

HW 01 - Standardní vstup/výstup v C++

Termín odevzdání	24.3.2024 7:00
Možný bodový zisk	až 8b (3+3+2)
Počet uploadů	10

Napište program, který načte formátovaný text ze standardního vstupu, zpracuje jej a výsledek vytiskne na standardní výstup.

Zadání a kritéria

- Načítaná data jsou rozdělena na konfiguraci a data oddělena rovná se `=` na samostatném řádku
- Konfigurace začíná klíčovým slovem `config` a může nabývat těchto vlastností přes tečkovou notaci, viz. příklad níže
 - `min` je min. hodnota, která se může vyskytnout v načítaných datech
 - `max` je max. hodnota, která se může vyskytnout v načítaných datech
 - `width` který označuje počet znaků buňky na výstupu a min. hodnota je 1.
 - `align` zarovnává čísla v buňkách vlevo když příkaz nabývá hodnoty `left` nebo vpravo při hodnotě `right`
- Čísla na vstupu jsou celá, v rozsahu `config.min` - `config.max` včetně krajních hodnot.
- Počet řádků není omezen.
- Jednotlivé sloupce jsou označeny `A` - `Z`.
- Jednotlivé řádky jsou označeny čísly `1` - `n`, kde `n` je poslední řádek
- V případě nezadání konfiguračních parametrů na vstupu (viz dále) budou použity defaultní hodnoty a uživatel o tom bude informován.
 - `min=-99`, na standardní výstup se vypíše `config.min=-99`
 - `max=100`, na standardní výstup se vypíše `config.max=100`
 - `width=3`, na standardní výstup se vypíše `config.width=3`
 - `align=left`, na standardní výstup se vypíše `config.align=left`
- Buňky jsou vždy odděleny vždy středníkem `;` a řádek končí znakem `\n`
- V načítaných datech se může vyskytnout příkaz `SUM(od:do)`, který bude sčítat hodnoty na daném řádku podle rozsahu `od:do`. Do příslušné buňky poté uloží výsledek
- Pokud řádek bude kratší, vypíše se prázdná buňka
- Vypsání parametrů a vykreslení tabulky bude následující
 - Každý řádek má na konci znak nového řádku `\n`
 - Nejprve se vypíší všechny parametry v pořadí, které jsou uvedeny v bodu 2
 - Parametry a tabulka jsou odděleny novým řádkem
 - Každá buňka je ohraničena `+` v jejím rohu
 - Sloupce jsou odděleny pomocí svislítka `|`
 - Řádky jsou odděleny pomocí pomlčky `-`
 - První a poslední znak v buňce bude vždy mezera
 - V případě splnění všech kritérií, návratová hodnota bude `0`

Chybová hlášení

1. Na vstupu bude číslo mimo rozsah `config.min` a `config.max`, na chybový výstup se vypíše `Out of range` a návratová hodnota bude `100`
2. Na vstupu se objeví místo čísla písmeno nebo textový řetězec vyjma příkazu `SUM(od:do)`, na chybový výstup se vypíše `Invalid input` a návratová hodnota bude `101`. Tato chybová hláška se také vypíše, když interval v příkazu `SUM()` bude rozsah, který program nemůže spočítat (např. existují buňky A-C a program bude chtít spočítat A-G).
3. Hodnota vstupních parametrů bude mimo povolené hodnoty, na chybový výstup se vypíše `Invalid configuration` a návratová hodnota bude `102`. Toto platí i v případě `min > max`.
4. Šířka buňky bude moc úzká na vypsání čísla, na chybový výstup se vypíše `Cell is too short` a návratová hodnota bude `103`

Bonusové zadání a kritéria

K základnímu zadání přibudou ještě tato následující kritéria, zbytek zůstává ze základního zadání

1. Konfigurace `config` bude rozšířena o následující parametry
 1. **stretch** parametr udává, zda-li se buňka má roztahovat podle největšího čísla, v případě, že se do buňky nevejde (`stretch=1`) nebo se místo čísla vypíše `n x #` (tj. buňka bude vyplněna znaky `#`), kde `n` je šířka buňky (`stretch=0`)
 2. **header** vypíše hlavičku jednotlivých sloupců a řádků když `header=1`, jinak ji vypisovat nebude `header=0`
2. V případě nezadání bonusových konfiguračních parametrů na vstupu, se bude uvažovat, že se tyto parametry nezadaly a do standardního výstupu se nebudou vypisovat

Chybová hlášení pro bonus

1. Chybová hlášení zůstávají stejná vyjma posledního bodu, který je řešením bonusového zadání. Pokud není parametr `stretch` zadáný, na chybový výstup se vypíše `Cell is too short` a návratová hodnota bude `103`.

Implementace načítacích funkcí

Pokud chcete získat další 2 body, je třeba implementovat vlastní řešení načítání konfigurace a sumy. V souborech `parse.cpp/hpp` je uvedeno jedno z možných řešení jak načítání řešit. V případě vlastního řešení načítacích funkcí je třeba informovat cvičícího, který rozhodne jestli body přidělit nebo nepřidělit. Brute bohužel toto neumí rozpoznat a je třeba manuálního řešení.

Příklady

Standardní vstup (cin)	Očekávaný výstup (cout)	Očekávaný chybový výstup (cerr)	Návratová hodnota
<pre>config.min=-99 config.max=150 config.width=3 config.align=left = 10;20;20;SUM(A:C) -10;-20;-30;-40;-50</pre>	<pre>config.min=-99 config.max=150 config.width=3 config.align=left +---+---+---+---+---+---+ A B C D E +---+---+---+---+---+---+ 1 10 20 20 50 +---+---+---+---+---+---+ 2 -10 -20 -30 -40 -50 +---+---+---+---+---+---+</pre>		0

Standardní vstup (cin)	Očekávaný výstup (cout)	Očekávaný chybový výstup (cerr)	Návratová hodnota
<pre>config.width=3 config.align=right = 10;20;30;SUM(A:C) -40;-50;-60;-70 90;80;-20;-30</pre>	<pre>config.min=-99 config.max=100 config.width=3 config.align=right +-----+-----+-----+-----+ A B C D +-----+-----+-----+-----+ 1 10 20 30 60 +-----+-----+-----+-----+ 2 -40 -50 -60 -70 +-----+-----+-----+-----+ 3 90 80 -20 -30 +-----+-----+-----+-----+</pre>		0
<pre>config.min=0 config.max=10 = 10;20;30;SUM(A:C) 60;20;70</pre>		Out of range	100
<pre>config.min=0 config.max=1000 = a;b;c;400;500;SUM(A:E) 60;20;30;40;50;60;70</pre>		Invalid input	101
<pre>config.min=0 config.max=1000 config.width=-5 = 100;200;300;400;500;SUM(A:E) 60;20;30;40;50;60;70</pre>		Invalid configuration	102
<pre>config.min=0 config.max=1000 config.width=2 = 100;200;300;400;500;SUM(A:E) 60;20;30;40;50;60;70</pre>		Cell is too short	103
<pre>config.min=-5000 config.max=5000 config.width=3 config.align=left config.stretch=0 =</pre>	<pre>config.min=-5000 config.max=5000 config.width=3 config.align=left config.stretch=0 +-----+-----+-----+-----+ A B C D </pre>		0

Standardní vstup (cin)	Očekávaný výstup (cout)	Očekávaný chybový výstup (cerr)	Návratová hodnota
3300;4400;20;SUM(A:C) -40;-50;-60	+-----+-----+-----+-----+-----+ 1 ### ### 20 ### +-----+-----+-----+-----+-----+ 2 -40 -50 -60 +-----+-----+-----+-----+-----+		
config.min=-5000 config.max=5000 config.width=3 config.align=left config.stretch=1 = 3300;4400;20;SUM(A:C) -40;-50;-60	config.min=-5000 config.max=5000 config.width=4 config.align=left config.stretch=1 +-----+-----+-----+-----+-----+ A B C D +-----+-----+-----+-----+-----+ 1 3300 4400 20 7720 +-----+-----+-----+-----+-----+ 2 -40 -50 -60 +-----+-----+-----+-----+-----+		0
config.min=-99 config.max=100 config.width=4 config.align=left config.header=0 = 33;44;20;SUM(A:C) -40;-50;SUM(A:B);SUM(A:C) 3;14;15;92	config.min=-99 config.max=100 config.width=4 config.align=left config.header=0 +-----+-----+-----+-----+ 33 44 20 97 +-----+-----+-----+-----+ -40 -50 -90 -180 +-----+-----+-----+-----+ 3 14 15 92 +-----+-----+-----+-----+		0

Odevzdání

Do systému BRUTE odevzdejte zabalený archiv s *.cpp a případně *.hpp soubory s libovolným jménem. Všechny soubory musí být v kořenovém adresáři archivu.

Visual Studio Code nepodporuje jednoduchou správu projektů složených z více zdrojových souborů. Doporučujeme proto připravit si makefile a make spouštět z příkazového řádku.

Veřejné příklady + Template: [HW01.zip \[wiki/_media/courses/b2b99ppc/hw/hw01.zip\]](#) ([gitlab \[https://gitlab.fel.cvut.cz/viteks/ppc/-/tree/master/homeworks/hw01\]](#))

Povinné i bonusové zadání	
Název v BRUTE	HW01
Soubor/y	jakékoliv množství *.cpp/*.hpp souborů
Argumenty při spuštění	žádné

Povinné i bonusové zadání	
Kompilace pomoci	g++ -pedantic -Wall -Werror -std=c++17
Procvičované oblasti	standardní vstup a výstup v C++

courses/b2b99ppc/hw/hw01.txt · Last modified: 2024/02/15 15:21 by nentvond

Copyright © 2024 CTU in Prague | Operated by [IT Center](#) of [Faculty of Electrical Engineering](#) | Bug reports and suggestions
[Helpdesk CTU](#)