Kategorie mladší

Úloha 1A Amfory

Už na začátku má Tomáš v zelené amfoře 0 l vody. Lehce získá 10 l, stačí naplnit zelenou amforu. Pro získání 8 l a 3 l mu stačí naplnit příslušnou amforu a přelít ji do zelené. Zbylá množství musí Tomáš získat buď „sčítáním“ amfor, tj. sléváním vody z více amfor do jedné, nebo „odčítáním“ amfor, tj. odléváním části vody do jiné amfory.

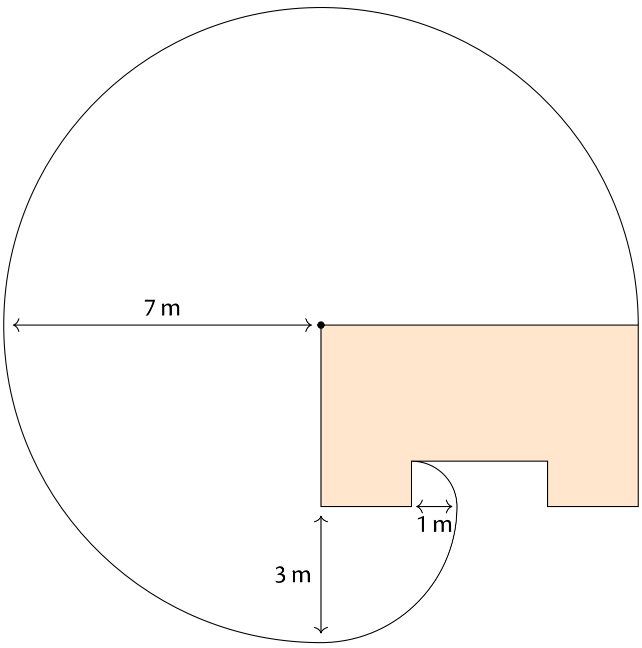
Sléváním může získat třeba 6 l a 9 l, a to tak, že naplní červenou amforu, její obsah přelije do zelené amfory, znovu ji naplní a její obsah přilije do zelené amfory, kde získá 6 l (pro 9 l přilije obsah červené amfory do zelené amfory ještě jednou). Odléváním dokáže Tomáš získat třeba 2 l, 5 l nebo 7 l. 2 l získá tak, že naplní zelenou amforu a část jejího obsahu přelije do žluté amfory. V zelené amfoře mu tak zbydou 2 l. Stejně získá i 7 l, s tím rozdílem, že místo do žluté amfory vodu odlije do červené. 5 l získá Tomáš přelitím vody ze žluté amfory do červené, ale nesmí zapomenout přelít správný objem vody ze žluté amfory do zelené. 4 l získá Tomáš naplněním zelené amfory a jejím postupným odléváním do červené (naplněnou zelenou odlije do červené, tu vylije a znovu do ní odlije vodu ze zelené, ve které tak zůstanou požadované 4 l). Když z takto vytvořených 4 l odlije vodu ještě jednou do červené amfory, dostane 1 l. Tomáš tedy dokáže do zelené amfory odměřit jakýkoliv celočíselný objem od 0 do 10 l.
Pozn. Některé objemy lze získat i jinými postupy.

Úloha 2A Zaječice zalévá

Kristýna otočí kolem o 2 otáčky, protočí se tedy celkem $2 \cdot 20 = 40$ zubů. Tím se desetizubé kolečko otočí o $40 : 10 = 4$ otáčky a protože je na něm cívka o obvodu 50 cm, zvedne sud o $4 \cdot 50 = 200$ cm. Adam tedy musí sud zvednout do stejné výšky, aby zůstal vodorovně. Obvod cívky patnáctizubého kolečka je 80 cm, proto se musí otočit $200 : 80 = 2{,}5$krát. Dále spočítáme počet zubů za tyto 2,5 otočky, vydělíme počtem zubů velkého kola (27) a tím získáme počet otoček dvacetisedmizubého kola. Tento počet pak vynásobíme 360, abychom získali otočení ve stupních (1 otočka má 360 stupňů). Adam tedy musí otočit velkým kolem o $2{,}5 \cdot 15 : 27 \cdot 360 = 500$ stupňů.

Úloha 3A Kozel zahradníkem

Pokud chce Mustafa posekat co nejvíce trávy, tak je nejvýhodnější využít v každé chvíli celých 7 m kabelu. Nejlepší tedy je sekat kružnice nebo jejich části. Když začne od severozápadního rohu domu, tak může posekat $\frac{3}{4}$ obsahu kruhu o poloměru 7 m bez toho, aby někde musel zahýbat za roh (zahnutím za roh se manipulační délka kabelu zkrátí). Okolo jihozápadního rohu domu může Mustafa vysekat čtvrtkruh (plocha na západ od západní stěny je už posekaná) o poloměru 3 m (aby se Mustafa dostal na jih od domu, musí kabel natáhnout podél západní zdi, která je dlouhá 4 m, a s takto nataženým kabelem teprve může sekat čtvrtkruh). Kabel Mustafovi vystačí ještě na zahnutí do výklenku domu, kde může vysekat čtvrtkruh o poloměru 1 m (viz obrázek 1). Nikam dál už s takto dlouhým kabelem nedosáhne. Když jednotlivé části sečteme, vyjde nám, že Mustafa poseče $\frac{3}{4} \cdot 7^2 \cdot \pi + \frac{1}{4} \cdot 3^2 \cdot \pi + \frac{1}{4} \cdot 1^2 \cdot \pi = \frac{157}{4} \cdot \pi \doteq \frac{147}{4} \cdot 3{,}1416 = 123{,}3078\,\mathrm{m}^2$ trávníku.



Obrázek 1: Oblast trávníků, kterou Mustafa poseče.

Úloha 4A Cesty na síti

Martin se na křižovatku, z níž vyrazil, určitě někdy vrátí. K tomu, abychom to ukázali, není (příliš) důležité, jak přesně vypadá řetězec písmen, podle něhož cestuje – rozmyslíme si, že důležité je pouze to, jak a kolikrát při cestování zatáčí.

Ukažme si to nejprve na příkladu kratšího řetězce *VLRRV*. Podle tohoto řetězce se jedním jeho projitím celkově přesune (pokud na začátku hleděl na sever) o jednu křižovatku na východ a dvě na sever (viz obrázek 2a). Tento celkový posun si lze představit jako šipku, jejíž délka a směr je dána tím, jak vypadá Martinův řetězec. Povšimněme si ale, že poté bude hledět na východ – v řetězci *VLRRV* se nachází dvě *R*, ale pouze jedno *L*, tedy se musel během cesty celkově otočit o 90° doprava. Pokud nyní bude pokračovat podle tohoto řetězce znovu, vykoná stejnou cestu, pouze otočenou o 90° doprava, tedy přesune se o jednu křižovatku na jih a dvě na východ. Tento přesun si lze opět představit jako šipku – je stejně dlouhá jako v předchozím případě, ale otočená o 90° doprava. Stejný proces se zopakuje ještě dvakrát a Martin se vrátí zpět do počátečního bodu cesty (viz obrázek 2b) – to lze znázornit tak, že příslušné čtyři šipky dohromady složí čtverec, neboť jsou každá stejně dlouhá a oproti té předchozí otočená o 90° doprava. To, že se Martin vrátí do počátečního bodu, tedy plyne už z toho, že se po jednom projití řetězce otočí o 90° doprava – nepotřebujeme ani vědět, jak přesně cesta podle tohoto řetězce vypadá.

(a) Prvních pět kroků cesty podle řetězce *VLRRV*; červená šipka značí celkové posunutí.

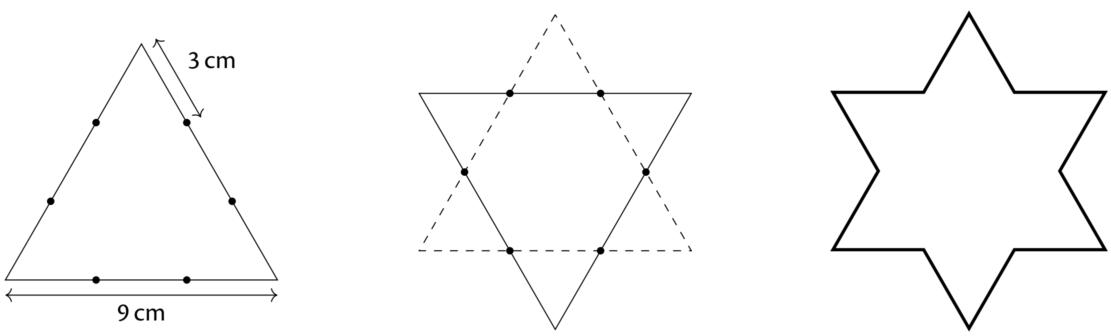
(b) Návrat do počátečního bodu

Obrázek 2

Nyní se podívejme na to, jak se Martin otočí při cestě podle řetězce, který byl zadán v úloze. Protože nás zajímá jenom otáčení, můžeme zcela zapomenout na znaky *V*. Potom se v řetězci střídají *R* a *L*, kterých je v řetězci dohromady 2019. To je liché číslo, takže když řetězec začíná (odmyslíme-li si všechna *V*) znakem *R*, musí jím i končit. Znaků *R* je tedy v řetězci o jeden více, než znaků *L*. Martin se tedy jeho projitím otočí o 90° doprava. Stejně jako v příkladu výše se tedy po čtyřech projitích celého řetězce musí vrátit na místo, z něhož vyšel. (Tato úvaha říká pouze to, že se Martin vrátí na začátek *nejpozději* po čtyřech projitích řetězce. Pro tento konkrétní řetězec se na začátek nevrátí nikdy dříve, ale pro nějaký jiný, např. *RVR* nebo *RRRRVVVVLLL* by se tak mohlo stát už někdy dříve).

Úloha 5A Vánoční hvězda

Můžeme si představit, že hvězda je složená ze dvou rovnostranných trojúhelníků o straně 9 cm položených přes sebe. Narýsujeme polopřímku a třikrát na ni kružítkem naneseme tři centimetry. Získáme úsečku délky devět centimetrů, pak už můžeme narýsovat první z rovnostranných trojúhelníků. Do kružítka si opět nabereme délku 3 cm a z každého vrcholu ji přeneseme na přilehlé strany trojúhelníka. Teď můžeme vést přímky vždy v každých dvou bodech vzdálených 3 cm od každého vrcholu, a tím vytvoříme druhý rovnostranný trojúhelník, a tedy i šesticípou hvězdu (viz obrázek 3). (Toto je pouze jeden z mnoha možných postupů, k cíli vede i spousta dalších.)



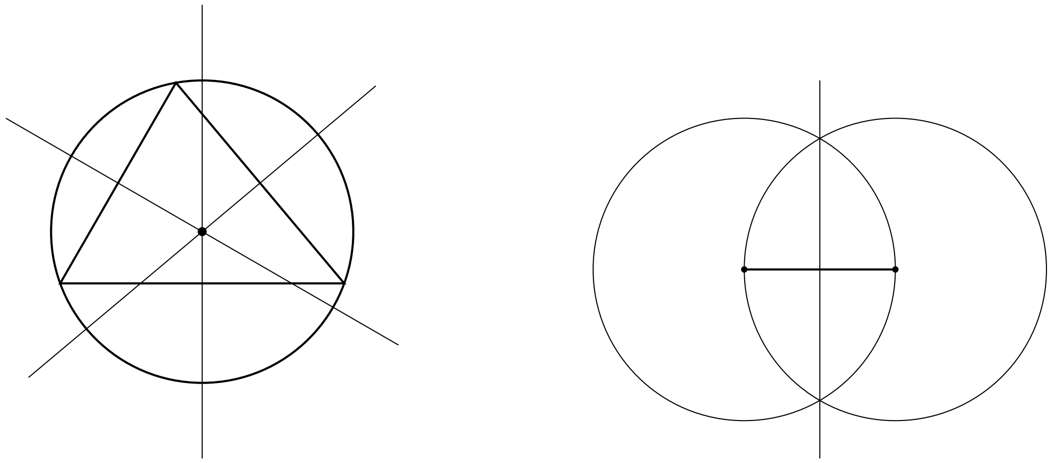
(a) (b) (c)

Obrázek 3: Konstrukce hvězdy

Kategorie starší

Úloha 1B Střed kružnice

Kružnici, kterou má Matěj nakreslenou, nazývejme k. Využijeme toho, že střed opsané kružnice trojúhelníka lze sestrojit jako průsečík os jeho stran (viz obrázek 4a). Osu úsečky přitom umíme zkonstruovat tak, že narýsujeme dvě kružnice vždy se středem v jednom krajním bodě a procházející druhým (viz obrázek 4b).



(a) Konstrukce středu opsané kružnice tojúhelníku

(b) Konstrukce osy úsečky

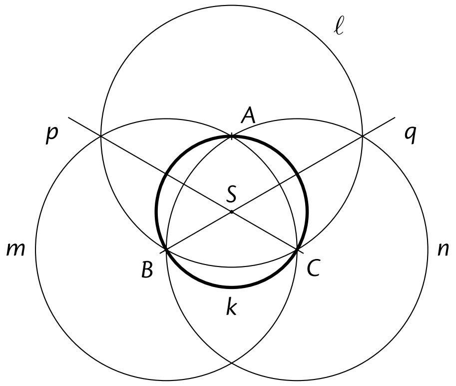
Obrázek 4

Nyní by tedy stačilo si zvolit nějaký trojúhelník s vrcholy ležícími na kružnici a sestrojit střed jeho kružnice opsané. Samotné strany trojúhelníku k ničemu nepotřebujeme (jejich osy dokážeme sestrojit i bez nich, stačí nám jejich krajní body), takže si vystačíme jen s dvěma přímkami – stačí zkonstruovat průsečík dvou os stran, není třeba rýsovat i třetí. Jenže takto by nám nestačily tři kružnice, neboť k sestrojení jedné osy strany bychom potřebovali kružnice dvě. Opatříme se tak, aby dvě z těchto čtyř kružnic, které v konstrukci narýsujeme, byly totožné. To se stane, pokud bude trojúhelník, který zvolíme, rovnoramenný. (Úlohu lze vyřešit i s obecným trojúhelníkem – pak bychom prostě použili tři kružnice se středy v jeho vrcholech se stejným poloměrem takovým, aby se tyto kružnice navzájem protínali a bylo možno narýsovat osy příslušných stran. Použitím rovnoramenného trojúhelníku si ale konstrukci zjednodušíme.)

Matějovi tedy lze doporučit následující postup (viz obrázek 5):

1. Zvolit libovolné dva různé body A,B, které leží na nakreslené kružnici.
2. Narýsovat kružnici ℓ, která má střed v bodě A a prochází bodem B.
3. Druhý průsečík kružnice ℓ s k (ten, který není B) označit C.
4. Narýsovat kružnici m, která má střed v B a prochází bodem A, a kružnici n, která má střed v C a prochází bodem A.
5. Narýsovat přímku p, která prochází oběma průsečíky kružnic ℓ a m, a přímku q, která prochází oběma průsečíky kružnic ℓ a n.
6. Průsečík přímek p a q označit S.

Bod S je pak středem kružnice k.



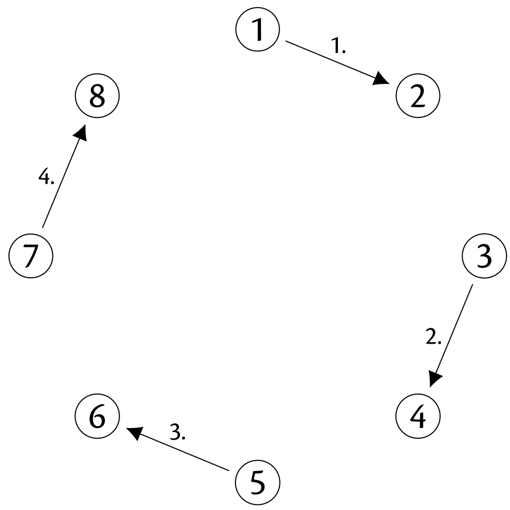
Obrázek 5: Konstrukce středu kružnice

Úloha 2B Podivná vybíjená

Musíme si ujasnit, jak vlastně popsaná vybíjená funguje. Zvířátko na pozici x svým hodem vybije zvířátko na pozici $x+1$, načež míč dostane zvířátko na pozici $x+2$, a toto se bude opakovat dál, až míč obejde celé kolečko. Během prvního „oběhu“ míče kolem dokola tedy bude vybito každé druhé zvířátko. Navíc pokud by byl zvířátek sudý počet, vrátil by se po jednom oběhu míč do rukou zvířátka na pozici 1 (viz obrázek 6). Když bude dokonce počet zvířátek mocnina dvou, neboli nějaké číslo, které se dá zapsat jako

$$2^n = \underbrace{2 \cdot 2 \cdot \dots \cdot 2 \cdot 2}_{n\text{-krát}}$$

jako třeba 2, 4, 8, 16, 32,… , tak si můžeme rozmyslet následující: hra začne tak, že zvířátko na pozici 1 vybije zvířátko na pozici 2 a postupně budou vybita všechna zvířátka na sudých pozicích (2, 4,…).



Obrázek 6: Během jednoho oběhu míče bude vybito každé druhé zvířátko – příklad pro hru osmi zvířátek. Očíslované šipky značí postupné vybíjení se.

Míč se vrátí do rukou zvířátka na pozici 1 a zvířátek bude přesně poloviční počet. Polovina z mocniny dvou je ale pořád mocnina dvou: $\frac{32}{2} = 16$, $\frac{16}{2} = 8$, $\frac{8}{2} = 4$ atd. Takže můžeme stejnou myšlenku použít znovu – míč jednou oběhne kolečko, vybije polovinu zvířátek a vrátí se do rukou zvířátka na pozici 1. Toto se bude opakovat, až jednou budou v kroužku jen dvě zvířátka, to na pozici 1 vybije svého posledního protivníka a vyhraje.

Dosavadní myšlenku můžeme shrnout takto: pokud zbude v některém okamžiku hry v kroužku takový počet zvířátek, který je mocninou dvou, tak už určitě vyhraje to zvířátko, které má právě míč v ruce (tj. chystá se vybít souseda po své levici). Jenže 65 je rovno $64 + 1$ a 64 se rovná

$$2^6 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2.$$

Pokud tedy vybíjená začne s 65 zvířátky, bude se vyvíjet takhle: zvířátko na pozici 1 vybije zvířátko 2 a míč do ruky dostane zvířátko číslo 3. V tu chvíli bylo jedno zvířátko vybito, takže jich v kroužku zbylo 64. To je mocnina dvou, takže už určitě vyhraje to zvířátko, které má v tu chvíli míč v ruce, což je to zvířátko, které začalo na pozici 3.

Úloha 3B Ciferný součet

Nejprve předpokládejme, že číslo n^2 má ciferný součet rovný 3. Ze školy víme, že ciferný součet čísla, které je násobkem tří, je násobek tří, a naopak, pokud je ciferný součet čísla dělitelný třemi, pak je i samo číslo násobkem tří (pokud ti vrtá hlavou, proč tomu tak je, stačí se dočíst na konec řešení této úlohy). Číslo n^2 lze tedy zapsat jako $3k$, kde k je nějaké přirozené číslo. Jenže zároveň víme, že n^2 lze rozložit na součin dvou celých čísel jako $n \cdot n$. Platí tedy, že tři dělí $n \cdot n$, a jelikož trojka je prvočíslo, plyne z tohoto, že musí dělit i n (pokud by n nebylo násobkem tří, pak by jím nemohlo být ani $n \cdot n$).

Dostáváme tedy, že n je dělitelné třemi, takže $n^2 = n \cdot n$ musí být dělitelné dokonce devíti. Jenže pro dělitelnost devíti platí podobné pravidlo jako pro dělitelnost třemi – přirozené číslo je dělitelné devíti, právě pokud je devíti dělitelný jeho ciferný součet. Můžeme tedy usoudit, že ciferný součet n^2 má být násobkem devíti. To nás zahání do úzkých, neboť cifry n^2 mají mít (z předpokladu) součet 3, což násobek devíti není. Proto n^2 nemůže mít ciferný součet rovný 3.

Na závěr si ještě v rychlosti ukažme, že číslo $m = 3k$ (resp. $m = 9k$), tj. libovolný násobek tří (resp. devíti) má opravdu součet cifer dělitelný třemi (resp. devíti), a naopak. Zápis přirozeného čísla v desítkové soustavě není nic jiného než součet několika jednotek, několika desítek, několika stovek a tak dále – to lze zapsat jako $m = a + 10b + 100c + \dots$, konkrétně třeba $2019 = 9 \cdot 1 + 1 \cdot 10 + 0 \cdot 100 + 2 \cdot 1000$. Ciferný součet čísla m pojmenujme x. Ten bude roven $a + b + c + \dots$. Pokud tedy od m odečteme x a tento rozdíl nazveme y, dostaneme

$$y = (a + 10b + 100c + \dots) - (a + b + c + \dots) = a(1 - 1) + b(10 - 1) + c(100 - 1) + \cdots = 0 \cdot a + 9b + 99c + \dots,$$

kde v každém dalším členu součtu bude vystupovat číslo tvořené samými devítkami. Potom ale musí y být vždy násobkem devíti, a tedy i tří (pro každé přirozené číslo m). Z toho konečně můžeme usoudit: pokud je m násobkem tří (resp. devíti), pak můžeme vyjádřit $x = m - y$, takže x musí být jakožto rozdíl dvou násobků tří (devíti) taktéž násobkem tří (devíti). Naopak, je-li x násobkem tří (devíti), musí být násobkem tří (devíti) i $m = x + y$ – součet dvou násobků tří (devíti) je opět násobkem tří (devíti).

Úloha 4B Heslo

Zamysleme se nejprve nad tím, jak zjistit, jaká je jedna konkrétní, řekněme `x`-tá, cifra správného hesla, tedy zdali pro otevření trezoru má odpovídající žárovka svítit, či býti zhasnuta. Toho lze docílit následovně: nejprve funkcí `vyzkousej()` získáme momentální počet správně zadaných cifer, žárovku přepneme (tj. rozsvítíme, pokud byla zhasnutá, anebo zhasneme, pokud svítila) funkcí `prepni(x)`, znovu zavoláme `vyzkousej()` a hodnoty porovnáme. Pokud se přepnutím žárovky počet správně zadaných cifer hesla zvýšil, pak musí tím správným být nový stav žárovky (po přepnutí). Pokud se naopak snížil, byl správným stavem ten původní – měli bychom tedy následně tuto `x`-tou žárovku přepnout nazpět. (Nemůže se stát, že by počet správných cifer zůstal nezměněn – vždy buďto jednu správnou přepneme na špatnou, nebo naopak jednu špatnou na správnou, zatímco zbytek zůstane stejný.) Takovýto proces zjištění (a zároveň nastavení) správné hodnoty `x`-té cifry hesla lze v jazyce Múzy 2.0 zapsat následovně:

```
pocetspravnych ← vyzkousej()
prepni(x)
novypocetspravnych ← vyzkousej()
Jestlize(novypocetspravnych < pocetspravnych) pak (
    prepni(x)
)
```

Takovýto krátký program vykoná (řádek po řádku) následující: Nejprve se do proměnné `pocetspravnych` uloží hodnota, kterou vrátí funkce `vyzkousej()`, představující počet cifer, které jsou právě zadány správně. Následně se přepne `x`-tá žárovka a do proměnné `novypocetspravnych` se uloží počet správných cifer po přepnutí žárovky. Poté dojde k porovnání – pokud je hodnota v proměnné `novypocetspravnych` menší než ta v proměnné `pocetspravnych`, tedy pokud jsme přepnutím `x`-té žárovky přepsali správnou cifru na špatnou, pak bude žárovka přepnuta nazpět, čímž už bude `x`-tá cifra určitě do trezoru zadána správně. Pokud je naopak hodnota v `novypocetspravnych` větší než ta v `pocetspravnych`, pak není třeba dělat nic, neboť `x`-tá žárovka už je ve správném stavu.

Nyní bychom mohli toto zopakovat osmkrát, pro každou cifru, resp. žárovku jednou. K tomu by stačilo prostě uložit do proměnné `x` jedničku a následně vždy provést příkazy zapsané v programu výše a zvýšit `x` o jedna, dokud hodnota této proměnné nepřeroste číslo 8. To zapíšeme následnovně:

```
x ← 1
Dokud(x ≤ 8) opakuj (
    pocetspravnych ← vyzkousej()
    prepni(x)
    novypocetspravnych ← vyzkousej()
    Jestlize(novypocetspravnych < pocetspravnych) pak (
        prepni(x)
    )
    x ← x + 1
)
```

V tomto programu ale pro zjištění jedné cifry voláme funkci `vyzkousej()` dvakrát, takže na zdárné určení celého hesla potřebujeme tuto funkci volat hned šestnáctkrát. Podle zadání má ale stačit nanejvýš deset volání funkce `vyzkousej()`. K opravení tohoto neduhu ale stačí jen malá změna.

Zamysleme se nad tím, co by se dělo, pokud bychom prostě výše napsaný program pro správné nastavení `x`-té cifry použili opakovaně několikrát za sebou na jednotlivé cifry, jak jsme zamýšleli. Pro každou cifru znovu voláme funkci `vyzkousej()`, abychom vrácenou hodnotu uložili do `pocetspravnych`. Počet v tu chvíli správně zadaných cifer bychom ale snadno mohli získat už z nastavování předchozí cifry – je jím buďto hodnota proměnné `pocetspravnych`, nebo `novypocetspravnych` podle toho, zdali jsme pro tuto předchozí cifru prvním přepnutím žárovky počet správně zadaných cifer snížili, nebo zvýšili. Pokud bychom program takto upravili, stačilo by nám funkci `vyzkousej()` volat jednou na začátku programu a pak už jen jednou u zjišťování každé cifry, tedy celkem pouze devětkrát. S tímto vylepšením už lze celkový program pro nastavení správného hesla zapsat následovně:

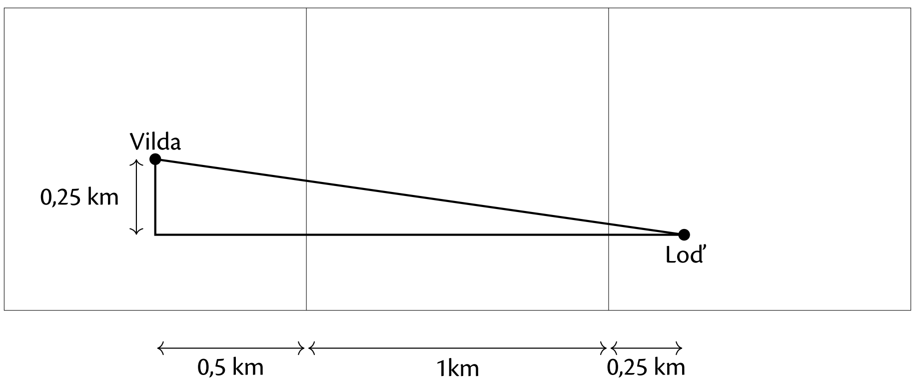
```
pocetspravnych ← vyzkousej()
x ← 1
Dokud(x ≤ 8) opakuj (
    prepni(x)
    novypocetspravnych ← vyzkousej()
    Jestlize(novypocetspravnych < pocetspravnych) pak (
```

```
        prepni(x)
    )
    Jinak (
        pocetspravnych ← novypocetspravnych
    )
    x ← x + 1
)
```

Zde nejprve do `pocetspravnych` uložíme zavoláním funkce `vyzkousej()` počet cifer, které jsou správně zadány, ještě než začneme cokoliv přepínat. Následně postupně pro každou cifru zkusíme přepnout žárovku, a nový počet správně zadaných cifer uložíme do `novypocetspravnych`. Pokud jsme přepnutím žárovky snížili počet správně zadaných cifer, pak tuto žárovku přepneme nazpět a počet správně zadaných cifer bude odpovídat tomu, jež byl před přepínáním této žárovky uložen v `pocetspravnych` (neboť jsme počet správně zadaných cifer vlastně ponechali beze změny). Pokud jsme ale počet správně zadaných cifer přepnutím zvýšili, pak není důvod žárovku přepínat nazpět – naopak hodnotu z proměnné `novypocetspravnych` uložíme do `pocetspravnych`, aby v této proměnné nadále (pro použití při zjišťování další cifry) byl uložen počet cifer, které jsou právě zadány správně. Takto postupně nastavíme každou cifru hesla správně a po nastavení všech cifer bude správně celé heslo.

Úloha 5B Hranatá planetka

Nejprve potřebujeme zjistit, kudy povede Vildova cesta. Víme, že nejrychleji to půjde po přeponě pravoúhlého trojúhelníku. Kde se u nás nachází? Planetku si můžeme rozložit do sítě, aby se nám s ní lépe pracovalo. Nejkratší bude ta cesta, kde půjdeme přes co nejméně stěn (v našem případě tři, jelikož Vilda je na protější stěně vůči lodi) a kde zároveň dojdeme k cíli z hrany nejblíže lodi. Planetku si tedy můžeme zjednodušit takto:



Obrázek 7: Planetka

Víme, že Vilda je uprostřed stěny, tudíž je půl kilometru od každé hrany a že loď je ve čtvrtině a tudíž je 0,25 km vzdálená od nejbližších hran. Teď máme rozměry našeho pravoúhlého trojúhelníku a můžeme pomocí Pythagorovy věty ($c^2 = a^2 + b^2$) dopočítat délku Vildovy trasy: trasa $= \sqrt{1{,}75^2 + 0{,}25^2}$km, tedy trasa $\doteq$ 1,77 km. Vildova trasa k lodi je tedy 1,77 km dlouhá.