

Citáty

Termín odevzdání:	30.10.2016 23:59:59
Hodnocení:	1.1000
Max. hodnocení:	1.0000 (bez bonusů)
Odevzdaná řešení:	1 / 20 Volné pokusy + 10 Penalizované pokusy (-10 % penalizace za každé odevzdání)
Nápovědy:	0 / 2 Volné nápovědy + 2 Penalizované nápovědy (-10 % penalizace za každou nápovědu)

Úkolem je vytvořit program, který bude zobrazovat vybrané citáty.

Vstupem programu je jedno celé číslo. Toto číslo je z rozsahu 1 až 5 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 1 až 5.

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 přiložený 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:

```
m1' nob:
1
Qapla'
noH QapmeH wo' Qaw'lu'chugh yay chavbe'lu' 'ej wo' choqmeH may' DoHlu'chugh lujbe'lu'.
```

m1' nob:
2
Qapla'
Qu' buSHa'chugh SuvwI', batlhHa' vangchugh, qoj matlhHa'chugh, pagh ghaH SuvwI''e'.

m1' nob:
3
Qapla'
qaStaHvIS wa' ram loS SaD Hugh SIjlaH qetbogh loD.

m1' nob:
4
Qapla'
Ha'DIbaH DaSop 'e' DaHechbe'chugh yIHohQo'.

m1' nob:
5
Qapla'
leghlaHchu'be'chugh mIn lo'laHbe' taj jej.

m1' nob:
0
luj

m1' nob:
6
luj

m1' nob:
abc
luj

Poznámky:

- 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 1abcd považujte za správný. Základní použití funkce scanf bude takové vstupy interpretovat jako číslo (zde číslo 1) a zbytek vstupu nebude čist. Šlo by samozřejmě program rozšířit, aby takovéto vstupy odmítal, v této úloze to ale není potřeba dělat. Automatizované testování vstupy tohoto typu se nežadává.

Vzorová data:

Download

Referenční řešení

1 22.10.2016 23:33:00

Download

Stav odevzdání: Ohodnoceno

Hodnocení: 1.1000

- **Hodnotitel: automat**
 - Program zkompileován
 - Test 'Základní test s parametry podle ukázky': Úspěch
 - Dosaženo: 100.00 %, požadováno: 100.00 %

- Max doba běhu: 0.004 s (limit: 1.000 s)
- Celková doba běhu: 0.032 s
- Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test ošetření nesprávných vstupů': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Max doba běhu: 0.004 s (limit: 1.000 s)
 - Celková doba běhu: 0.044 s
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Celkové hodnocení: 100.00 % (= 1.00 * 1.00)
- Celkové procentní hodnocení: 100.00 %
- Bonus za včasné odevzdání: 0.10
- Celkem bodů: $1.00 * (1.00 + 0.10) = 1.10$

		Celkem	Průměr	Maximum	Jméno funkce
SW metriky:	Funkce:	1	--	--	--
	Řádek kódu:	29	29.00 ± 0.00	29	main(void)
	Cyklomatická složitost:	9	9.00 ± 0.00	9	main(void)