

---

# Zadání soutěžních úloh

## Kategorie programování žáci a mládež

20. až 22. dubna 2017

Soutěž v programování – 31. ročník

Krajské kolo 2016/2017

Úlohy můžete řešit v libovolném pořadí a samozřejmě je nemusíte vyřešit všechny. Za každou úlohu můžete dostat maximálně 10 bodů, z nichž je většinou 9 bodů vyhrazeno na ohodnocení funkčnosti programu, jeho shody se zadáním a efektivity a jeden bod na dokumentaci a přehlednost zdrojového kódu (vhodné členění zdrojového kódu, vhodně zvolené názvy identifikátorů, komentáře na místech, kde je to potřeba, atd.). Body získané za každou úlohu se ještě násobí koeficientem, který odráží složitost úlohy.

Na řešení úloh máte 4 hodiny čistého času.

Před zahájením soutěže vám pořadatel oznámí, kde najdete testovací soubory a kam máte ukládat řešení úloh.

# Prasátko

## Koeficient 1

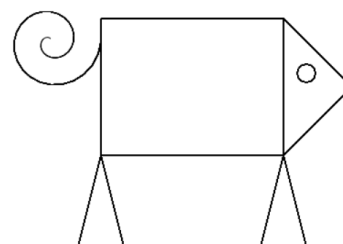
Prohlédněte si prosím animaci v příloženém souboru `prase.html`. Vaším úkolem je napsat program, který zobrazí obdobný výsledek. Nejvíce bodů získáte, pokud provedete plynulou animaci vykreslování čar (za stejný čas se vždy vykreslí stejná délka čáry). Pozornost věnujte i postupnému vykreslení oka a ocásku, u nich se hodnotí kulatost jak oka, tak ocásku a u ocásku také shoda s předlohou (správný směr zakroucení, stoupání spirály, počet otáček).

Pokud je celé zadání nad vaše síly, můžete zpracovat jednodušší variantu – animovat postupně po automaticky spouštěných krocích, kdy krokem je zobrazení jedné celé čáry, a pak celého oka a celého ocásku. Několik málo bodů dostanete i v případě, že program celé prasátko nakreslí najednou.

Body dostanete i za to, že v průběhu animace okno vašeho programu „nevytuhne“ a lze s ním např. pohybovat.

Je zakázáno využívat zdrojový kód vzorového řešení a obrázek prasátka ze zadání. Nebudou také hodnoceny programy zobrazující předpřipravené obrázkové soubory.

Pokud budete mít problém se zobrazením vzorové animace, můžete vyjít z příloženého obrázku a následujícího popisu: Animace začíná vykreslením čáry za hlavou prasátka shora dolů, následuje spodní hrana těla, zadní hrana a pak hrana horní. Následně dvě čáry hlavy od té horní (začíná shora), přední nohy, zadní nohy, oko a nakonec ocásek. Nohy a ocásek se vždy začínají kreslit na straně blíže k tělu. Pokud na sebe dvě čáry navazují (tělo, hlava), začíná se ta druhá kreslit ve směru od té první.



# Dvanáct set troska

Koeficient 3

Zdělili jste po svém pradědečkovi luxusní automobil Škoda 1203. Poté, co jste spatřili své dědictví, které jste ušlechtili nazvali *dvanáct set troska*, rozhodli jste se ho odvézt zhodnotit na skládku. Po cestě se stala ale nemilá věc – zasekl se vám volant a nejde otočit doleva, můžete tedy na křižovatkách jezdit jen rovně nebo zatáčet doprava! Vaším úkolem je s takto poškozeným vozem najít co nejkratší cestu na skládku, aby se vám náhodou po cestě nerozbilo ještě něco.

Napište program, který na vstupu dostane mapu města, výchozí pozici dvanáct set trosky a pozici skládky, a najde nejkratší cestu dvanáct set troskou na skládku tak, aby na každé křižovatce jela buď rovně nebo zahla doprava. Vypište délku nejkratší cesty a *interaktivně* nalezenou cestu zobrazte.

## Vstup

Vstup načtete ze souboru `mesto.txt`. Tento soubor obsahuje mapu města, což je čtverečková síť o rozměrech 100×100 políček. Mapa města je uložena na 100 řádcích s tím, že každý řádek je ukončený znaky CR a LF a obsahuje (kromě znaků ukončení řádku) přesně 100 znaků – jednotlivá políčka tohoto řádku mapy města. Znaky v souboru mohou být následovné:

- . silnice (průjezdné pole)
- X budova (nepřůjezdné pole)
- @ skládka
- % dvanáct set troska

Můžete předpokládat, že soubor je vždy v pořádku, obsahuje právě jednu skládku a právě jednu dvanáct set trosku. Navíc počáteční pozice dvanáct set trosky sousedí s právě jednou silnicí – předpokládejte tedy, že dvanáct set troska je na začátku natočena směrem k této silnici.

## Výstup

Pro daný soubor najděte nejkratší cestu dvanáct set trosky na skládku tak, aby dvanáct set troska jela jen po silnici a zatáčela jen doprava (tj. pokud z nějakého směru přijede na políčko silnice, může buď pokračovat rovně, nebo zatočit o 90° doprava). Můžete předpokládat, že taková cesta vždy existuje. Pokud jich existuje více, najděte libovolnou z nich.

Až cestu naleznete, vypište její délku – počet políček, které na cestě navštívíte, včetně políčka skládky. Pokud nějaké políčko navštívíte vícekrát, počítejte ho vícekrát. Poté nalezenou cestu interaktivně zobrazte – umožněte jednak vykreslení kompletní cesty, a jednak krokování jednotlivých políček cesty.

## Příklad

Pro následující mapu města (pro názornost je zobrazený jen její levý horní roh o velikosti 7×7, všechna ostatní nezobrazená políčka jsou X):

```
XXXXXXXXX
.....
.X.X.X.
.....
.X.X.X.
.X@X%X.
XXXXXXXXX
```

je nejkratší cesta na skládku dlouhá 22 políček. Kdyby bylo možné zatáčet doleva, byla by nejkratší cesta dlouhá 6 políček. Nejkratší správná cesta je zobrazena na následující mapě:

```
XXXXXXXXX
|   |   | |
| X | X | X |
|   |   |
|.X | X | X.|
|.X@X%X.
XXXXXXXXX
```

# Simulátor Enigma

## Koeficient 2

Enigma byl šifrovací přístroj, používaný za druhé světové války. Tento šifrovací stroj uměl šifrovat vstupní text pomocí šifrování jednotlivých znaků vstupního textu na jiné znaky výstupního textu.

Celkový proces šifrování strojem Enigma spočíval v postupném šifrování jednoho znaku průchodem přes několik šifrovacích tabulek, což mohlo vypadat např. následovně:

E	→	B	→	I	→	S	→	M	→	K	→	J	→	T
N	→	P	→	U	→	J	→	R	→	Q	→	R	→	X
I	→	E	→	J	→	B	→	O	→	X	→	L	→	W
G	→	C	→	P	→	D	→	Z	→	G	→	M	→	Y
M	→	F	→	Y	→	P	→	G	→	D	→	X	→	S
A	→	G	→	L	→	W	→	C	→	E	→	K	→	V

Celkově ENIGMA → TXWYSV

První sloupec → poslední sloupec

Vaším úkolem je napsat program, který bude simulovat činnost šifrovacího stroje Enigma.

Šifrují se pouze velká písmena anglické abecedy.

Máme několik šifrovacích tabulek ve tvaru 26 dvojic velkých písmen anglické abecedy A až Z oddělených čárkou („bílé“ znaky ignorujte). Každé písmeno abecedy je v tabulce obsaženo právě dvakrát. Jednou je ve dvojici vlevo, jednou vpravo. Například:

EK,	JF,	TE,	DA,	BX,	LO,	AP,	CJ,	KG,	SL,	RI,	UZ,	QD,
OB,	IV,	XQ,	GN,	NT,	HU,	PR,	YW,	MC,	VM,	ZY,	WS,	FH

Při šifrování se znak A změní na P, B na X atd., až Z na Y.

Vstupem programu jsou textové soubory, jednotlivé řádky jsou odděleny znaky CR LF, vstupní kódování je ASCII. Výstupem programu je výpis zašifrovaného textu na obrazovku.

Postupně v programu naimplementujte následující funkce:

- Uživatel si vybere vstupní soubor pro šifrování.
- Načtení vstupního souboru a nastavení šifrování pro tento běh programu z prvního řádku.
- Vypsání nastavení šifrování.
- Vypisování nezašifrovaného a zašifrovaného textu ze vstupního souboru po jednotlivých řádkách.

První řádka vstupního souboru obsahuje jména použitých šifrovacích tabulek. Jména jsou oddělena čárkou, bílé znaky se ignorují.

Například při nastavení ve tvaru

```
I.A, II.A, III.A, P1.AB, III.B, II.B, I.B
```

budou pro šifrování postupně použity šifrovací tabulky ze souborů I.A, II.A, ... a jako poslední ze souboru I.B.

Další řádky vstupního souboru obsahují text zprávy určený k zašifrování psaný velkými písmeny anglické abecedy. Mezery se přenášejí beze změny.

Po načtení nastavení šifrování program toto nastavení (obsah první řádky vstupního souboru) vypíše. Následovně bude vstupní soubor po řádkách šifrovat. Pro každou řádku vstupního souboru program vypíše dvě řádky – první řádka bude obsahovat originální text a druhá zašifrovaný text. Nezašifrovaný řádek začíná textem VSTUP= a zašifrovaný textem VYSTUP=.

V ukázkovém adresáři máte kromě šifrovacích tabulek i několik vzorových datových souborů VZORx.DAT.

Například pro soubor VZOR6.DAT:

```
I.A, UKWB.A, I.B  
SKAKAL PES PRES OVES  
IMNMNW RHI RPHI FJHI  
RPHI YHWHAFB WFBMB
```

by měl výstup programu vypadat takto:

```
I.A, UKWB.A, I.B  
VSTUP=SKAKAL PES PRES OVES  
VYSTUP=IMNMNW RHI RPHI FJHI  
VSTUP=IMNMNW RHI RPHI FJHI  
VYSTUP=SKAKAL PES PRES OVES  
VSTUP=RPHI YHWHAFB WFBMB  
VYSTUP=PRES ZELENOU LOUKU
```