

Soutěž dětí a mládeže v programování  
Obvodní kolo Prahy 1 a 2  
rok 2013

---

Seznam úloh (v závorce uvedena koeficient obtížnosti).  
Řešení ukládejte do složky C:\Prijmeni\Nazev\_ulohy.

### **Úloha 1: graf (1,0)**

Vytvořte obrázek - graf ze vstupního souboru.

Na ose X je pořadí jednotlivých hodnot (v rozsahu 0..[počet řádek - 1]), na osu Y pak vynášíte jednotlivé hodnoty (v rozsahu [minimální hodnota]..[maximální hodnota] - případně s nějakým odsazením; rozhodně však dynamicky!).

- Graf by měl mít velikost v px nezávislou na vstupních datech - například 1200x900
- Označte osy rovnoměrně rozmístěnými popisky s hodnotami.
- Vyznačte osu x (hodnota y=0) jako tenkou přímku.
- Body na grafu můžete dle libosti vyznačit například křížkem, kruhem nebo obdelníkem od osy x (=sloupcový graf)
- **Použijte knihovny pro práci s grafikou (kreslení do png obrázku), NEPOUŽÍVEJTE knihovny pro kreslení celých grafů.**

#### **Příklad vstupu - datový soubor:**

```
0.4  
1.3  
-5  
4  
3.3  
-2.6  
0  
-3
```

#### **Příklad výstupu - graf**

Výstupní grafy v příkladech jsou pouze orientační.

---

### **Úloha 2: příšera v bludišti (1,0)**

V datovém souboru na vstupu máte bludiště; na prvním řádku jsou dvě přirozená čísla - výška (počet řádků) a šířka (počet znaků na řádku).

Od druhého řádku následuje vykreslení bludiště - zadané jako posloupnost znaků 'X' reprezentující stěnu, mezera ' ' reprezentující volné políčko a znak 'A' znamenající příšeru. Příšera je v bludišti pouze jedna.

Druhý soubor je posloupnost znaků reprezentující pohyb příšery

- 'n' - north=sever=nahoru

- 's' - south=jih=dolů
- 'e' - east=východ=doprava
- 'w' - west=západ=doleva

Úkolem je vypočítat, kde se bude příšera nacházet po těchto krocích a vykreslit znovu bludiště s příšerou na daném políčku.

Pokud jde v kroku příšera "hlavou proti zdi", zůstane na svém původním místě.

Počítejte s tím, že bludiště může být i velmi rozlehlé (třeba 1000x1000 dílků)

Detekujte případný únik příšery z bludiště a program v té chvíli ukončete s hlášením "Prisera unikla".

Není potřeba ošetřovat vstup - předpokládáte, že daný datové soubory mají zadaný formát.

### Příklad vstupu - *bludiste.txt*

```
5 7
XXXXXXX
X A  XX
XX   X
XXX  X
XXXXXX
```

### Příklad vstupu - *kroky.txt*

```
eesesssw
```

### Výstup

```
XXXXXXX
X   XX
XX  X
XXX A X
XXXXXXX
```

## Úloha 3: placení mincemi (2)

Jsme v zemi, ve které se používají mince v hodnotě 1, 15, 18 a 49. Vytvořte program, který dostane jako parametr (či v souboru) přirozené číslo a jako výstup dá číslo, kolik **nejméně** potřebujeme mincí k zaplacení dané částky.

Pozor: uvědomte si, že *hladový algoritmus* fungující u nás (s mincemi 1,2,5,10,20,50) jdoucí od nejvyšší hodnoty mince "dokud to jde", není v této zemi správný!

Například: částku 103 byste tímto algoritmem zaplatili jako 49+49+1+1+1+1+1 (7mincí), ale správné řešení je 49+18+18+18 (4mince)

**Více bodů** dostanete, zobecníte-li program a hodnoty mincí v oné zemi budete načítat z textového souboru.

**Více bodů** dostanete za efektivnější (rychleji počítající) algoritmus. Částka kterou chcete rozdělit může být v řádech milionů.

Test pro hodnoty mincí 1,15,18,49:

Vstup (částka)	Výstup (nejmenší počet mincí k zaplacení)
30	2
103	4
202	7
987	24
10000	208
14010	296
140103	2863
9531602	194526