sirka_koef+1 do s2:=' '+s2;
       if vstup2<0 then s2:='-'+s2
       else             s2:='+'+s2;
       s3:=prevod(vstup3,false);
       while length(s3)<sirka_koef+1 do s3:=' '+s3;
       s4:=prevod(vstup4,false);
       while length(s4)<sirka_koef+1 do s4:=' '+s4;
       s5:=prevod(abs(vstup5),false);
       while length(s5)<sirka_koef+1 do s5:=' '+s5;
       if vstup5<0 then s5:='-'+s5
       else             s5:='+'+s5;
       s6:=prevod(vstup6,false);
       while length(s6)<sirka_koef+1 do s6:=' '+s6;
       s1:=s1+'x '+s2+'y = '+s3;
       s2:=s4+'x '+s5+'y = '+s6;
       SMAZ(50,190+40+2*vyska_pisma,50+46*sirka_pisma,190-(2*vyska_pisma),0);
       disp(50,190,s1,b_ram);
```

```
        disp(50,200,s2,b_ram);
        if ((vstup1<>odpov1) OR (vstup2<>odpov2) OR (vstup3<>odpov3)
           OR (vstup4<>odpov4) OR (vstup5<>odpov5) OR (vstup6<>odpov6))
           AND (i=1) then warning(' Špatně ... Zkuste to znovu ');
        inc(i);
    end;
    line(70,205,190+sirka_koef*10,205);
    i:=1;
    disp(250,210,'Po sečtení (resp. odečtení) rovnic dostaneme',b_mes);
    odpov1:=vstup1+vstup4;
    odpov2:=vstup2+vstup5;
    odpov3:=vstup3+vstup6;
    vstup1:=0;
    vstup2:=0;
    vstup3:=0;
    while i<3 do begin
        SMAZ(50,220+3*vyska_pisma,50+46*sirka_pisma,220-vyska_pisma,0);
        ZOBR_D(2,50,220);
        vstup1:=zadani(50,220,7,prevod(vstup1,false));
        vstup2:=zadani(50+12*sirka_pisma,220,7,prevod(vstup2,false));
        vstup3:=zadani(50+27*sirka_pisma,220,7,prevod(vstup3,false));
        if ((vstup1<>odpov1) OR (vstup2<>odpov2) OR (vstup3<>odpov3)) then begin
           if i<>1 then       begin
                                   warning(' Zase špatně ... ');
             if cis_prom=1 then s1:='x = '+prevod(odpov1,false)+'y '+prevod(odpov2,true)
             else               s1:='y = '+prevod(odpov1,false)+'x '+prevod(odpov2,true);
             SMAZ(50,220+3*vyska_pisma,50+46*sirka_pisma,220-vyska_pisma,0);
             disp(430,230,'Chyba součtu-rozdílu !',b_err);
             vstup1:=odpov1;
             vstup2:=odpov2;
             vstup3:=odpov3;
             chyby:=chyby+0.2;
                end;
        end else i:=3;   {Spravne}
        s1:=prevod(vstup1,false);
        while length(s1)<sirka_koef+1 do s1:=' '+s1;
        s2:=prevod(abs(vstup2),false);
        while length(s2)<sirka_koef+1 do s2:=' '+s2;
        if vstup2<0 then s2:='-'+s2
        else             s2:='+'+s2;
        s3:=prevod(vstup3,false);
        while length(s3)<sirka_koef+1 do s3:=' '+s3;
        s4:=s1+'x '+s2+'y = '+s3;
        SMAZ(50,220+3*vyska_pisma,50+46*sirka_pisma,220-vyska_pisma,0);
        disp(50,230,s4,b_ram);
        if ((vstup1<>odpov1) OR (vstup2<>odpov2) OR (vstup3<>odpov3))
           AND (i=1) then warning(' Špatně ... Zkuste to znovu ');
        inc(i);
    end;
    i:=1;
    disp(250,250,'Vyřešime tuto rovnici',b_mes);
    if cis_prom=1 then begin
                              s5:=prevod(vstup2,false)+'y = '+s3;
        if vstup2<>0 then odpov1:=vstup3/vstup2;
    end else begin
        s5:=prevod(vstup1,false)+'x = '+s3;
        if vstup1<>0 then odpov1:=vstup3/vstup1;
    end;
    vstup1:=0;
    disp(50+(21-pos('=',s5)-5)*sirka_pisma,260,s5,b_ram);
    if ((reseni[cis_prikl,1]=9) OR (reseni[cis_prikl,1]=0)) then begin
            p_objekt[1].init(350,300,350,20,'Soustava nemá žádné řešení',1,14,b_men);
            p_objekt[2].init(350,325,350,20,'Soustava má pouze jedno řešení',1,14,b_men);
            p_objekt[3].init(350,350,350,20,'Soustava má nekonečně mnoho řešení',1,14,b_men);
            i:=MENU(3,0,0,639,479);
       case i of
            1: i:=0;
            2: i:=1;
            3: i:=9;
       end;
       if (reseni[cis_prikl,1]=9) then s2:='  CHYBA - Soustava má nekonečně řešení !  '
       else
                        s2:='  CHYBA - Soustava nemá žádné řešení !  ';
       if (reseni[cis_prikl,1]<>i) then begin
                          warning(s2);
          chyby:=chyby+0.4;
       end;
       disp(40,360,'Odpověd :',b_tex);
       if (reseni[cis_prikl,1]=9) then
              disp(40,380,'Soustava lineárních rovnic má tedy nekonečně mnoho řešení',b_mes)
       else
              disp(40,380,'Soustava lineárních rovnic nemá tedy žádné řešení',b_mes);
       disp(350,470,'DET x:'+prevod(reseni[cis_prikl,2],false)+'
y:'+prevod(reseni[cis_prikl,3],false),b_ram);
       disp(520,470,'CHYBY : '+prevod(zao(chyby),true),b_ram);
       cekani;
       poc_scitac:=false;
       exit;
```

```
    end else begin
       s4:='x = 0';
       while i<3 do begin
          ZOBR_XYs3(cis_prom,50+(19-pos('=',s4))*sirka_pisma,280);
          vstup1:=zadani(50+(19-pos('=',s4))*sirka_pisma,280-
2*vyska_pisma,7,prevod(vstup1,false));
          if (vstup1<>odpov1) then begin
                if (i=1) then warning(' Špatně ... Zkuste to znovu ')
             else begin
                  warning(' Zase špatně ... ');
                disp(430,280,'Chyba výpočtu !',b_err);
                vstup1:=odpov1;
                chyby:=chyby+0.2;
             end;
          end else i:=4;
          SMAZ(50,280+2*vyska_pisma,50+46*sirka_pisma,280-2*vyska_pisma,0);
                           inc(i);
       end;
       s3:=prevod(vstup1,false);
       while length(s3)<sirka_koef+1 do s3:=' '+s3;
                if cis_prom=1 then s6:='y = '+s3
       else                                  s6:='x = '+s3;
       disp(50+(16-pos('=',s4))*sirka_pisma,280,s6,b_ram);
       line(50+(16-pos('=',s4))*sirka_pisma,282,50+(16-
pos('=',s4)+length(s6))*sirka_pisma,282);
       line(50+(16-pos('=',s4))*sirka_pisma,284,50+(16-
pos('=',s4)+length(s6))*sirka_pisma,284);
       i:=1;
            p_objekt[1].init(420,350,350,20,'Proměnnou dosadíme do první rovnice',1,14,b_men);
            p_objekt[2].init(420,380,350,20,'Proměnnou dosadíme do druhé rovnice',1,14,b_men);
            cis_rov:=MENU(2,0,0,639,479);
       if cis_rov=0 then begin
                        poc_scitac:=true;
         exit;
       end;
       if cis_rov=1 then disp(250,300,'Proměnnou dosadíme například do první rovnice',b_mes)
       else              disp(250,300,'Proměnnou dosadíme například do druhé rovnice',b_mes);
       if cis_prom=1 then begin
          s1:=prevod(rov[cis_prikl,cis_rov].a,false)+'x =';
          s1:=s1+' '+prevod((-1)*rov[cis_prikl,cis_rov].b,false);
          s1:=s1+'y ';
          s1:=s1+prevod(rov[cis_prikl,cis_rov].c,true);
          s2:=prevod(rov[cis_prikl,cis_rov].a,false)+'x =';
          s2:=s2+' '+prevod((-1)*rov[cis_prikl,cis_rov].b,false);
          s2:=s2+'*( '+prevod(vstup1,false)+' )';
                           s2:=s2+prevod(rov[cis_prikl,cis_rov].c,true);
          odpov1:=rov[cis_prikl,cis_rov].a;
          odpov2:=(-1)*rov[cis_prikl,cis_rov].b*vstup1+rov[cis_prikl,cis_rov].c;
       end else begin
          s1:=prevod(rov[cis_prikl,cis_rov].b,false)+'y =';
          s1:=s1+' '+prevod((-1)*rov[cis_prikl,cis_rov].a,false);
          s1:=s1+'x ';
          s1:=s1+prevod(rov[cis_prikl,cis_rov].c,true);
          s2:=prevod(rov[cis_prikl,cis_rov].b,false)+'y =';
          s2:=s2+' '+prevod((-1)*rov[cis_prikl,cis_rov].a,false);
          s2:=s2+'*( '+prevod(vstup1,false)+' )';
                           s2:=s2+prevod(rov[cis_prikl,cis_rov].c,true);
          odpov1:=rov[cis_prikl,cis_rov].b;
          odpov2:=(-1)*rov[cis_prikl,cis_rov].a*vstup1+rov[cis_prikl,cis_rov].c;
       end;
       disp(50+(16-pos('=',s1))*sirka_pisma,310,s1,b_ram);
       disp(50+(16-pos('=',s2))*sirka_pisma,330,s2,b_ram);
       disp(250,350,'Po úpravě dostaneme',b_mes);
       vstup1:=0;
       vstup2:=0;
       i:=1;
       while i<3 do begin
           SMAZ(50,370+2*vyska_pisma,50+46*sirka_pisma,370-2*vyska_pisma,0);
                      ZOBR_XY(cis_prom,50+3*sirka_pisma,370);
           vstup1:=ZADANI(50+3*sirka_pisma,370-8-vyska_pisma,7,prevod(vstup1,false));
           vstup2:=ZADANI(50+17*sirka_pisma,370-8-vyska_pisma,7,prevod(vstup2,false));
           if cis_prom=1 then s1:=prevod(vstup1,false)+'x = '+prevod(vstup2,false)
           else               s1:=prevod(vstup1,false)+'y = '+prevod(vstup2,false);
           SMAZ(50,370+2*vyska_pisma,50+46*sirka_pisma,370-2*vyska_pisma,0);
           disp(50+(16-pos('=',s1))*sirka_pisma,370,s1,b_ram);
           if (vstup1<>odpov1) OR (vstup2<>odpov2) then begin
                  if (i=1) then warning(' Špatně ... Zkuste to znovu ')
              else begin
                    warning(' Zase špatně ... ');
                 SMAZ(50,370+2*vyska_pisma,50+46*sirka_pisma,370-2*vyska_pisma,0);
                 disp(430,370,'Chyba úpravy !',b_err);
                 vstup1:=odpov1;
                 vstup2:=odpov2;
                 if cis_prom=1 then s1:=prevod(vstup1,false)+'x = '+prevod(vstup2,false)
                 else               s1:=prevod(vstup1,false)+'y = '+prevod(vstup2,false);
                 chyby:=chyby+0.2;
              end;
           end else i:=4;
```

```
          inc(i);
        end;
        disp(50+(16-pos('=',s1))*sirka_pisma,370,s1,b_ram);
        disp(250,380,'Z toho plyne',b_mes);
        vstup1:=0;
        s1:='x = 0';
        i:=1;
        odpov1:=odpov2/odpov1;
        while i<3 do begin
          SMAZ(50,400+2*vyska_pisma,50+46*sirka_pisma,400-2*vyska_pisma,0);
           if cis_prom=1 then ZOBR_XYs3(2,50+(19-pos('=',s1))*sirka_pisma,400)
          else                ZOBR_XYs3(1,50+(19-pos('=',s1))*sirka_pisma,400);
          vstup1:=zadani(50+16*sirka_pisma,400-8-vyska_pisma,7,prevod(vstup1,false));
          if cis_prom=1 then s1:='x = '+prevod(vstup1,false)
          else               s1:='y = '+prevod(vstup1,false);
          SMAZ(50,400+2*vyska_pisma,50+46*sirka_pisma,400-2*vyska_pisma,0);
          disp(50+(16-pos('=',s1))*sirka_pisma,400,s1,b_ram);
          if (vstup1<>odpov1) then begin
            if (i=1) then warning(' Špatně ... Zkuste to znovu ')
             else begin
                  warning(' Zase špatně ... ');
                SMAZ(50,400+2*vyska_pisma,50+46*sirka_pisma,400-2*vyska_pisma,0);
                disp(430,410,'Chyba výpočtu !',b_err);
                vstup1:=odpov1;
                if cis_prom=1 then s1:='x = '+prevod(vstup1,false)
                else               s1:='y = '+prevod(vstup1,false);
                chyby:=chyby+0.2;
             end;
          end else i:=4;
          inc(i);
        end;
        disp(50+(16-pos('=',s1))*sirka_pisma,400,s1,b_ram);
        line(50+(16-pos('=',s1))*sirka_pisma,402,50+(16-
pos('=',s1)+length(s1))*sirka_pisma,402);
        line(50+(16-pos('=',s1))*sirka_pisma,404,50+(16-
pos('=',s1)+length(s1))*sirka_pisma,404);
        disp(40,420,'Odpověd :',b_tex);
        disp(40,430,'Soustava lineárních rovnic má tedy jedno řešení',b_mes);
        disp(40,440,'právě pro hodnoty :',b_mes);
        while pos(' ',s3)<>0 do delete(s3,pos(' ',s3),1);
        if cis_prom=1then disp(40,450,s1+'  y = '+s3,b_mes)
        else               disp(40,450,s1+'  x = '+s3,b_mes);
                disp(350,470,'DET x:'+prevod(reseni[cis_prikl,2],false)+'
                y:'+prevod(reseni[cis_prikl,3],false),b_ram);
        disp(520,470,'CHYBY : '+prevod(zao(chyby),true),b_ram);
        poc_scitac:=false;
        cekani;
      end;
    end;
        function POC_DOSAZ(cis_prikl:integer):boolean;
    var i,cis_rov,cis_prom:integer;
        vstup1,vstup2:real;
        odpov1,odpov2:real;
        odpov3,odpov4:real;
        odpov5,odpov6:real;
        pom:real;
        fce_ok:boolean;
        procedure ZOBR_XY3(co:byte;x,y:integer);
        var s:string;
        begin
          if co=1 then s:='x ='
          else         s:='y =';
                    disp(x+(pos('=',s3)-3)*sirka_pisma,y,s,b_ram);
          bar3d(x+pos('=',s3)*sirka_pisma,y+vyska_pisma,x+(pos('=',s3)+9)*sirka_pisma,y-
2*vyska_pisma,0,topoff);
          if vstup1<>0 then disp(x+(pos('=',s3)+2)*sirka_pisma,y,prevod(vstup1,false),b_ram);
        end;
        procedure ZOBR_XY(co:byte;x,y:integer);
        begin
          if co=1 then
                              disp(x,y,'x =',b_ram)
          else
                              disp(x,y,'y =',b_ram);
          bar3d(x+4*sirka_pisma,y+vyska_pisma,x+(13)*sirka_pisma,y-2*vyska_pisma,0,topon);
             if vstup1<>0 then disp(x+5*sirka_pisma,y,prevod(vstup1,false),b_ram);
          if co=1 then
                  disp(x+14*sirka_pisma,y,'y ',b_ram)
          else
                  disp(x+14*sirka_pisma,y,'x ',b_ram);
          bar3d(x+(16)*sirka_pisma,y+vyska_pisma,x+(25)*sirka_pisma,y-2*vyska_pisma,0,topoff);
             if vstup2<>0 then disp(x+17*sirka_pisma,y,prevod(vstup2,false),b_ram);
        end;
        procedure ZOBR_XY2(pr,rovn,co:byte;x,y:integer);
        begin
          s3:='';
          if co=1 then begin
                              s3:=prevod(rov[pr,rovn].a,false);
```

```
                        s3:=s3+'*('+copy(s1,5,length(s1))+')
'+prevod(rov[pr,rovn].b,true)+'y = '+prevod(rov[pr,rovn].c,false);
        end else begin
                        s3:=prevod(rov[pr,rovn].a,false)+'x'+prevod(rov[pr,rovn].b,true);
                        s3:=s3+'*('+copy(s1,5,length(s1))+') =
'+prevod(rov[pr,rovn].c,false);
        end;
                disp(x,y,s3,b_ram);
        y:=y+20;
        bar3d(x+(pos('=',s3)-13)*sirka_pisma,y+vyska_pisma,x+(pos('=',s3)-4)*sirka_pisma,y-
2*vyska_pisma,0,topoff);
        if vstup1<>0 then disp(x+(pos('=',s3)-12)*sirka_pisma,y,prevod(vstup1,false),b_ram);
        if co=1 then
                disp(x+(pos('=',s3)-3)*sirka_pisma,y,'y = ',b_ram)
        else
                disp(x+(pos('=',s3)-3)*sirka_pisma,y,'x = ',b_ram);
        bar3d(x+(pos('=',s3))*sirka_pisma,y+vyska_pisma,x+(pos('=',s3)+9)*sirka_pisma,y-
2*vyska_pisma,0,topoff);
        if vstup2<>0 then disp(x+(pos('=',s3)+1)*sirka_pisma,y,prevod(vstup2,false),b_ram);
      end;
  begin
    i:=1;
    cis_rov:=1;
    cis_prom:=1;
    vstup1:=0;
    vstup2:=0;
         RESENI_ROV(cis_prikl);
    PRAC_PLOCHA(1);
         ZOBR_ROVNICI(50,120,cis_prikl,1);
         ZOBR_ROVNICI(50,130,cis_prikl,2);
    line(70,135,190+sirka_koef*10,135);
         p_objekt[1].init(420,150,350,20,'Proměnnou vyjádříme z PRVNÍ rovnice',1,14,b_men);
         p_objekt[2].init(420,180,350,20,'Proměnnou vyjádříme z DRUHÉ rovnice',1,14,b_men);
         cis_rov:=MENU(2,0,0,639,479);
    if cis_rov=0 then begin
       poc_dosaz:=true;
      exit;
    end;
    if (rov[cis_prikl,cis_rov].a=0) OR (rov[cis_prikl,cis_rov].b=0) then fce_OK:=false
    else        fce_ok:=true;
    if fce_ok then begin
                p_objekt[1].init(420,150,350,20,'Vyjádříme proměnnou X',1,14,b_men);
                p_objekt[2].init(420,180,350,20,'Vyjádříme proměnnou Y',1,14,b_men);
                cis_prom:=MENU(2,0,0,639,479);
    end else
    if (rov[cis_prikl,cis_rov].b=0) then cis_prom:=1
    else                                 cis_prom:=2;
              if cis_prom=0 then begin
                     poc_dosaz:=true;
                    exit;
    end;
    if cis_rov=1 then
              if cis_prom=1 then disp(270,120,'Z první rovnice vyjadříme x',b_mes)
           else                disp(270,120,'Z první rovnice vyjadříme y',b_mes)
    else
              if cis_prom=1 then disp(270,130,'Z druhé rovnice vyjadříme x',b_mes)
           else                disp(270,130,'Z druhé rovnice vyjadříme y',b_mes);
    while i<3 do begin
       ZOBR_XY(cis_prom,50,160);
            vstup1:=ZADANI(50+4*sirka_pisma,160-(2*vyska_pisma),7,prevod(vstup1,false));
       vstup1:=ZAO(vstup1);
            vstup2:=ZADANI(50+16*sirka_pisma,160-(2*vyska_pisma),7,prevod(vstup2,false));
       vstup2:=ZAO(vstup2);
       s1:=prevod(vstup1,false);
       s2:=prevod(vstup2,true);
       if cis_prom=1 then begin
              odpov1:=ZAO(-1*rov[cis_prikl,cis_rov].b/rov[cis_prikl,cis_rov].a);
              odpov2:=ZAO(rov[cis_prikl,cis_rov].c/rov[cis_prikl,cis_rov].a);
       end
       else begin
              odpov1:=ZAO(-1*rov[cis_prikl,cis_rov].a/rov[cis_prikl,cis_rov].b);
              odpov2:=ZAO(rov[cis_prikl,cis_rov].c/rov[cis_prikl,cis_rov].b);
       end;
       if cis_prom=1 then s1:='x = '+s1+'y '
       else               s1:='y = '+s1+'x ';
       if vstup2>0 then   s1:=s1+s2
       else               s1:=s1+s2;
       SMAZ(25,160+vyska_pisma,50+35*sirka_pisma,160-(2*vyska_pisma),0);
              disp(50,160,s1,b_ram);
       if ((vstup1<>odpov1) AND (cis_prom=1) OR (vstup1<>odpov1) AND (cis_prom=2)
       OR (vstup2<>odpov2) AND (cis_prom=1)  OR (vstup2<>odpov2) AND (cis_prom=2)) then begin
          if i=1 then begin
              warning(' Špatně ... Zkuste to znovu ');
              SMAZ(25,160+vyska_pisma,50+35*sirka_pisma,160-(2*vyska_pisma),0);
          end else begin
              warning(' Zase špatně ... ');
            if cis_prom=1 then s1:='x = '+prevod(odpov1,false)+'y '+prevod(odpov2,true)
            else               s1:='y = '+prevod(odpov1,false)+'x '+prevod(odpov2,true);
```

```
            disp(400,150,'Správné vyjádření je :',b_err);
               disp(400,160,s1,b_tex);
            chyby:=chyby+0.25;
            cekani;
         end;
      end else i:=3;
      inc(i);
   end;
   i:=1;
   SMAZ(25,160+vyska_pisma,50+25*sirka_pisma,160-(2*vyska_pisma),0);
   disp(50,160,s1,b_ram);
   disp(270,180,'Vyjádření dosadíme do zbývající rovnice',b_mes);
   disp(270,195,'a zjednodušíme zápis',b_mes);
   vstup1:=0;
   vstup2:=0;
            while i<3 do begin
      if cis_rov=1 then ZOBR_XY2(cis_prikl,2,cis_prom,50,210)
      else              ZOBR_XY2(cis_prikl,1,cis_prom,50,210);
      vstup1:=ZADANI(50+(pos('=',s3)-13)*sirka_pisma,230-
(2*vyska_pisma),7,prevod(vstup1,false));
      vstup1:=ZAO(vstup1);
      vstup2:=ZADANI(50+(pos('=',s3))*sirka_pisma,230-(2*vyska_pisma),7,prevod(vstup2,false));
      vstup2:=ZAO(vstup2);
      if cis_prom=1 then s2:=prevod(vstup1,false)+'y = '+prevod(vstup2,false)
      else               s2:=prevod(vstup1,false)+'x = '+prevod(vstup2,false);
      if cis_rov=1 then begin
         if cis_prom=1 then begin
                     odpov3:=ZAO(rov[cis_prikl,2].a*odpov1+rov[cis_prikl,2].b);
                     odpov4:=ZAO(rov[cis_prikl,2].c-rov[cis_prikl,2].a*odpov2);
               end
            else begin
                     odpov3:=ZAO(rov[cis_prikl,2].a+rov[cis_prikl,2].b*odpov1);
                     odpov4:=ZAO(rov[cis_prikl,2].c-rov[cis_prikl,2].b*odpov2);
               end;
      end else begin
         if cis_prom=1 then begin
                     odpov3:=ZAO(rov[cis_prikl,1].a*odpov1+rov[cis_prikl,1].b);
                     odpov4:=ZAO(rov[cis_prikl,1].c-rov[cis_prikl,1].a*odpov2);
                  end
                  else begin
                     odpov3:=ZAO(rov[cis_prikl,1].a+rov[cis_prikl,1].b*odpov1);
                     odpov4:=ZAO(rov[cis_prikl,1].c-rov[cis_prikl,1].b*odpov2);
               end;
      end;
if (((vstup1<>odpov3) OR (vstup2<>odpov4)) AND ((vstup1<>(-1)*odpov3) OR (vstup2<>(-
1)*odpov4))) then begin
if i=1 then    begin
   warning(' Špatně ... Zkuste to znovu ');
   SMAZ(25,230+vyska_pisma,50+40*sirka_pisma,230-(2*vyska_pisma),0);
  end else begin
         warning(' Zase špatně ... ');
            if cis_prom=1 then s2:=prevod(odpov3,false)+'y = '+prevod(odpov4,false)
            else               s2:=prevod(odpov3,false)+'x = '+prevod(odpov4,false);
            disp(400,210,'Správné vyjádření je :',b_err);
                        disp(400,220,s2,b_tex);
            chyby:=chyby+0.25;
            cekani;
         end;
      end else i:=3;
      inc(i);
   end;
   SMAZ(25,230+vyska_pisma,50+40*sirka_pisma,230-(2*vyska_pisma),0);
   disp(50+(pos('=',s3)-pos('=',s2))*sirka_pisma,230,s2,b_ram);
   i:=1;
   if ((reseni[cis_prikl,1]=9) OR (reseni[cis_prikl,1]=0)) then begin
         p_objekt[1].init(350,300,350,20,'Soustava nemá žádné řešení',1,14,b_men);
         p_objekt[2].init(350,325,350,20,'Soustava má pouze jedno řešení',1,14,b_men);
         p_objekt[3].init(350,350,350,20,'Soustava má nekonečně mnoho řešení',1,14,b_men);
         i:=MENU(3,0,0,639,479);
      case i of
           1: i:=0;
           2: i:=1;
           3: i:=9;
      end;
      if (reseni[cis_prikl,1]=9) then s2:='  CHYBA - Soustava má nekonečně řešení !  '
      else
                     s2:='  CHYBA - Soustava nemá žádné řešení !  ';
      if (reseni[cis_prikl,1]<>i) then begin
                        warning(s2);
         SMAZ(25,250+vyska_pisma,50+55*sirka_pisma,250-(2*vyska_pisma),0);
         chyby:=chyby+0.5;
      end;
      disp(40,360,'Odpověd :',b_tex);
      if (reseni[cis_prikl,1]=9) then
            disp(40,380,'Soustava lineárních rovnic má tedy nekonečně mnoho řešení',b_mes)
      else
      disp(40,380,'Soustava lineárních rovnic nemá tedy žádné řešení',b_mes);
```

```
            disp(350,470,'DET x:'+prevod(reseni[cis_prikl,2],false)+'
     y:'+prevod(reseni[cis_prikl,3],false),b_ram);
     disp(520,470,'CHYBY : '+prevod(zao(chyby),true),b_ram);
     cekani;
     poc_dosaz:=false;
     exit;
  end else begin
    if cis_prom=1 then ZOBR_XY3(2,50,250)
    else               ZOBR_XY3(1,50,250);
    vstup1:=0;
    while i<3 do begin
       vstup1:=ZADANI(50+pos('=',s3)*sirka_pisma,250-(2*vyska_pisma),7,prevod(vstup1,false));
       vstup1:=ZAO(vstup1);
       odpov5:=ZAO(odpov4/odpov3);
       if cis_prom=1 then s4:='y = '+prevod(vstup1,false)
       else               s4:='x = '+prevod(vstup1,false);
       if odpov5<>vstup1 then
             if i=1 then    begin
                  warning(' Špatně ... Zkuste to znovu ');
               SMAZ(25,250+vyska_pisma,50+55*sirka_pisma,250-(2*vyska_pisma),0);
             end else begin
                  warning(' Zase špatně ... ');
               if cis_prom=1 then s4:='y = '+prevod(odpov5,false)
               else               s4:='x = '+prevod(odpov5,false);
               disp(400,240,'Správné vyjádření je :',b_err);
               disp(400,250,s4,b_tex);
               chyby:=chyby+0.25;
               cekani;
             end
       else i:=3;
       SMAZ(25,250+vyska_pisma,50+40*sirka_pisma,250-(2*vyska_pisma),0);
       disp(50+(pos('=',s3)-3)*sirka_pisma,250,s4,b_ram);
       inc(i);
    end;
  end;
  setcolor(b_ram);
  line(50+(pos('=',s3)-3)*sirka_pisma,251,50+((pos('=',s3)-3)+length(s4))*sirka_pisma,251);
  line(50+(pos('=',s3)-3)*sirka_pisma,253,50+((pos('=',s3)-3)+length(s4))*sirka_pisma,253);
  i:=1;
  s3:=s1;
  if cis_prom=1 then begin
             insert('*('+copy(s4,4,length(s4))+')',s1,pos('y',s1));
             delete(s1,pos('y',s1),1);
  end else begin
             insert('*('+copy(s4,4,length(s4))+')',s1,pos('x',s1));
             delete(s1,pos('x',s1),1);
  end;
  disp(50,290,s1,b_ram);
  disp(270,280,'Dosadíme do naší první vyjádřené rovnice',b_mes);
  vstup1:=0;
  while i<3 do begin
     if cis_prom=1 then ZOBR_XY3(1,50,310)
     else               ZOBR_XY3(2,50,310);
     vstup1:=ZADANI(50+pos('=',s3)*sirka_pisma,310-(2*vyska_pisma),7,prevod(vstup1,false));
     vstup1:=ZAO(vstup1);
     odpov6:=ZAO(odpov1*odpov5+odpov2);
     if cis_prom=1 then s1:='x = '+prevod(vstup1,false)
     else               s1:='y = '+prevod(vstup1,false);
     if odpov6<>vstup1 then
           if i=1 then      begin
                        warning(' Špatně ... Zkuste to znovu ');
                     SMAZ(25,310+vyska_pisma,50+25*sirka_pisma,310-(2*vyska_pisma),0);
           end else begin
                        warning(' Zase špatně ... ');
             if cis_prom=1 then s1:='x = '+prevod(odpov6,false)
             else               s1:='y = '+prevod(odpov6,false);
             disp(400,300,'Správné vyjádření je :',b_err);
                        disp(400,310,s1,b_tex);
             chyby:=chyby+0.25;
            cekani;
            end
     else i:=3;
    SMAZ(25,310+vyska_pisma,50+25*sirka_pisma,310-(2*vyska_pisma),0);
    disp(50+(pos('=',s3)-3)*sirka_pisma,310,s1,b_ram);
    inc(i);
  end;
  setcolor(b_ram);
  line(50+(pos('=',s3)-3)*sirka_pisma,311,50+((pos('=',s3)-3)+length(s1))*sirka_pisma,311);
  line(50+(pos('=',s3)-3)*sirka_pisma,313,50+((pos('=',s3)-3)+length(s1))*sirka_pisma,313);
  disp(40,360,'Odpověd :',b_tex);
  disp(40,380,'Soustava lineárních rovnic má tedy jedno řešení',b_mes);
  disp(40,390,'právě pro hodnoty :',b_mes);
  disp(40,410,s1+'  '+s4,b_mes);
  disp(350,470,'DET x:'+prevod(reseni[cis_prikl,2],false)+'
y:'+prevod(reseni[cis_prikl,3],false),b_ram);
  disp(520,470,'CHYBY : '+prevod(zao(chyby),true),b_ram);
  cekani;
  poc_dosaz:=false;
```

```
  end;
begin
  p_objekt[1].init(x_max div 2,y_max div 2-30,250,20,'Metoda dosazovací',1,14,b_men);
  p_objekt[2].init(x_max div 2,y_max div 2,250,20,'Metoda sčítací',1,14,b_men);
  p_objekt[3].init(x_max div 2,y_max div 2+30,250,20,'Návrat',1,14,b_men);
  START_CASU;
  case MENU(3,195,200,445,280) of
    1: for cyklus:=1 to pocet_prikl do begin
         if (NOT konec_odp) then begin
               if POC_DOSAZ(posloup[cyklus]) then exit;     {`}
                     if (NOT (START_OK)) then begin
                        VYSLEDKY(cyklus);
                        exit;
                     end;
               end else begin
           VYSLEDKY(cyklus-1);
           exit;
       end;
       end;
    2: for cyklus:=1 to pocet_prikl do begin
         if (NOT konec_odp) then begin
               if POC_SCITAC(posloup[cyklus]) then exit;     {`posloup[cyklus]}
                                 if (NOT (START_OK)) then begin
           VYSLEDKY(cyklus);
           exit;
         end;
               end else begin
           VYSLEDKY(cyklus-1);
           exit;
       end;
       end;
  else begin
              konec_odp:=true;
                                     exit;
       end;
  end;
  VYSLEDKY(pocet_prikl);
end;


{*****************************************************************************
******************** T E L O   P R O G R A M U ****************************
*****************************************************************************}
begin
       GR_INIT;
       if ParamCount<2 then begin
              warning(' Na příkazové řádce chybí parametry ');
    exit;
  end;
  val(ParamStr(1),pocet_prikl,code);
  val(ParamStr(2),delka_odpov,code);
  PR_INIT;
  HLAVICKA;
  CLEARSCREEN(1);
  if (NOT NACTENI_ROVNIC) then warning('  Chyba při načítání dat ze souboru   ')
  else begin
     konec:=FALSE;
     repeat
       VYBER;
       chyby:=0;
       case HL_MENU of
              1: POCETNI;
              2: GRAFICKY;
       else konec:=true;
             end;
       CLEARSCREEN(1);
     until konec;
  end;
  closegraph;
end.
```

$\frac{m-n}{3} = 3m + 6n - 1 \quad |\cdot 3$

$m - n = 9m + 18n - 3 \quad |-9m$

$8m - n = 18n - 3 \quad |-18n$

$m - 19n = -3$

$\frac{m-n}{3} = 3m + 6n - 1 \quad |\cdot 3$

$m - n = 9m + 6$

$\frac{m-n}{3} = 3m + 6n - 1 \quad |\cdot 3$

$m + 3n = 3(1 - 3n)$

$m - n = 9m + 18n - 3 \quad |-9m; -18n$

$m + 10n = 3 - 9n \quad |+9n$

$m - 19n = -3$

$m + 19n = 3$

$m + 0n = 0$

15b

Pátá strana 8.13

celkem 15b

1. ... 15b – 13b
2. ... 12b – 11b
3. ... 10b – 6b
4. ... 5b – 3b
5. ... 2b – 0b

Ø známka: 63 : 19 ≐ 3,32

60

30

110

5b

Soustava má nekonečno řešení.

$\frac{3x - 2y}{2} = 6 \quad |\cdot 2$

$\frac{y + 8}{x} = 2 \quad |\cdot x$

$3x - 2y = 12$

$y + 8 = 2x \quad |-2x, -8$

$3x - 2y = 12$

$y - 2x = 8$

$x + y = 8$

$3x - 2y = 12$

$x = 20 + y$

$-2x + y = 8$

$3x - 2y = 12$

$-(20 + y) + y = 8$

$3(20 + y) - 2y = 12$

$-40 - 2y + y = 8$

$60 + 3y - 2y = 12$

$-20 \neq 8$

$60 + y = 12 \quad |-60$

$y =$

Soustava nemá řešení

5b B 4

3

$u+5v) = 3\cdot(1-3v)$

$-v = 9u + 18v - 3$

$4u+5v) = 3\cdot(1-3v)$

$-v = 9u + 18v - 3 \quad /\cdot(-8)$

$+10v = 3 - 9v$

$+8v = 72u + 144v - 24$

$+10v = 3 - 9v$

$18v = 72u + 135v - 21 \quad /-135v$

$-117v = 72u - 21 \quad /-72u$

$72u - 1$

1,5b

ná řešení

$\frac{x-2y}{2} = 6 \quad /\cdot 2$

$x-2y = 12 \quad /+2y$

$3x = 2y + 12 \quad /:3$

$\boxed{x = \frac{2}{3}y + 4}$ ~~nemá jediné řešení~~

$\frac{-2y = 6}{2} \quad /\cdot 2$

$\frac{8}{} = 2 \quad /\cdot 1?$

$-2y = 12$

$8 = 2$

0b

③ $\frac{x+3}{2y-1} = 2$

$3(x-2y) = 2\cdot(3y+2)$

$\frac{x+3}{2y-1} = 2 \quad /\cdot 2y-1$

$3x - 6y = 6y + 4$

$x + 3 = 4y - 2 \quad /-3$

$3x - 6y = 6y + 4$

$x = 4y - 5$

$3\cdot(4y-5) - 6y = 6y + 4$

$12y - 15 - 6y = 6y + 4 \quad /-4$

$12y - 19 - 6y = 6y \quad /:6$

$2y - \frac{19}{6} = y$

$\boxed{-\frac{19}{6} + 2y = y}$

3,5b

15b

$\frac{u-v}{3} = 3u+6v-1 \quad |\cdot 3$

$(u+5v) = 3(1-3v)$

$u-v = 3(3u+6v-1)$

$u+10v = 3-9v \quad |+9v$

$u-v = 9u+18v-3 \quad |-u$

$u+19v = 3$

$-v = 8u+18v-3 \quad |+v$

$3 = 8u+19v$

$0 = 8u+19v-3 \quad |+3$

$3 = 8u+19v$

$3 = 8u+19v \quad |\cdot(-1)$

$3 = 8u+19v$

$-3 = -8u-19v$

$3 = 8u+19v$

$\boxed{0x = 0}$ rovnice má nekonečně mnoho řešení.

5b

$\frac{x+3}{2y-1} = 2 \quad |\cdot(2y-1)$

$(x-2y) = 2(3y+2)$

$x+3 = 2(2y-1)$

$3x-6y = 6y-4 \quad |-6y$

$x+3 = 4y-2 \quad |-4y$

$x-12y = -4$

$x+3-4y = -2 \quad |-3$

$3x-12y = -4$

$x-4y = -5 \quad |\cdot(-3)$

$3x-12y = -4$

$-3x+12y = 15$

$3x-12y = -4$

$\boxed{0y = 11}$

rovnice nemá řešení.

5b

2. $\frac{3x-2y}{2} = 6 \quad |\cdot 2$

$\frac{y+8}{x} = 2 \quad |\cdot x$

$3x-2y = 12$

$y+8 = 2x \quad |-y$

$3x-2y = 12$

$8 = 2x-y$

$3x-2y = 12$

$2x-y = 8 \quad |\cdot(-1{,}5)$

$3x-2y = 12$

$-3x+1{,}5y = -12$

$-0{,}5y = 0 \quad |:(-0{,}5)$

$\boxed{y = 0}$

5b

$\frac{y+8}{x} = 2$

$\frac{0+8}{x} = 2 \quad |\cdot x$

$8 = 2x \quad |:2$

$\boxed{4 = x}$

rovnice má pouze 1 řešení.
Uspořádanou dvojici $[4;0]$

zk: $L_1 = \frac{3x-2y}{2} = \frac{3\cdot 4-2\cdot 0}{2} = \frac{12-0}{2} = \frac{12}{2} =$

$P_1 = \boxed{6}$ $\boxed{L_1 = P_1}$

$L_2 = \frac{y+8}{x} = \frac{0+8}{4} = \frac{8}{4} = \boxed{2}$

$P_2 = \boxed{2}$ $\boxed{L_2 = P_2}$

Desetiminutovka na ZŠ Vesec

① $\frac{u - v}{3} = 3u + 6v - 1 \quad /\cdot 3$

$2(4u + 5v) = 3(1 - 3v)$

$3u - 3v = 6u + 18v - 3 \quad /-3u$

$2\cdot 4u + 2\cdot 5v = 3\cdot 1 - 3\cdot 3v$

$-3v = 3u + 18v - 3 \quad /-18v$

$8u + 10v = 3 - 9v \quad /-9v$

$-21v = 3u + 3 \quad /\cdot(-1)$

$8u + v = 3 \quad /-v$

$21v = -3u - 3 \quad /:21$

~~$8u + v = 3 \quad /-v$~~

$8u = 3 - v$

$v = -\frac{3}{21} - \frac{3}{21} = -\frac{1}{7} - \frac{1}{7}$

$8u = 3 - 0$

$u = \frac{3}{8} - 0 \quad /:8$

0,5b

NEMÁ ŘEŠENÍ

$\frac{3x - 2y}{2} = 6 \quad /\cdot 2$

$\frac{y + 8}{x} = 2 \quad /\cdot x$

$6x - 4y = 12 \quad /-6x$

$yx + 8x = 2x \quad /-yx$

$-4y = 12 - 6x \quad /\cdot(-$

$8x = 2x - yx \quad /:8$

$4y = 12 - 6x \quad /:4$

$\boxed{x = \frac{2x}{8} - \frac{yx}{8}}$

$y = 3 - \frac{6x}{4}$

0b

$\frac{x + 3}{2y - 1} = 2 \quad /\cdot(2y-1)$

$(x - 2y) = 2(3y + 2)$

$(x + 3)\cdot\cancel{(2y - 1)} = 2\cdot(2y - 1)$

$x - 2y\cdot 3 = 2\cdot 3y + 2\cdot 2$

$(x + 3)\ \cancel{2y - 1\cdot(x+3)} = 2\cdot 2y - 1\cdot 2$

$x - 6y = 6y + 4 \quad /\cdot(-1)$

$x + 3 = 4y - 2 \quad /+2$

$3x + 6y = -6y + 4 \quad /+6y$

$x + 5 = 4y \quad /:4$

$-3x = -4 \quad /\cdot(-1)$

~~$\boxed{x = 4y + 5}$~~

$3x = 4 \quad /:3$

$x = \frac{4}{3}$

$\boxed{y = \frac{x}{4} + \frac{5}{4}}$

2,5b

$y = \frac{\frac{4}{3}}{\frac{4}{1}} + \frac{5}{4} = \frac{4}{12} + \frac{5}{4} = \frac{9}{12}$