

Домашно 3

Общи.

Предавайте всичко необходимо за да може вашият код да бъде компилиран и пуснат (това включва C++ кода, както и файловете на средата, която използвате).

Архива който ще предавате трябва да може да се разархивира и кода да бъде компилиран.

Има ограничение за размера на качените файлове и техният тип - може да изчистите файловете получени при компилация и да архивирате в .zip. Може да се предава повече от веднъж и е желателно да предадете каквото успеете да направите, дори да е непълно.

Задача 1

Напишете функция, която получава като аргумент двумерен масив relations от имена на хора, описващ роднински връзки, както е показано в таблицата:

<i>Име на лице</i>	<i>Майка на лицето</i>	<i>Баща на лицето</i>
Ivan Petrov	Galina Ivanova	Petar Petrov
Penka Georgieva	Tatyana Georgieva	Georgi Georgiev
...
Julia Dimova	Gergana Dimova	

Имената са сортирани лексикографски по първата колоната - „Име на лице“. Някои от клетките във втората или третата колона може да са празни низове, когато няма информация за съответния родител.

Функцията също ще получава като аргументи два символни низа. Задачата е да се провери дали единият е прародител на другия, и ако да - на какво разстояние (през колко поколения). Приемете, че ако единият е родител на другия - разстоянието е 1, ако е дядо/баба на другия - 2 и т.н.

Реализирайте подходяща програма, чрез която да демонстрирате използването на функцията. Тя трябва първоначално да въвежда таблицата с взаимоотношенията, след което да пита потребителя за две имена и да проверява дали те са в роднинска връзка, посредством реализираната функция. Това се повтаря докато потребителя не въведе специалната дума exit за първо име.

Задача 2

Напишете рекурсивна функция която приема “компресиран” низ и връща динамично заделен низ който съдържа декомпресираните данни. Напишете main функцията, която чете от стандартния вход низ, подава го на декомпресиращата функция и извежда резултата на стандартния изход.

Компресираният низ съдържа 2 вида конструкции, и може да считате, че ще е винаги верен:

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

Примери:

AABC -> AABC

R2(AB)3(Z) -> RABABZZZ

AB12(X)2(B3(A)) -> ABXXXXXXXXXXXXBAAABAAA

Задача 3

Имплементирайте функцията

```
char **stringSplit(const char *string, const char *delimiter, bool
keepEmptyElements);
```

Функцията трябва да върне динамично заделен масив от под-стрингове които са получени чрез премахването на всички срещания на `delimiter` от `string`. Масивът трябва да е заделен с точна големина като последният му елемент е `nullptr`. Елементите на масива трябва да са динамично заделен низове с точна дължина. Булевият флаг `keepEmptyElements` определя дали върнатият масив съдържа празни стрингове или не (както в случай на последователни разделители).

Пример:

```
stringSplit("the quick brown fox jumps over the brown fence ", " ", true) ->
{"the", "", "quick", "brown", "fox", "jumps", "over", "the", "brown", "fence",
"", nullptr}
```

Покажете че имплементацията ви работи коректно.

Задача 4

Имате дадения код: <https://gist.github.com/poseidon4o/97cc50c38a347128520f5aa1df1d3f5e>
Опишете какво прави кода, и как работи. Направете промени така че да подобрите качеството на кода.