

Symetrická čísla #2

Termín odevzdání:	10.11.2019 23:59:59	1310102.424 sec
Pozdní odevzdání s penalizací:	06.01.2020 23:59:59 (Penále za pozdní odevzdání: 100.0000 %)	
Hodnocení:	0.0000	
Max. hodnocení:	5.0000 (bez bonusů)	
Odevzdaná řešení:	0 / 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 počítat čísla, která mají symetrický zápis v zadané číselné soustavě. Jedná se o rozšíření jednodušší úlohy, rozšíření spočívá v možnosti udát základ číselné soustavy. Doporučujeme nejprve řešit úlohu jednodušší a po jejím úspěšném odevzdání program rozšířit.

Vstupem programu je posloupnost příkazů k hledání. Zadávaní je ukončené dosažením konce vstupu (EOF). Každý příkaz se skládá ze čtveřice údajů X R L0 HI. První znak udává, zda chceme nalezená symetrická binární čísla všechna vypsat (na vstupu bude znak **l** - list) nebo zda je chceme pouze spočítat (na vstupu bude znak **c** - count). Za znakem následuje základ číselné soustavy (v této soustavě chceme hledat symetrická čísla) a konečně dolní a horní mez prohledávaného intervalu - čísla L0 a HI. Prohledává se uzavřený interval hodnot, tedy do prohledávání jsou zahrnuta i obě čísla L0 a HI.

Výstupem programu je odpověď na každý vstupní příkaz. Odpovědi na příkaz **l** je seznam nalezených čísel, formát je zřejmý z ukázky. Odpovědi na příkaz **c** je počet nalezených čísel.

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:

- neznámý příkaz (ani **l** ani **c**),
- nečíselné zadání základu soustavy, základ soustavy mimo interval <2;36>,
- nečíselné zadání intervalu nebo chybějící meze L0 a HI,
- dolní mez je záporná,
- dolní mez je větší než horní mez.

Ukázka práce programu:**Vstupní intervaly:**

```
l 2 0 20
0 = 0 (2)
1 = 1 (2)
3 = 11 (2)
5 = 101 (2)
7 = 111 (2)
9 = 1001 (2)
15 = 1111 (2)
17 = 10001 (2)
c 2 0 20
Celkem: 8
l 3 0 40
0 = 0 (3)
1 = 1 (3)
2 = 2 (3)
4 = 11 (3)
8 = 22 (3)
10 = 101 (3)
13 = 111 (3)
16 = 121 (3)
20 = 202 (3)
23 = 212 (3)
26 = 222 (3)
28 = 1001 (3)
40 = 1111 (3)
l 5 0 50
0 = 0 (5)
1 = 1 (5)
2 = 2 (5)
3 = 3 (5)
4 = 4 (5)
6 = 11 (5)
12 = 22 (5)
18 = 33 (5)
24 = 44 (5)
26 = 101 (5)
31 = 111 (5)
```

```

36 = 121 (5)
41 = 131 (5)
46 = 141 (5)
l 17 1918 2019
1927 = 6b6 (17)
1944 = 6c6 (17)
1961 = 6d6 (17)
1978 = 6e6 (17)
1995 = 6f6 (17)
2012 = 6g6 (17)
c 31 38 12345
Celkem: 396

```

Vstupni intervaly:
c 6 1234567 7654321
Celkem: 4955
l 16 255 255
255 = ff (16)
x 18 25 97
Nespravny vstup.

Vstupni intervaly:
c 100 90 100
Nespravny vstup.

Vstupni intervaly:
l radix 10 20
Nespravny vstup.

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).
- Úloha obsahuje testy povinné, nepovinné a bonusové:
 - nezvládnutí povinného testu (vysoká četnost nesprávných výsledků, překročení doby běhu, pád programu, ...) znamená odmítnutí úlohy a hodnocení 0 bodů,
 - nezvládnutí nepovinného testu znamená snížení celkového výsledku, program ale bude hodnocen nenulovým počtem bodů,
 - nezvládnutí bonusového testu nemá negativní vliv na hodnocení programu, naopak, zvládnutí bonusového testu celkové hodnocení zlepšuje.
- Pro zvládnutí povinných a nepovinných testů postačuje rozumná implementace naivních algoritmů. Vstupy v těchto testech nejsou velké, pro reprezentaci čísel postačuje datový typ `int`.
- V bonusovém testu jsou zadávány pouze typu C pro obrovské intervaly hodnot. Pro reprezentaci čísel je potřeba použít datové typu o velikosti 64 bitů, např. `long long int`. Navíc je pro výpočet potřeba použít lepší než naivní algoritmus. Budete-li při implementaci používat čísla typu `long long int`, použijte pro jejich načítání/zobrazování konverzi `%lld`.
- Úlohu lze vyřešit bez použití funkcí. Pokud ale správně použijete funkce, bude program přehlednější a bude se snáze ladit.
- Pro načítání vstupu se hodí funkce `scanf`.
- 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 (materiály -> cvičebnice -> video tutoriály), jak testovací data použít a jak testování zautomatizovat.
- Váš program bude spuštěn 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 s paměťovým omezením neměla mít problém). Pro povinné a nepovinné testy je doba běhu programu nastavena tak, aby postačovala pro program, který rozumně implementuje naivní algoritmus.
- Slovní popis struktury platných vstupních dat není zcela exaktní. Proto připojujeme i formální popis vstupního jazyka v EBNF:

```

input      ::= { whiteSpace } { cmd { whiteSpace } }
whiteSpace ::= ' ' | '\t' | '\n' | '\r'
cmd        ::= ( 'c' | 'l' ) { whiteSpace } integer { whiteSpace } integer { whiteSpace } integer
integer    ::= digit { digit }
digit      ::= '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'

```

Vzorová data:

[Download](#)

Odevzdat:

Choose File

[Odevzdat](#)

☒ Referenční řešení

- **Hodnotitel: automat**
 - Program zkompilován

- o Test 'Základní test se vstupy dle ukázky': Úspěch
 - Dosaženo: 100.00 %, požadováno: 100.00 %
 - Max doba běhu: 0.006 s (limit: 2.000 s)
 - Celková doba běhu: 0.023 s
 - Úspěch v závazném testu, hodnocení: 100.00 %
- o Test 'Test mezních hodnot': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Max doba běhu: 0.006 s (limit: 2.000 s)
 - Celková doba běhu: 0.089 s
 - Úspěch v nepovinném testu, hodnocení: 100.00 %
- o Test 'Test ošetření nesprávných vstupních dat': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Max doba běhu: 0.005 s (limit: 2.000 s)
 - Celková doba běhu: 0.073 s
 - Úspěch v nepovinném testu, hodnocení: 100.00 %
- o Test 'Test náhodnými daty': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Max doba běhu: 0.031 s (limit: 2.000 s)
 - Celková doba běhu: 0.118 s
 - Úspěch v nepovinném testu, hodnocení: 100.00 %
- o Test 'Test rychlosti (bonus)': Úspěch
 - Dosaženo: 100.00 %, požadováno: 100.00 %
 - Max doba běhu: 0.005 s (limit: 1.000 s)
 - Celková doba běhu: 0.020 s
 - Úspěch v bonusovém testu, hodnocení: 150.00 %
- o Celkové hodnocení: 150.00 % (= 1.00 * 1.00 * 1.00 * 1.00 * 1.50)
- Celkové procentní hodnocení: 150.00 %
- Bonus za včasné odevzdání: 0.50
- Celkem bodů: 1.50 * (5.00 + 0.50) = 8.25

		Celkem	Průměr	Maximum	Jméno funkce
SW metriky:	Funkce:	10	--	--	--
	Řádek kódu:	147	14.70 ± 9.53	33	cntSymmetricRadixFast
	Cyklomatická složitost:	37	3.70 ± 2.37	10	main