Programovanie (1) v C/C++ 2026/27
Cvičenia 10, miniprojekt
Editor

Tento **miniprojekt** je väčšia úloha, na ktorú máte viac času ako na bežné príklady z cvičení, ale máte za ňu aj viac bodov. Začnite na nej pracovať v predstihu, nenechávajte si ju na poslednú chvíľu.

Na miniprojekte pracujte **samostatne**, neukazujte svoj program spolužiakom a nepoužívajte nástroje umelej inteligencie. V prípade otázok kontaktujte vyučujúcich predmetu.

Študenti, ktorí majú z testu pre pokročilých uznané cvičenia 10, budú mať uznaný aj tento miniprojekt a nemusia ho odovzdávať.

Cieľom tejto úlohy je precvičiť si prácu s načítaním vstupu a s vhodným použitím dátových štruktúr. Predtým ako začnete programovať, si poriadne rozmyslite, aké dátové štruktúry (polia, matice, struct-y a pod.) chcete v programe použiť.

Aj v tomto miniprojekte ide predovšetkým o správnosť vášho programu, ale budeme hodnotiť aj programátorský štýl (odsadenie, vhodné členenie na funkcie, názvy premenných a funkcií, komentáre, celková zrozumiteľnosť kódu).

Študenti, ktorí majú z testu pre pokročilých uznané cvičenia 10, budú mať uznaný aj tento miniprojekt a nemusia ho odovzdávať.

V tejto úlohe budeme simulovať jednoduchý textový editor. Editovaný text pozostáva z niekoľkých riadkov, ktoré môžu obsahovať ľubovoľné tzv. vytlačiteľné (anglicky printable) znaky, t.j. znaky s ASCII kódom 32 (medzera) až 126 (vlnka `'~'`). Editovaný text teda nebude obsahovať diakritiku ani špeciálne riadiace znaky. Prvý riadok vstupu z konzoly obsahuje kladné celé čísla `MAXR`, `MAXC` so sprievodným textom vo formáte ako v príklade nižšie. Prvé číslo určuje maximálny povolený počet riadkov v editore a druhé číslo určuje maximálny povolený počet znakov na riadku (do tohto počtu sa počítajú iba vytlačiteľné znaky, ale nie znak konca riadku ani ukončovacia nula).

Nasledujúca časť vstupu obsahuje počiatočný text v editore. Prvý z týchto riadkov obsahuje iba text `START`, tento riadok ignorujte. Potom nasleduje niekoľko riadkov textu a za nimi ide riadok s textom `END`, ktorý slúži na ukončenie tejto časti vstupu, ale do editora ho nevkladajte. Každý riadok v počiatočnom texte bude obsahovať najviac `MAXC` vytlačiteľných znakov a počet riadkov bude najviac `MAXR`. Niektoré riadky môžu byť aj prázdne.

V každom bode simulácie je v editore kurzor na nejakom riadku a stĺpci, pričom riadky aj stĺpce číslujeme od nuly. Ak je v určitom riadku n vytlačiteľných znakov, kurzor sa môže nachádzať v stĺpcoch 0 až n, t.j. môže byť aj za posledným vytlačiteľným znakom riadku. Do znakov v riadku nepočítame znak konca riadku. V prázdnom riadku je kurzor vždy v stĺpci nula. Číslo riadku s kurzorom bude vždy v rozmedzí 0 až `MAXR-1`. Na začiatku simulácie je kurzor na pozícii 0,0.

Zvyšok vstupu obsahuje príkazy pre editor oddelené medzerami alebo koncami riadkov. Ich význam je nasledovný:

- Príkaz `LEFT` znamená posun kurzoru o znak doľava. Ak bol kurzor na najľavejšom znaku riadku, posunie sa **za** posledný znak predchádzajúceho riadku. Ak bol na úplnom začiatku textu, neposúva sa vôbec.
- Príkaz `RIGHT` znamená posun kurzoru o znak doprava. Ak bol kurzor za posledným znakom v riadku, posunie sa na začiatok nasledujúceho riadku. Ak by sa tým ale dostal za posledný povolený riadok `MAXR-1`, neposúva sa vôbec.
- Príkaz `INSERT` znamená vloženie nového znaku. Za týmto príkazom vždy nasleduje práve jedna medzera, apostrof, vložený vytlačiteľný znak a opäť apostrof. Pozor, vložený znak môže byť aj medzera, ale pre jednoduchosť to nebude apostrof. Znak sa vloží na pozíciu kurzora, pričom znaky napravo od pozície vkladania sa posunú doprava. Ak by tým počet znakov na riadku mal prekročiť `MAXC`, znak sa nevloží a riadok zostane v pôvodnom stave.

- Príkaz `DELETE` znamená zmazanie znaku, na ktorom je kurzor. Všetky znaky napravo od neho sa posunú o jedno miesto doľava. Ak je kurzor za posledným znakom na riadku, riadok sa nemení.
- Príkaz `PRINT` znamená výpis aktuálneho textu v editore. Za týmto príkazom ide buď meno súboru alebo pomlčka. Meno súboru nebude obsahovať biele znaky a bude mať dĺžku najviac 20 znakov. Ak bolo zadané meno súboru, tento výpis robte do súboru so zadaným menom, pričom pridávajte na jeho koniec, t.j. otvorte ho v móde `"a"` (append). Ak je meno súboru pomlčka, vypisujte výstup na konzolu. Výpis začnite a ukončite riadkami `START` a `END`, podobne ako na vstupe. Výpis končí posledným neprázdnym riadkom v editore, vypisujte ale prázdne riadky, za ktorými ešte ide neprázdny riadok.

  Pokúste sa v programe neopakovať kód na výpis editora dvakrát, ale spravte ho napríklad do funkcie, ktorá dokáže vypisovať na konzolu aj do súboru (viď prednáška 10).
- Príkaz `EXIT` ukončí prácu editora.

Po príkazoch `LEFT`, `RIGHT`, `INSERT` a `DELETE` vypíšte na konzolu práve vykonaný príkaz a číslo riadku a stĺpca, v ktorom sa nachádza kurzor (viď príklad výstupu).

V programe je zakázané používať polia konštantných veľkostí. Výnimkou sú len polia potrebné na uloženie jedného názvu súboru alebo jedného kľúčového slova vstupu (napr. `MAXR`, `LEFT` a pod.). Všetky ostatné polia alokujte dynamicky (príkazom `new` prípadne `malloc`) alebo použite štruktúry, ktoré menia veľkosť podľa potreby (`vector`, `string` a podobne). Dynamicky alokovanú pamäť odalokujte. Používajte kompilátor pre jazyk C++.

Príklad vstupu

```
MAXR 4 MAXC 3
START
a
 c
END
LEFT PRINT vystup.txt
DELETE RIGHT RIGHT
INSERT ' ' INSERT 'x'
PRINT - DELETE
INSERT 'y'
PRINT - RIGHT
RIGHT RIGHT RIGHT
RIGHT PRINT vystup.txt
EXIT
```

Výstup na konzolu

```
LEFT 0 0
DELETE 0 0
RIGHT 1 0
RIGHT 1 1
INSERT 1 1
INSERT 1 1
START

  c
END
DELETE 1 1
INSERT 1 1
START

 yc
END
RIGHT 1 2
RIGHT 1 3
RIGHT 2 0
RIGHT 3 0
RIGHT 3 0
```

Výstup do súboru `vystup.txt`

```
START
a
 c
END
START

 yc
END
```

Poznámka: Odporúčame vám odsimulovať si vstupné príkazy na papieri, aby ste rozumeli, ako vznikol uvedený výstup. Prázdny riadok v príklade vznikol vymazaním písmena `a`. Vloženie písmena `x` sa nepodarilo, lebo riadok už dosiahol maximálnu šírku.