

Úkolem je vytvořit program, který bude zobrazovat vybrané citáty.

Vstupem programu je jedno celé číslo. Toto číslo je z rozsahu 0 až 8 a určuje citát, který má být zobrazen.

Výstupem programu je citát odpovídající zadanému číslu na vstupu. Citáty odpovídající jednotlivým číslům jsou uvedené v ukázkových bězích programu níže. Pozor, za textem citátu je odřádkování (`\n`).

Pokud je vstup neplatný, program to musí detekovat a zobrazit chybové hlášení. Chybové hlášení zobrazujte na standardní výstup (ne na chybový výstup). Za chybu považujte:

- ze vstupu nelze přečíst celé číslo,
- číslo přečtené ze vstupu je mimo interval 0 až 8.

Povšimněte si, že program zobrazuje různá chybová hlášení. První chybové hlášení se uplatní, pokud program nemůže ze vstupu přečíst platné celé číslo. Druhé chybové hlášení se pak použije, pokud je načtené číslo mimo rozsah (viz ukázka).

Při programování si dejte pozor na přesnou podobu výpisů. Výstup Vašeho programu kontroluje stroj, který požaduje přesnou shodu výstupů Vašeho programu s výstupy referenčními. Za chybu je považováno, pokud se výpis liší. I chybějící nebo přebývající mezera/odřádkování je považováno za chybu. Abyste tyto problémy rychle vyloučili, použijte příložený archiv se sadou vstupních a očekávaných výstupních dat. Podívejte se na videotutoriál (courses → výuková videa), jak testovací data použít a jak testování zautomatizovat.

Váš program bude spouštěn v omezeném testovacím prostředí. Je omezen dobou běhu (limit je vidět v logu referenčního řešení) a dále je omezena i velikost dostupné paměti (ale tato úloha by ani s jedním omezením neměla mít problém). Testovací prostředí dále zakazuje používat některé "nebezpečné funkce" -- funkce pro spouštění programu, pro práci se sítí, ... Pokud jsou tyto funkce použité, program se nespustí. Možná ve svém programu používáte volání:

```
int main ( int argc, char * argv [] )
{
    ...

    system ( "pause" ); /* aby se nezavrelo okno programu */
    return 0;
}
```

Toto nebude v testovacím prostředí fungovat – je zakázáno spouštění jiného programu. (I pokud by se program spustil, byl by odmítnut. Nebyl by totiž nikdo, kdo by pauzu "odmáčkl", program by čekal věčně a překročil by tak maximální dobu běhu.) Pokud tedy chcete zachovat pauzu pro testování na Vašem počítači a zároveň chcete mít jistotu, že program poběží správně, použijte následující trik:

```
int main ( int argc, char * argv [] )
{
    ...

#ifdef __PROGTEST__
    system ( "pause" ); /* toto progtest "nevidí" */
#endif /* __PROGTEST__ */
    return 0;
}
```

**Ukázka práce programu:**

---

mĹ' nob:

0

Qapla'

noH QapmeH wo' Qaw'lu'chugh yay chavbe'lu' 'ej wo' choqmeH may' DoHlu'chugh Ĺujbe'lu'.

---

mĹ' nob:

1

Qapla'

bortaS bIr jablu'DI' reH QaQqu' nay'.

---

mĹ' nob:

2

Qapla'

Qu' buSHa'chugh SuvvI', batlhHa' vangchugh, qoj matlhHa'chugh, pagh ghaH SuvvI''e'.

---

mĹ' nob:

3

Qapla'

bISeH'eghlaH'be'chugh Ĺatlh Dara'laH'be'.

---

mĹ' nob:

4

Qapla'

qaStaHvIS wa' ram loS SaD Hugh SIjlaH qetbogh loD.

---

mĹ' nob:

5

Qapla'

Suvlu'taHvIS yapbe' HoS neH.

---

mĹ' nob:

6

Qapla'

Ha'DIbaH DaSop 'e' DaHechbe'chugh yIHohQo'.

---

mĹ' nob:

007

Qapla'

Heghlu'meH QaQ jajvam.

---

mĹ' nob:

8

Qapla'

leghlaHchu'be'chugh mIn lo'laHbe' taj jej.

---

mĹ' nob:

29

Qih mi' 29

---

mĹ' nob:

-5

Qih mi' -5

---

ml' nob:  
abc  
Neh mi'

---

#### Poznámky:

- Ukázkové běhy zachycují očekávané výpisy Vašeho programu (tučné písmo) a vstupy zadané uživatelem (základní písmo). Zvýraznění tučným písmem je použité pouze zde na stránce zadání, aby byl výpis lépe čitelný. Váš program má za úkol pouze zobrazit text bez zvýrazňování (bez HTML markupu).
- Znak odřádkování (\n) je i za poslední řádkou výstupu (i za případným chybovým hlášením).
- Výzvy programu a citáty jsou úmyslně zvoleny tak, aby bylo obtížné je ručně opisovat. Úloha se Vás snaží mj. naučit efektivní práci s počítačem, tedy např. s funkcemi copy & paste (které bohužel někteří studenti neznají nebo neumějí použít). Vyzkoušejte si a naučte se copy & paste jak pod Windows tak pod UNIXem (Linuxem), u závěrečného testu se to bude hodit!
- Ruční kontrola shody Vašeho a referenčního výstupu je nudná, člověk při porovnávání často přehlédne drobný rozdíl. Obzvláště to platí u textu v úloze, který pro většinu lidí není srozumitelný. Toto je opět záměr. Využijte volna na začátku semestru a naučte se v této jednoduché úloze efektivně testovat Vaše programy. Podívejte se na zmíněný videotutoriál (courses → výuková videa) a využijte pro porovnání Váš počítač. V dalších úlohách tím ušetříte mnoho času.
- Vstup typu 1.23 případně labcd není správný. Jeho ošetření je ale pracnější. Program proto můžete realizovat v jednodušší podobě, která takový vstup akceptuje jako číslo 1. Takový program bude hodnocen nominálním počtem bodů (100 %). Pro zájemce je k dispozici bonusový test, který zkouší zadávat právě tento typ vstupů. Pokud správně ošetříte i tento typ chyb, bude váš program hodnocen body navíc. Ukázka práce programu pro tento typ vstupu:

ml' nob:  
1.23  
bIjatlh 'e' yImev