HW 04 - Prvočíselný rozklad

Termín odevzdání	30.10.2021 23:59 Bonusová úloha - 08.01.2022 23:59 PST[1)]
Povinné zadání	2b kontrola Coding Stylu [/b211/courses/b0b36prp/resources/tessun/start]
Volitelné zadání	3b kontrola Coding Stylu [/b211/courses/b0b36prp/resources/tessun/start]
Bonusové zadání	**5b** kontrola Coding Stylu [/b211/courses/b0b36prp/resources/tessun/start]
Počet uploadů	20
Podpůrné soubory	b0b36prp-hw04.zip [/b211/_media/courses/b0b36prp/hw/b0b36prp-hw04.zip]

U této úlohy bude cvičící ručně hodnotit dodržení Coding Stylu [/b211/courses/b0b36prp/resources/tessun/start]. Hodnocení je popsáno zde [/b211/courses/b0b36prp/hw/start].
Možná vám to dosud nepřišlo, ale relativně značná část řešení domácích úkolů nespočívá v samotném kódování, ale v přípravě, tj. čtení a pochopení zadání, návrhu řešení a také testování. Samotné "*bušení*" do klávesnice přestane (by mělo) v tomto a následujících úkolech dominantní. Tím spíše, pokud se vaše "*psací*" dovednosti postupně zlepšují.
Implementujte program, který načte ze standardního vstupu seznam celých kladných čísel zakončený nulou a provede jejich prvočíselný rozklad. Vstupní čísla jsou na jednotlivých řádcích. Číslo 1 je speciální případ vstupu, při kterém vypište "1" jako "prvočíselný rozklad".

Pro řešení této úlohy (a to včetně bonusové části) vystačíte s polem. Pokud máte potřebu používat dynamickou alokaci, tak se ubíráte špatným směrem. Je dobré si uvědomit, že automatické proměnné jsou ukládány na zásobníku, který má zpravidla velikost 8MB. Pokud vám program končí chybou (např. segmentation fault ./a.out) je velmi pravděpodobné, že alokujete příliš mnoho paměti. Pro Eratosthenovo síto stáčí pole o velikosti 1M. Pokud bude mít prvek velikost 8 bytů, tak se již blížíte limitu. Velmi dobře lze přístup k paměti odladit programem valgrind [http://valgrind.org/docs/manual/quick-start.html].
Pokud bude na vstupu záporné číslo nebo jiný neočekávaný vstup, vypiště chybovou hlášku " `Error: Chybny vstup!` " na standardní chybový výstup a ukončete program s návratovou hodnotou `100` .

Pro testování funkčnosti program před jeho odevzdáním lze využít přiložené vstupní a referenční výstupní soubory. Dále je možné testovat také generátorem a referečním řešení viz Testování HW programů před odevzdáním [/b211/courses/b0b36prp/tutorials/testing]. Pro generování volitelného zadání použijte dodatečný přepínač `-optional` .
Generování velkých náhodných čísel (pro bonusovou část) může trvat relativně dlouho, proto jsou při generování vypisovány na standardní chybový výstup znaky `.` indikující, že program pracuje. Generujte přímo do souboru s přesměrování např. `./b0b36prp-hw04b-genref -generate >bonus_input-1.txt`

Příklad 1 - pub01

Standardní vstup	Očekávaný výstup	Očekávaný chybový výstup	Návratová hodnota

`1`
`11`
`120`
`8`
`0`
	`Prvociselny rozklad cisla 1 je:`
	`1`
	`Prvociselny rozklad cisla 11 je:`
	`11`
	`Prvociselny rozklad cisla 120 je:`
	`2^3 x 3 x 5`
	`Prvociselny rozklad cisla 8 je:`
	`2^3`
		žádný	0

Příklad 2 - pub02

Standardní vstup	Očekávaný výstup	Očekávaný chybový výstup	Návratová hodnota
12 -2 100 0	Prvociselny rozklad cisla 12 je: 2^2 x 3	Error: Chybny vstup!	100

Příklad 3 - pub03

Standardní vstup	Očekávaný výstup	Očekávaný chybový výstup	Návratová hodnota
12 b 100 0	Prvociselny rozklad cisla 12 je: 2^2 x 3	Error: Chybny vstup!	100

Povinné zadání

Na vstupu jsou pouze celá čísla menší než 10^8, která jsou plně reprezentovatelná datovým typem 32-bitového `int`. Na správné vyřešení v tomto případě postačí využít cykly a zpracovávat jednotlivá čísla jedno po druhém. Další datová struktura (pole) pro urychlení rozkladu na prvočísla není nutná.

Volitelné zadání

Na vstupu jsou pouze celá čísla, která jsou reprezentovatelná pomocí 64-bitového celočíselného znamenkového datového typu. Vzhledem k náročnosti úkolu předpokládáme, že největší prvočíslo v prvočíselném rozkladu je menší než 10^6. Aby byl váš algoritmus efektivní pro opakovaná volání, tak si předpočítejte tabulku všech prvočísel do hodnoty 10^6 algoritmem Eratosthenovo síto [https://cs.wikipedia.org/wiki/Eratosthenovo_s%C3%ADto]. Následně je možné zkoušet dělení pouze nalezenými prvočísly a významně tak výpočet urychlit. Proto je nutné uložit nalezená prvočísla do samostatného pole, aby nebylo potřeba vždy procházet celé Eratosthenovo síto.

Doporučení: Pro uložení nalezených prvočísel používejte “pouze” 32-bitový celočíselný znamenkový typ.

Příklad 4 - pub04

Standardní vstup	Očekávaný výstup	Očekávaný chybový výstup	Návratová hodnota
991350783547 0	Prvociselny rozklad cisla 991350783547 je: 995663 x 995669	žádný	0

Příklad časové náročnosti

Rychlost vašecho programu pro volitelnou část můžete testovat například na rozkladu číselné řady od 1 do N. Výpočetní časy různých verzí programu spuštěných na standardním počítači[2] a zkompilovaných se zapnutou optimalizací (-O3) jsou zaznamenány v následující tabulce. Pro představu jsme otestovali i instance opt01 až opt03 z odevzdávacího systému. V přepočtu na rychlost počítače v odevzdávacím systému odpovídá časový limit přibližně 1.8 až 2 s pro volitelné úlohy opt*.

Řešení / Vstupní soubor	1-10000	1-50000	1-100000	opt01	opt02	opt03
Naivní I	0.18 s	3.0 s	11.6 s	4.6 s	DNF [3]	DNF [4]
Naivní II	0.16 s	2.3 s	8.8 s	4.3 s	4.4 s	4.6 s
Referenční	0.04 s	0.3 s	1.1 s	0.4 s	0.4 s	0.4 s

Naivní řešení I

Vstupní číslo zkouší dělit všemi čísly menšími než je vstupní číslo.

Naivní řešení II

Vstupní číslo zkouší dělit všemi menšími čísly. Algoritmus končí, když je dělenec menší než dělitel.

Referenční řešení

Využívá předpočítaná prvočísla do hodnoty 10^6. Algoritmus končí, když je dělenec menší než dělitel.

Testovací instance

Optional 01

Přibližně 8000 čísel, které jsou většinou prvočísly z intervalu 2 až 100000.

Optional 02

Přibližně 7000 čísel, které jsou většinou složené ze dvou prvočísel z intervalu 2 až 100000.

Optional 03

Přibližně 300 čísel, které jsou většinou složené ze dvou prvočísel v intervalu 900000 až 1000000.

Bonusové zadání

Na vstupu mohou být celá kladná čísla dlouhá až 100 cifer. Je tedy nutné vytvořit jejich vlastní reprezentaci v počítači spolu s příslušnými operacemi celočíselného dělení se zbytkem. Největší číslo v prvočíselném rozkladu bude vždy menší než 10^6. Při implementaci nepoužívejte cízí kód ani žádnou specializovanou knihovnu pro práci s velkými čísly. V tomto úkolu nemusíte používat Eratostenovo síto, protože časový limit nebude nijak přísný. Cílem je především práce s velkými čísly.

Příklad 6 - pub06b

Rozkládané číslo[5] je $(995663 * 995669)^8$:

Vstup	`9328650737199920596297735136147893882665803050839205919257403713922543170645848557850889157457610`
Výstup	`Prvociselny rozklad cisla 9328650737199920596297735136147893882665803050839205919257403713922543170645848557850889:` `995663^8 x 995669^8`

Výpočetní čas na běžném počítači (bez optimalizace):

```
real    0m1.279s
user    0m1.264s
sys     0m0.004s
```

Časový limit v odevzdávacím systému je přibližně desetinásobek referenčního programu (10s).

Bonusové zadání má v odevzdávacím systému vlastní záložku HW04B. Bonusová část se po uplynutí termínu uzavře. Není proto nastavena žádná penalizace za pozdní odevzdání.

Odevzdání

U tohoto domácího úkolu (jako jedinného) je linkována matematická knihovna `math.h` pomocí přepínače `-lm` .
Neukládejte prvočísla přímo do zdrojového kódu, **maximální velikost souboru** `main.c` **je 50 kB**.
Veřejné příklady + Makefile: b0b36prp-hw04.zip [/b211/_media/courses/b0b36prp/hw/b0b36prp-hw04.zip]

	Povinné zadání	Volitelné zadání	Bonusové zadání
Název v BRUTE	HW04		HW04B
Odevzdávané soubory	main.c		
Argumenty při spuštění	žádné		
Návratová hodnota	`0` ; Program skončil úspěšně `100` ; "Error: Chybny vstup!" → `stderr`		

	Povinné zadání	Volitelné zadání	Bonusové zadání
Kompilace pomocí	clang -pedantic -Wall -Werror -std=c99 -O3 -lm		
Velikost zásobníku	8 MB		
Velikost haldy	*Dynamická alokace není potřeba*		
Očekávaná časová složitost[6]	$\mathcal{O}(v)$	$\approx \mathcal{O}\left(\frac{v}{\log v}\right)$	$\approx \mathcal{O}\left(\frac{v}{\log v}\right)$
Procvičované oblasti	vnořené cykly	pole	pole

1)

PST [https://www.timeanddate.com/time/zones/pst]

2)

Notebook, Intel Core i5 CPU M480 @ 2.67 GHz, 8GB RAM DDR3 1066 MHz.

3) , 4)

Více než 10 minut.

5)

Je možné si ověřit na Wolfram Alpha - výpočet [https://www.wolframalpha.com/input/?i=(995663+*+995669)%5E8].

6)

Rozklad jednoho vstupního čísla v závislosti na jeho velikosti v. Časová složitost u volitelné části je dána počtem prvočísel do zadané hodnoty v (Prime number [https://en.wikipedia.org/wiki/Prime_number]).