

Упражнение №2 по СДП

Потоци, файлове. Стек.

1. Потоци, файлове.

Обработка на текстови файлове и двоични файлове. Форматиране на изхода.

Задача 1.

Даден е текстов файл, в който е записана информация за горивото, което се продава на една бензиностанция за даден ден. На всеки ред от файла е записана информация от вида: <номер на колонка> <таб> <количество заредено гориво>.

Дефинирайте следните функции:

- функция, която намира прихода от дадена колонка, ако цената на горивото също е подадено като аргумент на функцията;
- функция, която генерира двоичен файл с информация за средното количество гориво, което се зарежда от всяка колонка, на базата на данните от текстовия файл. Ако в текстовия файл няма информация за дадена колонка, в двоичния файл да се запише 0.
- функция, която записва в текстов файл информацията за колонката, от която е заредено най-много гориво общо. Данните да бъдат в следния формат: <номер на колонка> <общо количество гориво> <обща стойност на горивото>. Всяко поле да е с ширина 15 позиции, числовите стойности да се извеждат закръглени до втория символ след десетичната запетая.

Задача 2.

Даден е файл в csv формат. Всеки ред представлява отделен запис, а отделните данни в записа са разделени със запетая.

Файлът съдържа информация за вашата телефонна сметка и данните са следните: <дата>, <номер, до който е позвъняването>, <продължителност в минути>, <цена на разговора без ДДС>.

Дефинирайте следните функции:

- функция, която проверява дали в текстов файл с горния формат има запис за даден телефонен номер;
- функция, която намира общата продължителност на разговорите с даден телефонен номер;
- функция, която генерира нов файл с обобщена информация за проведените разговори, като всеки телефонен номер присъства точно веднъж. Файлът е със следното форматиране: всяко поле е с дължина 15 символа, подравняването е в дясно, числовите стойности са с точност до втория символ след десетичната запетая.

2. Стек.

Задача 3.

Дадени са два стека от цели числа, подредени в намаляващ ред от върха към дъното. Дефинирайте функция `mergeStacks()`, която чрез операциите `push()` и `pop()` построява нов стек, който се състои от всички елементи на дадените два стека, подредени в нарастващ ред от върха към дъното.

Пример: 5,3,1 6,2,1 -> 1,1,2,3,5,6

Задача 4.

Дадени са бинарните операции:

$$a \$ b = \min(a, b) \text{ и } a @ b = (a + b) \% 10,$$

където a и b са едноцифрени числа. Да се напише програма, която изчислява изрази от вида:

<израз> = <цифра> | (<израз> <операция> <израз>);

<операция> = \$ | @;

Пример: 5; (2 \$ 3); (2 @ (3 \$ 4)) и т.н.

Задача 5.

Дефинирайте функция, която получава като аргумент валиден израз в инфиксен запис и го преобразува в обратен полски запис. Разрешените операции +, -, * и / са двуаргументни, а аргументите са само цифри. (Внимание с приоритета и асоциативността на операциите!)

Задача 6.

Редактор на низове, симулиране на операциите `undo()` и `redo()`.

Даден е низ (може да се използва `std::string`). Върху него могат да се прилагат операциите *изтриване* и *добавяне*. За операцията *изтриване* са необходими начална и крайна позиция за изтриване. Операцията *добавяне* се нуждае от начална позиция и допълнителен низ, който да бъде добавен в основния на посочената позиция. Действието `undo()` отменя последната приложена операция върху низа. Ако операцията е била *изтриване*, изтриятият подниз се възстановява. (Обърнете внимание, че върху низа се прилага обратната операция!) Ако операцията е била *добавяне*, новодобавеният подниз се премахва. След отмяната на операцията, тя не може да бъде отменена повторно, но пак може да приложена отново с действието `redo()`.

Дефинирайте подходящ клас или йерархия от класове, които да описват операциите, които могат да се прилагат върху един низ.

Дефинирайте клас `Editor`, който описва редактора на низове. Конструкторът на класа `Editor` получава като аргумент низа, върху който ще бъдат прилагани операциите. Като член-функции на класа са дефинирани функциите `insert()` и `remove()`, които се съпоставят на операциите *добавяне* и *изтриване* съответно, както и действията `undo()` и `redo()`.

С какъв механизъм ще бъдат реализирани действията `undo()` и `redo()`?

Задача 7.

Напишете програма, която при получен "компресиран" низ на стандартния вход, извежда на стандартния изход декомпресираната му версия.

Компресираният низ съдържа 2 вида конструкции:

- букви, които са символ от 'A' до 'Z' и се декомпресират до същата буква;
- групи, които започват с число, последвано от компресиран низ в скоби. Декомпресират се като се декомпресира низа в скобите и се повтори толкова пъти, колкото е числото.

Примери:

A -> A

AB -> AB

3(A) -> AAA

$3(AB2(B)) \rightarrow ABVBABVBABVB$

$12(ZQ) \rightarrow ZQZQZQZQZQZQZQZQZQZQ$

Задача 8.

Дадена е редица от числа, чиито членове се получават по-следния начин:

- първият елемент е N ;
- вторият се получава като съберем N с 1;
- третият – като се умножи първия с 2 и така последователно всеки елемент се събира с 1 и се добавя в края на редица, след което се умножава по 2 и отново се добавя в редицата.

Да се напише програма, която за дадено N и r намира r -тия пореден елемент на редицата.