Programovanie (1) v C/C++ 2025/26
Tréningový príklad na skúšku 2d
Vyfarbovanie

Toto je cvičný príklad pre druhý príklad na skúške. Odporúčame prečítať si zadanie až keď budete úlohu skutočne cvične riešiť. Tento príklad bol v minulosti použitý ako domáca úloha, svojim charakterom sa však pomerne podobá na druhý príklad zo skúšky.

V programe budeme pracovať s maticou s určitým počtom riadkov a stĺpcov. V každom políčku matice je uložené jedno malé písmeno anglickej abecedy ('a'..'z'), ktoré reprezentuje farbu tohto políčka. Avšak namiesto polí je matica uložená v štruktúre pripomínajúcej spájaný zoznam, v ktorom každé políčko je `struct` nazvaný `square`, s položkami, ktoré obsahujú údaje o tomto políčku a smerníky na štyri susedné políčka. Program si potom pamätá iba smerník na štruktúru pre ľavé horné políčko matice. Konkrétne `square` obsahuje nasledujúce položky:

- Položka `color` je farba políčka uložená ako znak (`char`)
- Položky `row` a `column` sú celé čísla predstavujúce číslo riadku a stĺpca daného políčka. Ľavé horné políčko má obidve súradnice 0.
- Položka `next` je pole štyroch smerníkov na susedné políčka. Konkrétne `next[0]` je políčko o riadok vyššie a ďalšie smerníky idú v smere hodinových ročičiek, t.j. `next[1]` je políčko vpravo od aktuálneho, `next[2]` dole a `next[3]` vľavo. Pre ľahšie používanie máme zadefinované aj konštanty `UP`, `RIGHT`, `DOWN` a `LEFT`, ktoré sú indexom príslušného smeru do poľa `next`. Napríklad `UP=0`, t.j. `next[UP]` je smerník na suseda smerom hore od aktuálneho políčka. Ak je políčko v niektorom z okrajových riadkov alebo stĺpcov, nemá všetkých susedov a smerníky na neexistujúcich susedov sú nastavené na `NULL`.

V programe je už naprogramovaný vstup aj výstup. Program sa má správať nasledovne. Zo vstupu načíta najskôr samotnú maticu. Tá je zadaná na prvých niekoľkých riadkoch vstupu. Na každom riadku sú malé písmená pre farby jednotlivých políčok zadané bez medzier. Za maticou nasleduje prázdny riadok, v ktorom je len znak konca riadku. Potom nasleduje riadok, ktorý obsahuje postupnosť písmen `U`, `D`, `L`, `R`, ktoré sú skratkami za smery `UP`, `RIGHT`, `DOWN` a `LEFT`. Program simuluje pohyb po matici, pričom začne z ľavého horného rohu a podľa jednotlivých príkazov sa v každom kroku pohne hore, dole, doľava alebo doprava. Po každom kroku vypíše aktuálne súradnice vo formáte (riadok,stĺpec,farba), rovnako vypíše aj počiatočné súradnice pred prvým pohybom. Ak by sme sa však aktuálnym pohybom dostali mimo matice, zostaneme stáť na pôvodnom políčku a vypíšeme namiesto súradníc a farby pomlčky. Keď program odsimuluje všetky zadané pohyby, spustí prefarbovanie oblasti, na ktorej zostal stáť, pričom pôvodnú farbu nahradí špeciálnou farbou bodka (`'.'`). Pri vyfarbovaní spočíta počet prefarbených políčok a ten vypíše na posledný riadok výstupu. Môžete predpokladať, že vstupná matica má aspoň jeden riadok a stĺpec a že dĺžka každého riadku vo vstupe je najviac 1000 znakov, vrátane konca riadka.

Vašou úlohou je naprogramovať nasledujúce tri funkcie. Nemeňte už hotové časti programu, ale v prípade potreby pridajte ďalšie pomocné funkcie.

- Funkcia `addPointers` doplní do matice chýbajúce smerníky. Pri načítaní sa v matici každému políčku správne nastaví smerník na pravého suseda (`next[RIGHT]`) a políčkam v najľavejšom stĺpci sa nastaví aj smerník na spodného suseda (`next[DOWN]`). Ostatné smerníky sú všetky nastavené na `NULL` a vašim cieľom je nastaviť im správne hodnoty. Budete teda nastavovať hodnoty `next[LEFT]` a `next[UP]` v celej matici a `next[DOWN]` všade okrem nultého stĺpca, kde už sú.
- Funkcia `computeArea` dostane smerník na určité políčko matice. Táto funkcia prefarbí súvislú jednofarebnú oblasť, ktorej súčasťou je zadané políčko, na novú farbu bodka ('.'). Ako návratovú hodnotu vráti počet prefarbených políčok. Na tento účel adaptujte rekurzívny algoritmus na vyfarbovanie z prednášky, prípadne použite jeho nerekurzívnu verziu s použitím zásobníka alebo radu.

- Funkcia `destroy` dostane smerník na ľavé horné políčko matice a odalokuje všetky políčka matice. Môžete vychádzať z kódu na uvoľnenie spájaného zoznamu. Dajte pozor, aby ste nepristupovali k už odalokovaným hodnotám.

Príklad vstupu:

```
abcd
abcb
abbb

UDDDRRRRUUULLLDR
```

Príklad výstupu:

```
(0,0,a)
(-,-,-)
(1,0,a)
(2,0,a)
(-,-,-)
(2,1,b)
(2,2,b)
(2,3,b)
(-,-,-)
(1,3,b)
(0,3,d)
(-,-,-)
(0,2,c)
(0,1,b)
(0,0,a)
(1,0,a)
(1,1,b)
area 6
```