

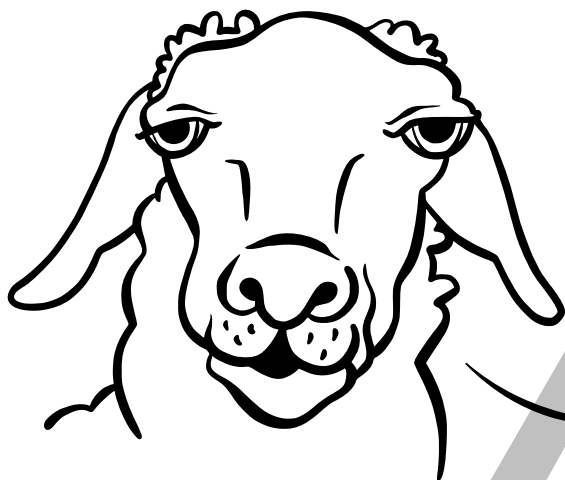
# Programování a matematika

A  
1



Vilém Vychodil  
PM Books s.r.o., 2023

# Programování a matematika A1



Držíte v kopytech první část naší série komiksových učebnic s pracovním sešitem: **Úroveň A, díl 1.** Budeme se spolu učit programovat a procvičíme matematiku!

Jestli zjistíte, že naše učebnice je nad vaše síly, nezoufejte! Zkuste se poradit ve své smečce. Pokud ani to nepomůže, vezměte do tlapek pastelky a použijte učebnici jako antistresové omalovánky.



Vilém Vychodil

PM Books s.r.o., 2023

# Programování a matematika A1

**Autor:** Vilém Vychodil

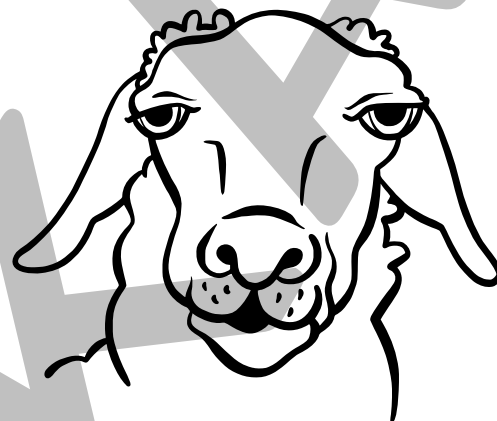
**Ilustrace:** Monika Klimková

**Jazyková korektura:** Zuzana Kačerová

**Ukázka anglické verze:** Nicola Catherine Fořtová, Vilém Vychodil

**Sazba a grafická úprava:** Vilém Vychodil

**Návrh obálky:** Vilém Vychodil



**Vydalo:** PM Books s.r.o., Dittrichova 346/4, Nové Město, 120 00 Praha 2 (<https://prog-mat.cz>)

**Tisk:** Lulu Press, Inc., 627 Davis Drive Suite 300, Morrisville, NC 27560, USA (<https://lulu.com>)

Všechna práva vyhrazena. Žádná část této publikace nesmí být reprodukována a rozšiřována jakýmkoli způsobem bez písemného souhlasu nakladatele, s výjimkou případů zákonem výslovně povolených.

Kapitola *Výrazy a Abaku* navazuje na metodu výuky počtů Abaku®, kterou vyvíjí společnost AL.21 s.r.o. Podrobné informace o metodě, deskové hře a její on-line verzi lze získat na webu projektu (<https://abaku.cz>).

Autorem deskové hry Chov zvířátek (Hodowla zwierząt), o které mluvíme na straně 68, je Karol Borsuk. Její moderní verzi Superfarmář vydává Granna Sp. z o.o. (<https://granna.pl>).

Autorem hry Mastermind (někdy psáno jako Master Mind), o které mluvíme na straně 78, je Mordecai Meirowitz. MASTERMIND je registrovaná ochranná známka vlastněná Hasbro, Inc. (<https://hasbro.com>).

© PM Books s.r.o., Praha 2023

1. vydání

ISBN 978-80-908730-0-1

Milí čtenáři,

na dalších stránkách se můžete ponořit do dobrodružství, které spolu prožívají beran Rohlík a vlk Hrubíáš. Jsou to staří dobráci, kteří se rozhodli, že z nich budou programátoři. Vlastně neměli tak úplně na vybranou, ale brzy zjistili, že je to náramná zábava. Moc bychom vám přáli, abyste sdíleli jejich nadšení a aby vás během čtení našeho komiksu bavilo objevovat nové věci.

Pokud už máte za sebou první třídu, můžete se směle pustit do čtení. Je možné, že některým věcem nebudete hned napoprvé rozumět, ale to je naprosto v pořádku. Naši hrdinové taky všemu nejprve musejí přijít na kloub a někdy jim to dává pěkně zabrat. Mají svého průvodce, pana Lambáka, který je občas popostrčí správným směrem. Pokud si najdete někoho staršího, kdo s vámi bude textem procházet, bude to jen dobře.

Patříte-li mezi starší čtenáře, pak nepochybujeme o tom, že budete schopni projít naši učebnici úplně samostatně. Některé úvodní kapitoly se vám možná budou zdát až příliš jednoduché, ale nenechte se zmýlit, i ony skrývají důležitá poselství a zajímavé příklady. A proč do programování taháme matematiku? Programování jako dovednost a umění souvisí s matematikou na mnoha úrovních, především ale v tom, jakým způsobem nutí člověka uvažovat a rozvíjet své schopnosti abstrakce. A to je přesně to, o co nám jde.

Autor

Milí rodiče, mentoři, učitelé, programátoři, hackeři, geekové a nerdi všeho druhu!

Děkujeme vám za podporu, protože jste to s největší pravděpodobností vy, kdo se postarali o to, aby vaši blízcí, dychtící po vzdělání, dostali naši nekonformní učebnici. Dovolte nám, abychom vám dali několik tipů, jak s naším textem a dodatečnými materiály pracovat.

Základní postup práce je v první řadě přečtení kapitoly, která obvykle seznamuje s novými pojmy. Následuje řešení problémů na papíře. Řada z problémů je vymyšlena tak, aby šlo ověřit jejich správnost na počítači, což vřele doporučujeme. Nelze očekávat, že se čtenář stane ihned programátorem. Ale tím, že ověřuje své úvahy na počítači zapisováním výrazů konkrétního programovacího jazyka, se přirozeně setkává s nejjednodušší formou programování. I když za ústřední považujeme práci s textem, budeme postupně vytvářet interaktivní verze příkladů z učebnice, které budou dostupné na našem webu (<https://prog-mat.cz>).

Některé z úkolů jdou nad rámec příslušných kapitol a jejich účelem je podnítit zvědavost. Není nezbytně nutné, aby čtenáři vyřešili všechny úkoly. Stejně tak je úplně v pořádku, když udělají chybu. Na většinu chyb čtenář časem přijde sám tím, jak se mu postupně rozšiřuje obzor, nebo během samotné interakce s počítačem.

V textu dodržujeme následující typografické konvence:

**nové pojmy** píšeme tučně,

*zdůrazněné pasáže* píšeme skloněným písmem,

matematické výrazy, jako je  $8 + 5 = 13$ , píšeme fontem Computer Modern,

výrazy programovacího jazyka, třeba `(= (+ 8 5) 13)`, píšeme neproporcionálním fontem Inconsolata,

příklady v částech na pocvičení rozdělujeme na **1** jednoduché, **2** standardní a **3** zapeklité.

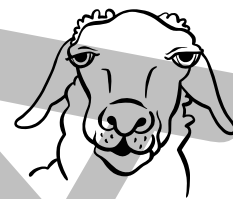
# Rohlík se představuje



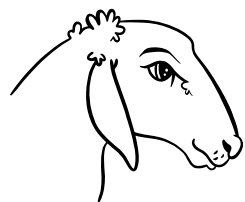
Na jednom hospodářství,



kdesi pod kopcem s mírně radioaktivním podložím,



se narodil beránek plemene Lacaune.



Neměl žádné rohy, a proto mu začali říkat Rohlík.

Zde by příběh mohl klidně skončit. Náš beran byl ale výjimečný. Nikdo vlastně neví proč, ale vlivem záhadné mutace se u něj vyvinul *neokortex*. A tak se z obyčejného sudokopytníka stalo prudce inteligentní stvoření.



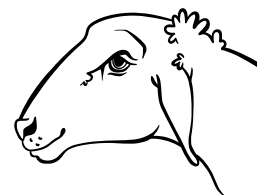
Ahoj, jsem Rohlík. Rád se pasu, žvýkám, přežvykuji, čtu knihy, vedu s bačou debaty o jsoucnu a bytí, a hlavně se rád starám o ovce.

Ale jeho rutinní život, který tolik miloval, se jednoho dne převrátil naruby. Vše začalo nevinným rozhovorem s bačou.



Nazdar Rohlíku! Zrovna jsem s tebou chtěl mluvit. A víš ty co? Možná si na to rovnou sedni.

Ahoj bačo! To zní jako opravdu důležité poselství. Jsem jedno ucho!



Situace je složitá a budeme pro tebe muset najít nové pracovní zařazení.

Jak to? Nedělal jsem snad svoji práci dobře? O ovečky jsem se vždycky příkladně staral.



No právě! Možná až moc dobře. Jistě sis všiml, že prakticky všechny naše ovce jsou tvoje dcery a my samozřejmě nechceme, aby nám tu docházelo k rodovému vyčerpání.

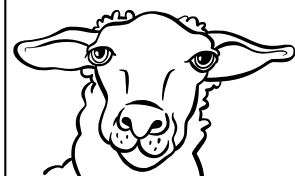


Jasně! Není potřeba zacházet do technických detailů. Vezmeš mě zase někam autem, abych se tam seznámil s novým stádem! Díky, jsi kamarád. Jen si skočím vyčistit zuby a trochu si opláchnout...



Abych pravdu řekl, to jsem zrovna na mysl neměl. Musíme o tvé budoucnosti přemýšlet v kontextu toho, že už nejsi nejmladší.

Já se sice cítím pořád fit, ale změna je život. Co navrhuješ?



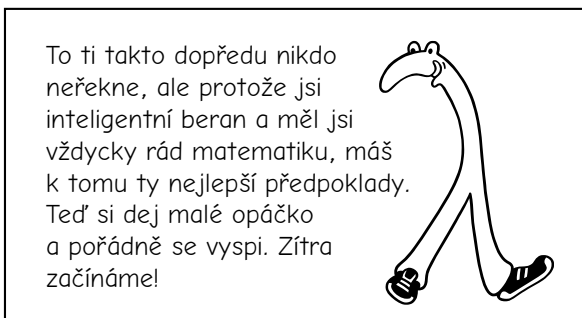
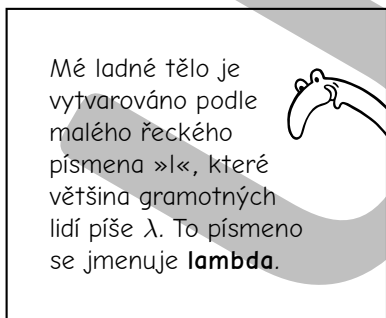
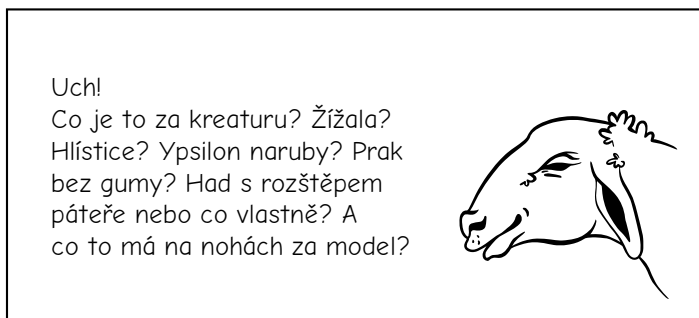
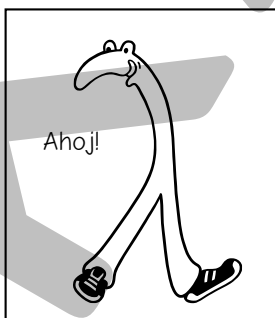
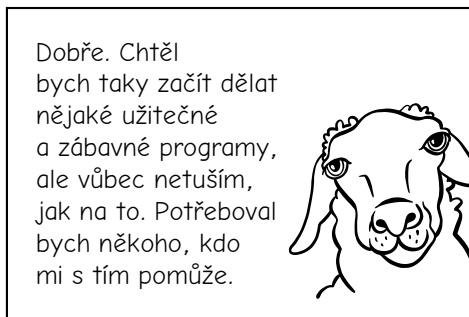
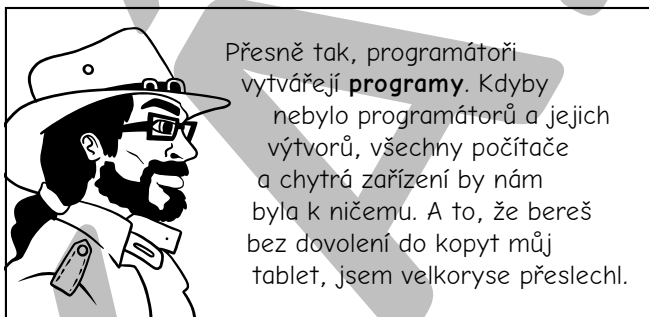
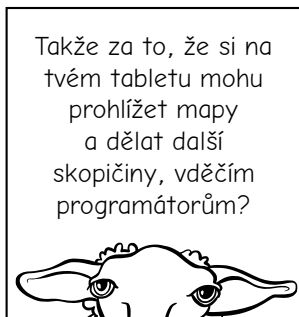
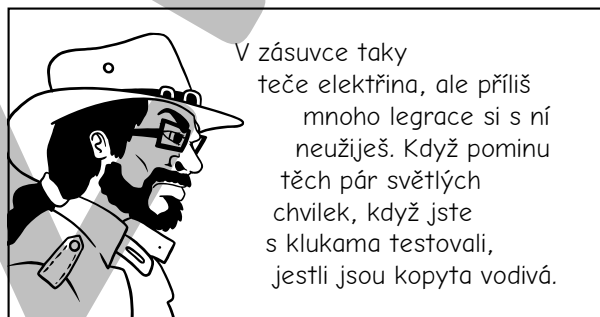
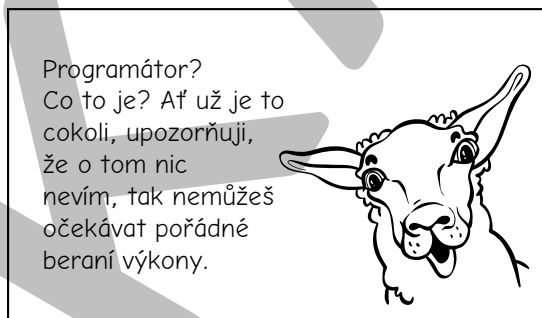
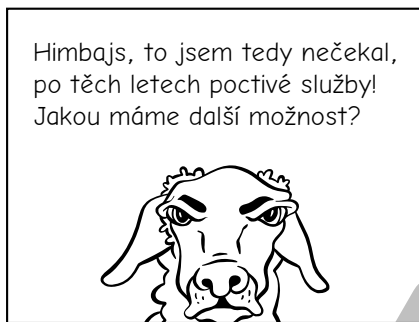
Vidím to tak, že máme v podstatě tři možnosti.



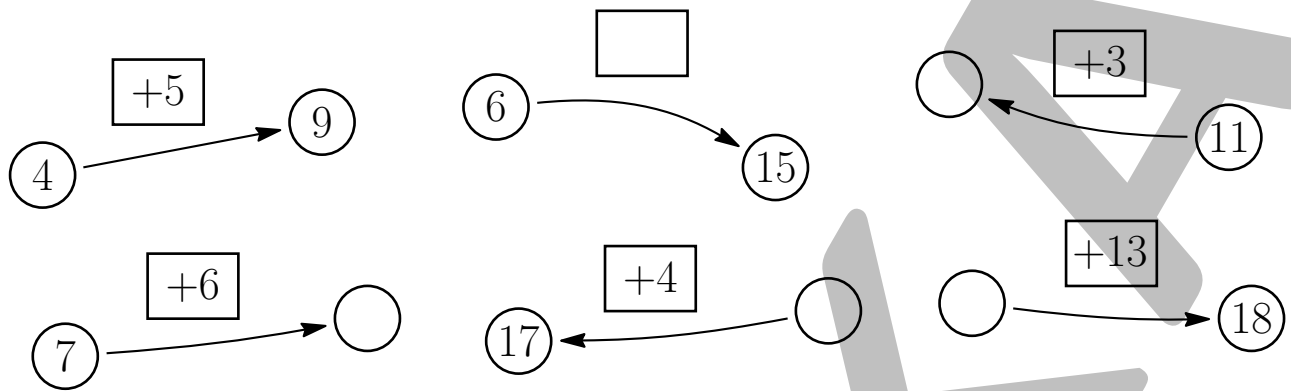
Tou první je pořádná hostina. Navrhuji udělat tvůj oblíbený rogan josh.

Skvělé! Jen bych prosil extra porci česnekového naanu a hodně medvědího česneku navrch. Pořádně si nacpu všechny žaludky! Sice nevím, jak to souvisí s mou prací, ale je mi to jedno.

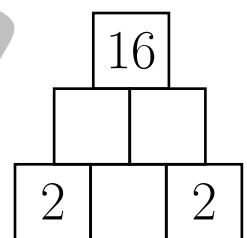
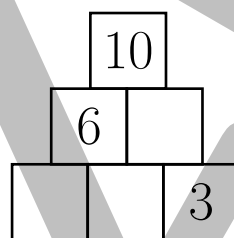
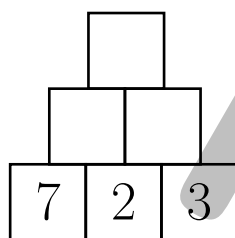
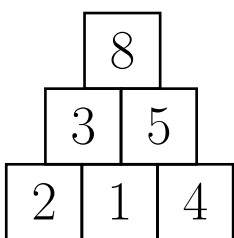




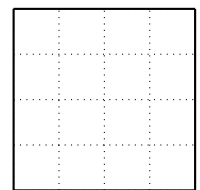
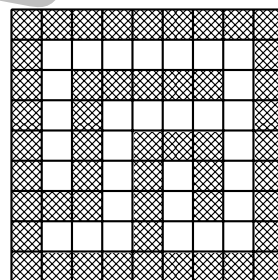
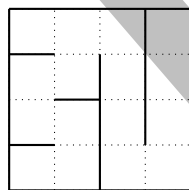
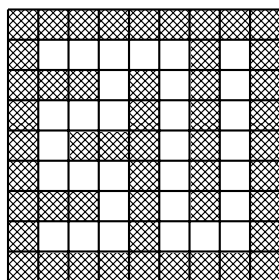
1 Doplň hodnoty do přechodových diagramů. Dej pozor na směr šipek.



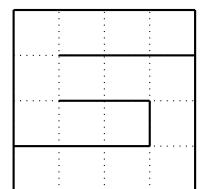
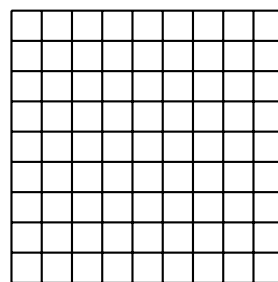
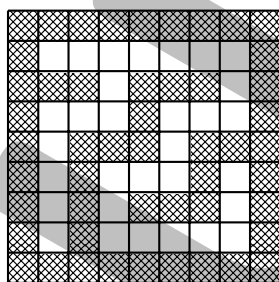
2 Doplň chybějící hodnoty v součtových pyramidách.



3 Překresli bludiště podle vzoru.



4 Doplň čárový a čtverečkový výkres bludiště.



# Vnořené seznamy



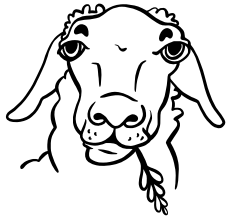
Z počátku jsem byl trochu skeptik, ale ty pravdivostní hodnoty se mi vážně líbí.



Bodejť by ne. Já myslím, že docela často potřebujeme vyjádřit, že nějaké podmínka je splněná, nebo není, ne?



Uvedeš mi příklad?



Tak třeba příklad podmínky: »Je dojírna plně obsazena?« Pokud je podmínka pravdivá, další ovce na vstupu musí čekat. Pokud je podmínka nepravdivá, ovce může dovnitř.

To se mi zdá! Dokážeš si vzpomenout na něco, co se netýká ovčích záležitostí?



Hraješ rád karty?



Jasně a jsem v tom vážně dobrý!



Já rád hraji prší. Zrovna onehdy jsem hrál s ostatními berany a vedli jsme u toho řeči o ovčích.



Proč mě to nepřekvapuje?



Ale to zajímavé ... Když jsme tak hráli a já byl na tahu, tak jsem si vlastně říkal: »Mohu odhodit kartu z ruky?« To byla *podmínka*, která byla buď *pravdivá*, nebo *nepravdivá*. Když byla pravdivá, odhodil jsem kartu. Když byla nepravdivá, musel jsem si brát z balíčku.



Garantuji ti, že takto nad hraním karet většina karbaníků nepřemýšlí.

Ale mě to baví. Líbí se mi, že v každodenních situacích teď z pohledu programátora dovedu rozpoznat nové věci. Dělá to ze mě lepšího berana.

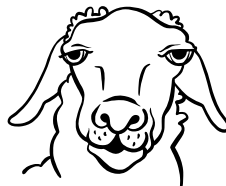


Ano, to jistě všechny tvoje bečící kamarádky z ohrady ocení. Už to slyším, »Ó, Rohlíku, řekni nám ještě něco o programování!«

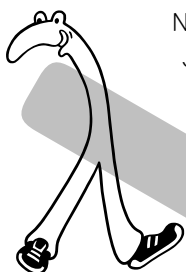
Počkej ale! Vždyť my bychom kromě ovčí agendy mohli na našich počítačích naprogramovat i nějaké hry. Třeba zrovna ty karty nebo něco dalšího.



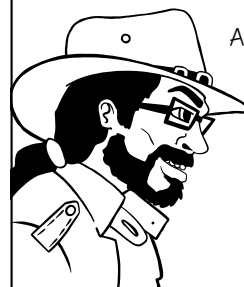
To by bylo skvělé. Má to jen jeden malý háček.



Zřejmě narážíš na to, že toho ještě neumíme tolik, abychom se do takových projektů mohli pustit, co? To ti musím dát za pravdu. Nemám zatím vůbec představu, jak bych něco takového udělal.



Nazdar, opět jsem tu a nebyl bych to já, kdybych neměl nějakou radu nad zlato. Na podobné věci si budete prostě muset chvíli počkat. Nikdo se nenaučil programovat přes noc nebo za pár dní. Chvilí to trvá. Budeme ale postupovat po malých krůčcích a uvidíte, že za chvíli zvládnete dělat mnohem zajímavější věci než teď.



Až budete mít hotovou práci, tak se pro mě za mě zavřete do sklepení a pořádejte si tam herní orgie. Hlavně nezapomeňte, že hlavní tester herního software jsem tady *já*.





No jo no. Ti dva dokážou vlkovi pokazit den. Ale vraťme se k programování. Já jsem si taky všiml jedné zajímavé věci. Když jsme začali používat ty podmínky, tak jsme vlastně začali psát i složitější seznamy, ne?

Jak jako složitější? Myslíš něco, co tvůj vlčí mozek nebyl schopen pobrat?



Ale ne, myslím složitější v tom, jak ty seznamy vypadaly...



Není to vlastně žádná velká věda, ale donedávna jsme vytvářeli pouze seznamy, které obsahovaly jen čísla a symboly. Tím chci říct, že když jsme chtěli zapsat seznam, začali jsme levou závorkou a pokračovali symboly a čísly a nakonec jsme jej ukončili pravou závorkou. V každém zapsaném seznamu tak byla vlastně jedna levá závorka a jedna pravá závorka.



Jasně, příkladem takového seznamu je třeba

$(+ 3 5)$

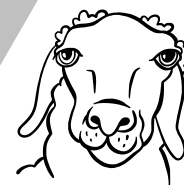
a podobně. Pořád ještě ale nevím, kam míříš.



Tak podívej, co jsem si před chvílí zkusil:

$(= 12 (+ 3 4 5)) \mapsto \text{true}$

Už chápu, co máš na mysli!

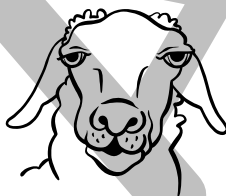


Je to seznam, který obsahuje další seznam!



Souhlasím!

V tomto případě je  $(= 12 (+ 3 4 5))$  vlastně tříprvkový seznam. Jeho prvním prvkem je symbol  $=$ , druhým prvkem je číslo 12 a posledním prvkem je seznam  $(+ 3 4 5)$ . Výsledek  $\text{true}$  mě navíc ani trochu nepřekvapil, číslo 12 je totiž rovno součtu sčítanců 3, 4 a 5.



Podobně

třeba  $(> (+ 3 4) 7)$

je taky tříprvkový seznam.

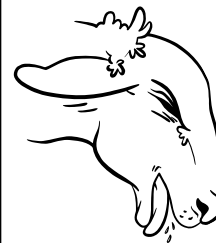
Prvním prvkem je symbol  $>$ , druhým prvkem je seznam  $(+ 3 4)$  a posledním prvkem je číslo 7.



Co jsme to vlastně objevili?

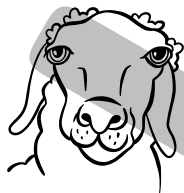


Přišli jste na další docela prostou věc. Seznamy v našem programovacím jazyku mohou obsahovat jako své prvky další seznamy. V takovém případě jim říkáme **vnořené seznamy**. A jak sami uvidíte, každý pořádný program se vlastně skládá z mnoha vnořených seznamů.



Pak se nám v programech začnou množit závorky jako dobře živené ovce.

Ale teď vážně, líbí se mi, jak se seznamy dají pěkně číst zleva doprava.



Nechápu!



No tak třeba ten tvůj seznam

$(= 12 (+ 3 4 5))$

přečtu takto:

»Porovnej 12 s výsledkem součtu čísel 3, 4 a 5.« To je bezva. Začínám programování milovat.



1 Doplň symbolické výrazy. Ověř správnost v REPLu.



$(= (+ 13 14) (+ 11 16)) \mapsto$    
 $(< (+ 3 14) (+ 8 5)) \mapsto$    
 $(\text{ } 8 (+ 2 5)) \mapsto$  true  
 $(> (+ 8 15) \text{ }) \mapsto$  true  
 $(< (+ 8 7) \text{ } (+ 8 9)) \mapsto$  true  
 $(\text{ } (+ 7 4) (+ 4 7)) \mapsto$  false  
 $(= (+ 7 9) (+ 5 \text{ })) \mapsto$  true  
 $(= (+ \text{ } 5) (+ 2 6)) \mapsto$  true  
 $(< (+ 2 8) (+ \text{ } 3)) \mapsto$  true



2 Odhadni výsledky a ověř v REPLu.

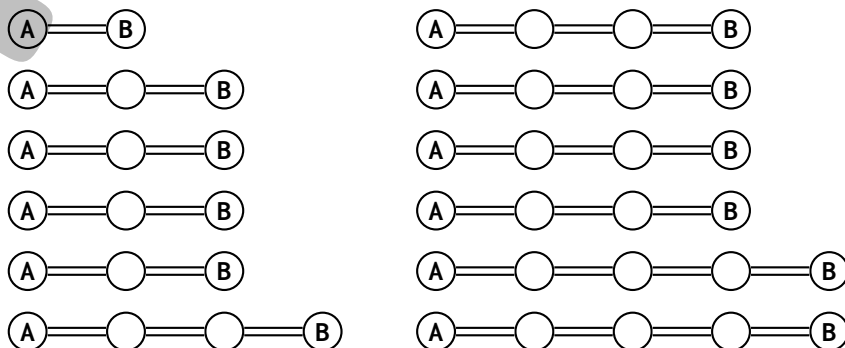
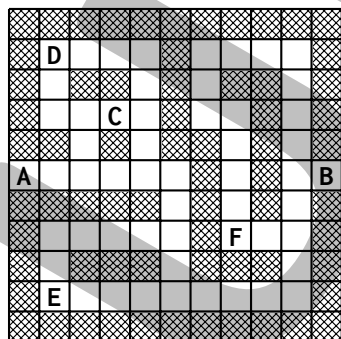
$(< 7 7) \mapsto$    
 $(<= 7 7) \mapsto$    
 $(< 5 7 7 9) \mapsto$    
 $(<= 5 7 7 9) \mapsto$

3 Jakou hodnotu jsme pojmenovali secret?

```

(def secret )
(def x (+ secret 7))
(def y (+ secret x))
y  $\mapsto$  23
  
```

4 Procházej bludištěm tak, abys na každé políčko vstoupil nejvýš jednou. Zapiš všechny cesty ze stanoviště A do B.



# Výrazy a Abaku



Řeknu ti, Rohlíku,  
minule to byla docela hustá  
teorie. Všechny ty funkce,  
aplikace a argumenty.  
Mám pocit, že se mi z toho  
uvaří mozkomíšní mok.



Já jsem  
rád, že jsme nasáli  
trochu té odborné  
terminologie,  
ale navrhuji  
udělat si oddechový  
den. Zahrajeme  
si spolu něco?

Klidně,  
ale krokování  
už mám  
z minula  
dost. Zkus  
vymyslet  
něco  
nevšedního.



Objevil jsem skvělou  
hru, u které můžeme oba  
otestovat svůj důvtip, rychlost  
uvažování a kombinatorické  
schopnosti. Jmenuje  
se to *Abaku* a dá se to hrát ve  
dvou a více hráčích. Bude nám  
k tomu stačit podložka a žetony.



Abaku říkáš? Jsem  
zvědavý, v čem to  
spočívá.



Nebudu teď  
vysvětlovat celá  
pravidla, zaměřím se  
jen na to podstatné.  
Co bys řekl na  
posloupnost žetonů

8 5 1 3?

Jsou to kolečka  
a na každém  
z nich je číslice!



Dokázal  
bys mezi  
číslíci vsunout  
symboly  
operací  
a rovnítko  
tak, aby vznikla  
rovnost?



Radši mi to,  
prosím, předved',  
dnes ráno nejsem  
v nejlepší kondici.



Tak sleduj:

$$\textcircled{8} + \textcircled{5} = \textcircled{1} \textcircled{3}$$

neboli  $8 + 5 = 13$ .



Aha! Tak to jsi  
měl na mysli! To  
je samozřejmě  
jednoduché.  
Dej mi, prosím,  
další příklad.

Co třeba

1 7 9 8?



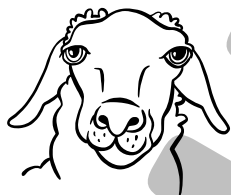
Mám to:

$$\textcircled{1} \textcircled{7} - \textcircled{9} = \textcircled{8}$$

neboli  $17 - 9 = 8!$



Nebo by to šlo  
vyjádřit jako  $17 = 9 + 8$ .  
To je v podstatě základní  
princip Abaku. Dostaneš  
do kopyt či pacek žetony  
s číslicemi a vykládáš je na  
herní plán tak, abys tvořil  
co možná nejdelší správné  
aritmetické výrazy.



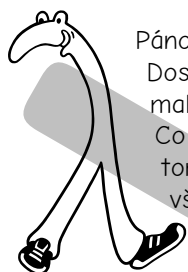
To je  
panečku  
zajímavá  
hra.  
Mohu se  
přidat?



Beze všeho.



Toho dne strávili  
naši tři kamarádi  
několik hodin nad  
novou hrou. Na  
sklonku příjemně  
stráveného  
odpoledne dostal  
Lambdák nápad.



Pánové!  
Dostal jsem  
malý nápad.  
Co kdybychom  
tomu  
všemu dali ještě  
trochu šřávu?

Šřávu? To jako  
kávu s ovčím  
mlékem?



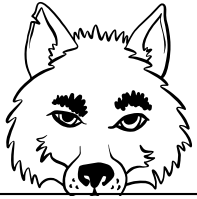
Šřávu? To jako  
jehněčí vývar?



Pojďme tu hru  
spojit s programováním!  
Mám posloupnost  
číslíci - vaším úkolem  
je sestavit symbolický  
výraz obsahující tyto číslice  
ve stejném pořadí a to  
tak, aby byl výraz pravdivý.



Tak prosím začni a předved' nám to!



Bez problému, dejme tomu, že mám v ruce žetony

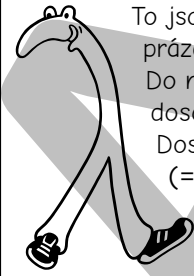
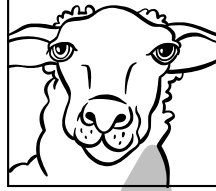
①, ⑤, ⑦ a ⑧.

Vymyslím k nim výraz:

$(= (- \square \square \square) \square)$



No a co s ním? Co jsou ty čtverečky?



To jsou jakoby prázdná místa. Do nich dosadím žetony. Dostanu tak výraz  $(= (- 15 7) 8)$ , který je pravdivý.

Ano, to platí!

Když tento výraz zadáme do REPLu, získáme:

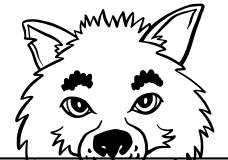
$(= (- 15 7) 8) \mapsto \text{true}$



Takže, Hrubíáši, dávám ti žetony ①, ②, ⑥ a ⑧. Vytvoříš mi k nim výraz?



Jasně, to je hračka:  $(= (* 2 8) 16)$



Pokud nás jednoduché výrazy omrzí, můžeme hledat složitější. Rohlíku, co bys vytvořil z žetonů ②, ⑥, ⑦ a ⑧?



Nechci tě podceňovat příteli, ale to už nevyřešíš.



Kdepak, přišel jsem na to a určitě bys to taky zvládl. Dám ti nápovědu. Použij následující výraz a vlož do něj cifry ve správném pořadí:

$(= (* \square \square) (+ \square \square))$



Beran zase překvapuje! Nesmím se od něj nechat zahanbit...

Už to mám! Pro jistotu ale zadám své řešení do REPLu a přesvědčím se o jeho správnosti:

$(= (* 7 2) (+ 6 8)) \mapsto \text{true}$

Musím říct, že Abaku se mi vážně líbí.



A mně se líbí, jaký jsme do něj udělali vzhled pomocí symbolických výrazů.



Jedna nesouvisící poznámka na okraj. Jistě jste si oba všimli, že naše výrazy začínají být čím dál tím delší. V takových situacích se snažíme dělit je na víc řádků. Nemusíme mít vůbec strach, prvky v seznamech lze totiž oddělovat nejen mezerami, ale i novými řádky.

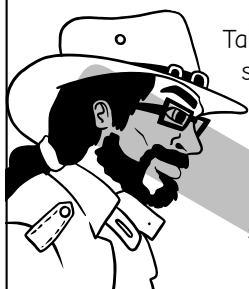


Jak takové dělení vypadá v praxi?



Třeba tak, že místo dlouhé nudle  $(= (* 7 2) (+ 6 8))$ , jak jsi ji napsal, Hrubíáši, výraz rozdělíš na dva řádky:

$(= (* 7 2) (+ 6 8))$

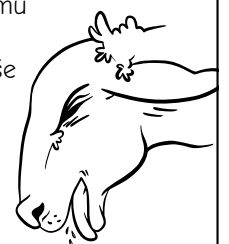


Tak, kdo z vás zvířat si dnes se mnou večer zahraje pořádný karban? Mám pocit, že dnes se na mě štěstěna usměje.



Pane! Kdepak karty, pane! Objevili jsme hru s menším podílem prvku náhody, pane! Uplatníte v ní své pověstné logické myšlení, pane!

Tady někomu asi moc kručí v břiše a přeje si k večeři od pána pořádnou porci.



**1** Doplň hodnoty ze žetonů ve správném pořadí tak, aby byly všechny uvedené symbolické výrazy pravdivé.

(1) (3) (5) (8)  
 (= (+ 5 8) 13)

(1) (5) (7) (8)  
 (= (+ □ □) □ □)

(1) (4) (5) (9)  
 (= (- □ □ □) □)

(2) (3) (7) (9)  
 (= (\* □ □) □ □)

**2** Doplň žetony (3), (5) a (8) do výrazů. Najdi všechna řešení.

(= (+ □ □) □)  $\mapsto$  true

(= (+ □ □) □)  $\mapsto$  true

(= (- □ □) □)  $\mapsto$  true

(= (- □ □) □)  $\mapsto$  true

(= □ (+ □ □))  $\mapsto$  true

(= □ (+ □ □))  $\mapsto$  true

(= □ (- □ □))  $\mapsto$  true

(= □ (- □ □))  $\mapsto$  true

**3** Použij cifry ze žetonů a podle vzoru sestav pravdivé symbolické výrazy.

(1) (2) (4) (6) (8)      (= (+ 16 8) 24)  $\mapsto$  true

(2) (2) (3) (5) (7)

(3) (3) (4) (5) (8)

(1) (2) (3) (6) (7)

(4) (4) (5) (5) (9)

**4** Ze kterých skupin žetonů sestavíš pravdivý výraz?

(1) (2) (6) (8)

(2) (4) (6) (7)

(3) (7) (8) (9)

(1) (3) (6) (8)

(1) (2) (3) (4)

**5** Ze žetonů (1), (2), (4) a (9) sestav pravdivý výraz.



# Pořadí

Bééééé-ktory!  
Bééééé-ktory!  
Bééééé-ktory!



Co psychické  
zdraví, Rohlíku?  
Pořád slouží?



Bez obav! Jsem v prvotřídní kondici! I když je mi trochu líto, že se pomalu dostáváme na konec. Zrovinka teď, když jsme objevili tak báječnou věc jako *vektory*. Když si pozorně poslechněš ovce v ohradě, o ničem jiném se už nechtějí bavit!



S vektory je opravdu zábava! Docela se mi ulevilo, když jsem zjistil, že je *zapisujeme* podobně jako seznamy. Je pěkné odrazit se od něčeho, co už znám.



Máš pravdu, dalo by se říct, že se pouze používají jiné závorky – hranaté místo kulatých. To by ale bylo příliš zjednodušující a taky trochu zavádějící. Z hlediska významu jsou totiž *seznamy* a *vektory* úplně jiné!

Bude k tomu ještě doplňující výklad od Lambďáka?



Jistě! Rohlík to hezky vystihl. V tomto okamžiku je to tak, že *seznamy* píšeme proto, abychom *vytvářeli programy*. Všechny ty báječné aplikace funkcí, díky kterým se něco děje, jsou iniciovány během vyhodnocování seznamů v našem programu.



A vektory?



Naproti tomu *vektory uchovávají data*. Každý vektor je něco jako taška, do které můžeš napat data pro další použití.



Ta analogie s taškou se mi ale moc nelíbí. Mně se v tašce věci obvykle promíchají, kdežto data ve vektoru jsou v něm uchována pěkně v tom *stejném pořadí*, v jakém jsem je tam dal.



Skvělý postřeh, máš úplnou pravdu. Omlouvám se za své vágní vyjadřování.

Ale no tak, přestaňte se vy dva škádlit, moje bránice by to nemusela vydržet!



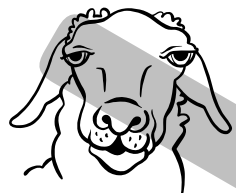
Neboj, jsme profesionálové, vždy konstruktivní a bez emocí. Nepočítám samozřejmě situace, kdy chci někoho zakousnout, protože mě obere o peníze během pěkně vypečené hry.



No dobře... Pojdme se ale vrátit k pojmu **pořadí**. Vektory jsou skutečně struktury, ve kterých jsou hodnoty uloženy v *konkrétním pořadí*. Například

$[3 \ 5 \ 8]$  a  $[3 \ 8 \ 5]$

jsou *dva různé vektory* – obsahují sice stejné hodnoty, ale v jiném pořadí!



Rozumím! Provedeme ti s Hrubíášem detailní rozbor situace. V obou případech máme vektory, které začínají trojkou. Vědecky řečeno – jejich *prvním prvkem* je číslo 3. Ale už druhý prvek je jiný, v prvním případě je to pěťka, v druhém případě je to osmička.

Souhlasím!

Druhý prvek vektoru  $[3 \ 5 \ 8]$  je číslo pět a třetí prvek vektoru  $[3 \ 5 \ 8]$  je číslo osm.

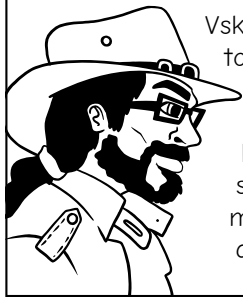
Druhý prvek vektoru  $[3 \ 8 \ 5]$  je číslo osm a třetí prvek vektoru  $[3 \ 8 \ 5]$  je číslo pět.

Tím slavnostně ukončuji náš rozbor!

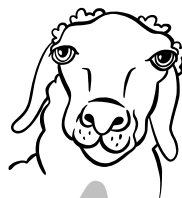




To byl ovšem rozbor od skutečných znalců! Děláte mi náramnou radost!



Vskutku, vypadá to, že na našem hospodářství došlo k nečekané synergii mezi býložravci a predátory!



Bodej by ne! Už jsme se taky něco naučili a dokážeme z našich znalostí vyvozovat další užitečné závěry. Třeba já dedukuji, že od sebe můžeme prvky ve vektorech oddělovat i více mezerami a dělit jejich zápis na víc řádků!

Správná úvaha! Vzpomínáte si ještě na čárky? Když zapisujeme vektory, jejich prvky od sebe můžeme oddělovat pomocí mezer, nových řádků a taky libovolného počtu čárek! Funguje to tedy podobně jako u seznamů.



[3 5 8]                      [3 5  
[3,5,8]                      8]  
[3 5,8]                      [,3,  
[3,5,8,]                      ,5,  
[3,,5,, ,8]                      ,8,]

Jako světaznalá border kolie vám řeknu, ten zápis s čárkami se mi líbí. Něco podobného jsem zahlédl i v učebnicích vyšší matematiky.



Tříprvkový vektor obsahující čísla tři, pět a osm v tomto pořadí můžeme zapsat mnoha způsoby. Nejpřehlednější z nich jsou [3 5 8] a [3,5,8].



Hrubiáši, jsem rád, že jsi neustrnul pouze u základního psího vzdělání! Vektor je opravdu pojem, který se používá v matematice. Vektorové prostory jsou krásné struktury založené na Abelových grupách a číselných tělesech... Pokud máš rád kryptografii, zajímat tě určitě budou i prostory nad konečnými tělesy...

Fascinuje mě váš rozhled, pane, to se jen tak nevidí, pane! Pane, cítím se ve vaší společnosti naprosto nicotný, pane!



To už tu dlouho nebylo!



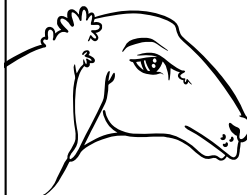
Pojďme dál rozšířit naše úvahy! Jaké typy hodnot můžeme ve vektorech ukládat?



Co jiného než čísla bys do nich chtěl dát?



Zrekapitulujme si, jaká známe *atomická data*. Kromě čísel totiž už známe *pravdivostní hodnoty* a *klíčová slova*. Přímou se nabízí, že bychom mohli vytvářet vektory pravdivostních hodnot a vektory klíčových slov. Nezní to zajímavě?



K čemu by to bylo dobré?



Já bych nápad měl! Pomocí klíčových slov budu reprezentovat barvy, třeba :red (červená), :blue (modrá), :green (zelená), :yellow (žlutá) a podobně. Každé takové klíčové slovo si můžeš představit jako jeden *barevný kolíček* a vektory takových klíčových slov reprezentují *posloupnosti barevných kolíčků*. To je základ mé oblíbené hry, ve které se lámou šifry!




To jsi mi moc radost neudělal! Já vidím svět jen ve žluté a modré! Navíc tu tvoji hru vůbec neznám!





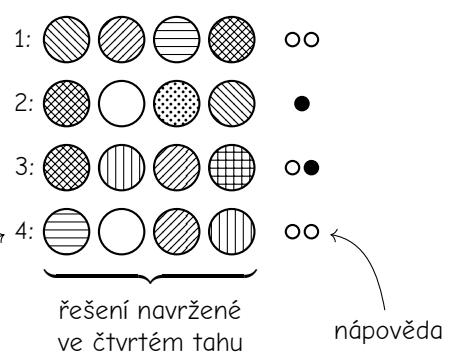
Pouze ve žluté a modré? Má to něco do činění s tím, že pocházíš z Ukrajiny?



Souvislost je zde pouze zdánlivá. Tak vidí svět všechna vypelichaná psiska!




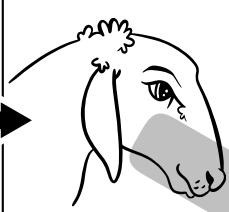
Moment! Všimli jste si, že máme stránku navíc? Skvělá příležitost k tomu, abychom Hrubíášovi vysvětlili principy naší hry! Ukážeme si navíc souvislost s vektory a trochu přičichneme k tomu, jak by se dala hra naprogramovat.



1: ○○  
 2: ○ ●  
 3: ○●  
 4: ○ ○○

číslo tahu      řešení navržené ve čtvrtém tahu      nápověda

Podívejte na ukázkou partie, kterou jsem rozehrál s bačou. Já jsem pán šifry, bača během svých čtyř tahů navrhl řešení, která jsem mu vždy ohodnotil. Zatím se ani jedno neshoduje s mojí šifrou, ale zdá se, že na to brzy přijde!

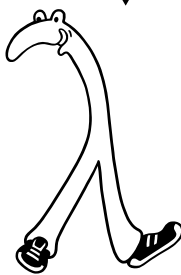



Jedná se o *asymetrickou* hru pro dva hráče. To znamená, že každý z hráčů má jinou roli. První hráč, budeme mu říkat »pán šifry«, si vymyslí *tajnou šifru*, která se skládá z několika různých barevných kuliček. Úkolem druhého hráče, kterému budeme říkat »lamač šifry«, je šifru odhalit neboli *rozlomit pouze na základě nápovědy*.

Důležité je pochopit systém nápovědy. Po každém tahu lamače šifry mu pán šifry tah ohodnotí pomocí *černých a bílých* kamenů:

**černý** kámen se přiděluje za každou *správnou barvu na správném místě*,

**bílý** kámen se přiděluje za každou *správnou barvu, která je na špatném místě*.



Protože je náš komiks černobílý, budeme si barvy vyznačovat pomocí rastrování:




- bílá* :white
- růžová* :pink
- červená* :red
- modrá* :blue
- zelená* :green
- žlutá* :yellow
- oranžová* :orange
- černá* :black

To je skvělé, můžeme si vzít do kopyt pastelky a všechno pěkně vybarvit.


Pro ty z nás, kteří mají problém s vnímáním barev, chci dodat, že můžeme hru hrát klidně i s čísly!

Když každé barvě přiřadíme číslo, bude to sice vypadat jinak, ale je to pořád stejná hra.




Já vám to rovnou předvedu! Takto by vypadala předchozí rozehraná partie, ve které jsem systematicky změnil barvy na čísla:

1: (5) (3) (4) (7) ○○  
 2: (7) (1) (6) (5) ●  
 3: (7) (2) (3) (8) ○●  
 4: (4) (1) (3) (2) ○○




Asi vám už došlo, že nejde o jednu hru, ale celou *parametrickou třídu her*, které se odlišují třeba v délce šifry, v počtu prvků, ze kterých můžeme šifru skládat, a také v tom, jestli můžeme v šifrách prvky opakovat! Volbou parametrů můžeme získat jednoduché i velmi těžké hry!




Už mi to došlo! Až budeme naši hru programovat, všechny šifry můžeme v programu reprezentovat vektory!

Třeba šifru mohou reprezentovat vektorem:  
 [:pink :black :blue :red]

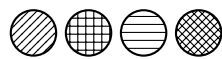
Přesně tak! Já bych k naší hře ještě něco dodal. Původně se jmenovala »Býci a krávy« a hrála se s tužkou na papíře. Hra je víc než sto let stará, její moderní verzi Mastermind vymyslel Mordecai Meirowitz. U nás je známá pod jménem Logik.




Mordecai Meirowitz

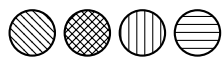


**1** Zahraj si na pána šifry! Pro každou dvojici skládající se z šifry (nahore) a pokusu (dole) doplň nápovědu.



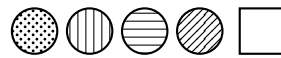
3 2 5 1

8 2 5 3 □



7 3 1 4

1 4 3 7 □



[2 8 1 3]

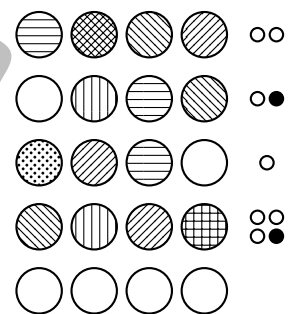
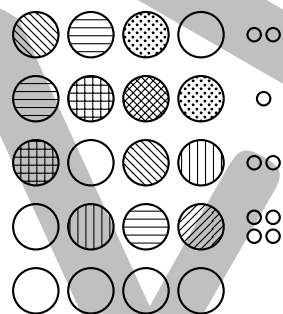
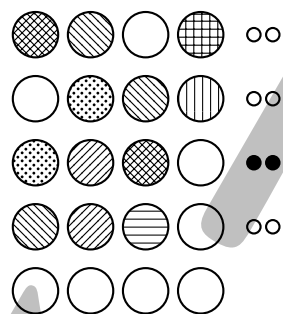
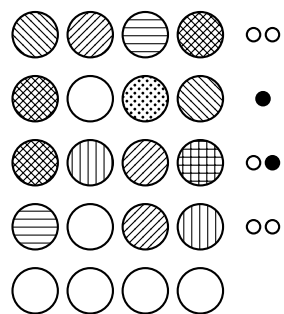
[8 3 1 5] □



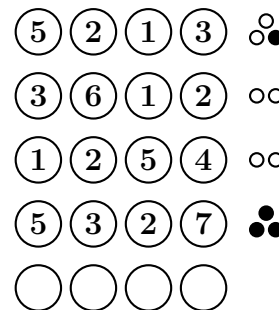
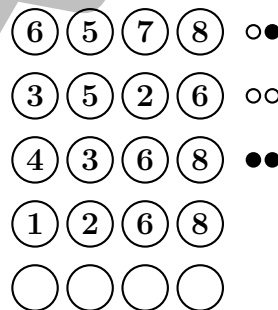
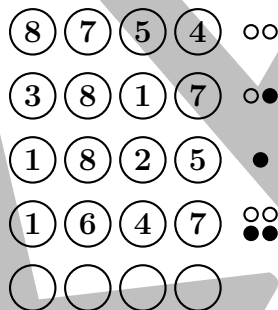
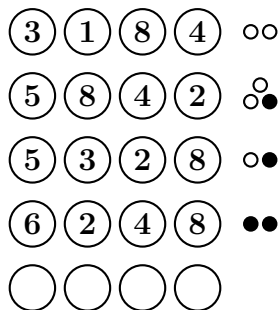
[1 6 4 8]

[2 5 8 3] □

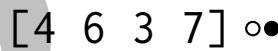
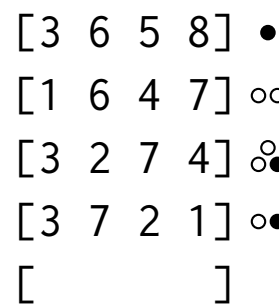
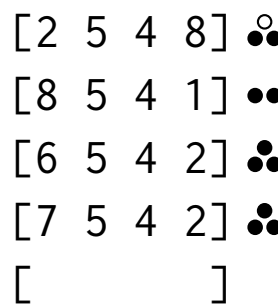
**2** Pokus se rozlomit následující šifry. Každý vzor se v šifře může vyskytovat nejvýš jednou.



**3** Pokus se rozlomit následující šifry. Každé číslo se v šifře může vyskytovat nejvýš jednou.



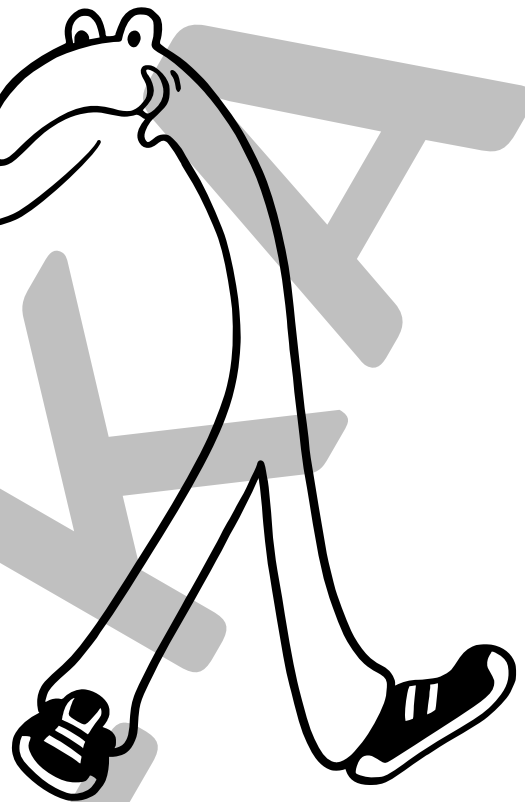
**4** Pořád stejná úloha. Tentokát jsou šifry reprezentovány vektory čísel.



Nechte stádo za sebou  
a poříd'te si naše učebnice  
v češtině i angličtině!



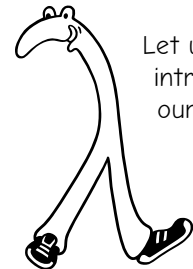
<https://prog-mat.cz>



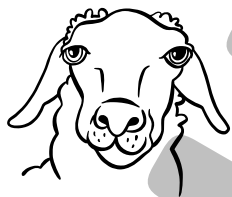
## Programming and Mathematics



You can also enjoy  
our educational stories  
in English. If you're  
looking for a way to  
practice your language  
skills and train your  
brain, look no further!



Let us  
introduce  
ourselves.



Hi, I'm Ramsy. One day  
I got an offer I couldn't refuse,  
so I became a programmer.  
And I'm hooked. It's almost  
as good as fooling around  
with sheep or teasing wolves!



I'm Barky, a domesticated  
wolf and Ramsy's main chum.  
We study together and often  
try and find solutions to various  
problems. I'm pretty into it! It's  
fun, but if you tease me too  
much, I go into attack mode.



The two of them were  
made for each other. If  
it weren't for me, the  
savvy farmer, there  
would be complete  
animal anarchy.  
When I say jump,  
everyone says how high!

Hey, I'm Mr. Lambda!  
A benevolent mentor,  
who sometimes has the  
task of steering the gang in  
the right direction, providing  
hints, or introducing them to  
new concepts. My greatest  
joy is when I see things  
suddenly click for them.



You can find us at



<https://prog-mat.com>

Líbí se vám naše učebnice?  
Podpořte nás a získajte  
přístup k dalším materiálům!



<https://patreon.com/progmat>



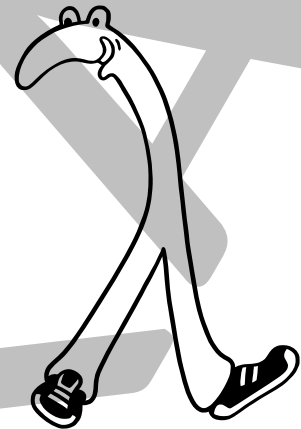
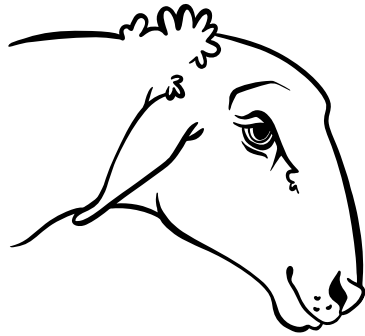
<sup>®</sup>**ABÁKU**

<https://abaku.cz>

Počítejte s námi!

## Obsah

Rohlík se představuje	4
Co je REPL?	7
Nečekané setkání	10
Čísla a symboly	13
Seznamy	16
Chybička se vloudila	19
Nová jména	22
Pravda, či nepravda?	25
Vnořené seznamy	28
Násobení	31
Odčítání	34
Funkce, aplikace a argumenty	37
Výrazy a Abaku	40
Hlavně přehledně!	43
Části celku	46
Pořad stejná úloha	49
Negace	52
Konjunkce	55
Sudá, nebo lichá	58
Znovu u konjunkce	61
Disjunkce	64
Klíčová slova	67
Jak na data?	70
Vektory	73
Pořadí	76



## Programování a matematika A1

**Autor:** Vilém Vychodil

**Ilustrace:** Monika Klimková

**Jazyková korektura:** Zuzana Kačerová

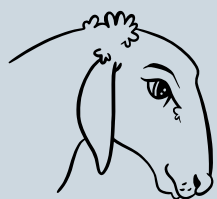
**Sazba a grafická úprava:** Vilém Vychodil

**Vydalo:** PM Books s.r.o., Dittrichova 346/4, Nové Město, 120 00 Praha 2 (<https://prog-mat.cz>)

**Tisk:** Lulu Press, Inc., 627 Davis Drive Suite 300, Morrisville, NC 27560, USA (<https://lulu.com>)

**Rok prvního vydání:** 2023

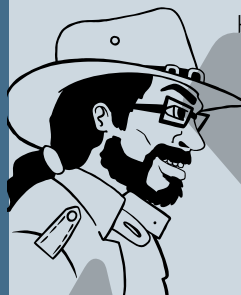
ISBN 978-80-908730-0-1



Tak jsme se zdárně prokousali celým sešitem.



Já jsem se prokousal, ty ses nanejdvůš tak prožvýkal!



Hlavně nezapomeňte, že všichni hrdinové v tomto postmoderním didaktickém díle jsou smyšlení. Jakákoli podobnost se skutečnými postavami, žijícími nebo mrtvými, je čistě náhodná.

To je pravda. Navíc bych vás chtěl poprosit, abyste svoje miláčky nevystavovali ionizujícímu záření. Neokortex jim potom s největší pravděpodobností nenaroste.



A pokud někdo chcete dovážku jídla, tak se na mě prosím neobracejte. Všechny ty nevyžádané telefonáty mi už lezou na nervy.



Teď to hlavní: Co jsme se vlastně naučili?



Prožili jsme blížká setkání s čísly, symboly, pravdivostními hodnotami, seznamy a vektory.



Taky jsme se naučili používat funkce. Sice je ještě neumíme vytvářet, ale to se určitě časem naučíme. Líbí se mi, že jsme taky nasáli nějakou odbornou terminologii. Musím přiznat, že obvykle si nové pojmy nezapamatují hned napoprvé, ale když se s nimi setkávám častěji, začínají mi pak připadat samozřejmé. Miluji, když se dokážu vyjadřovat pěkně odborně, dělá to ze mě lepšího vlka.

Jestli mě něco opravdu chytlo, pak to byly ty vektory. Cítím v kostech, že s nimi půjde dělat spousta zajímavých věcí. Budeme v tom pokračovat?



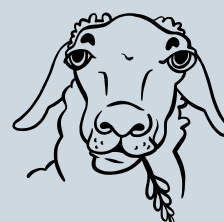
Určitě! Budeme si povídat o různých hrách a jejich stavu. Při reprezentaci stavu her využijeme vektory a ukážeme si další užitečné datové struktury.

A potkáme další kamarády?



Myslíš přidat další kamarády do vašeho hvězdného týmu? To by určitě šlo zařídít!

Pane! Skvělé, pane! Líbí se mi, jak všechno umíte zařídít, pane!



Zatím tedy na shledanou! Pokud jsme na vás udělali dojem, nezapomeňte se o nás zmínit před svými přáteli!



Programování a matematika A1

© PM Books s.r.o., Praha 2023

<https://prog-mat.cz>

ISBN 978-80-908730-0-1



9 788090 873001