Домашно 2

**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**

Имплементирайте функция, която по зададени параметри да “изрисува” прогрес бар с ASCII символи, и да го съхрани в масив от символи. Параметрите, които трябва да приема функцията са: сегашен прогрес - число в интервала [0;1], затапващи символи (ляв и десен), булев флаг, указващ дали в бара да се показват процентите, запълващ символ и празен символ, целочислена бройка на деленията на целия бар. Преценете как да ги подредите, така че за някои от тях да можете да зададете смислени стойности по подразбиране. Преценете как да се получи резултатът.

Напишете програма, която прочита от потребителя цяло число - брой секунди и използвайки вашата функция “анимира” запълването на прогрес бара за тези секунди, като той започва от празен и стига до запълнен.

За постигане на по-добро усещане в тази задача е разрешено да използвате готовата библиотечна функция за приспиване на програмата ви:  
std::this\_thread::sleep\_for(std::chrono::milliseconds(T));

за да накарате вашата програма да “чака” поне T милисекунди. Изберете подходящо време за чакане между извеждането на екрана и подходящо увеличение на прогреса, така че “анимацията” да бъде плавна.

Може да принтирате всяко ново състояние на нов ред, или да използвате някоя от функциите за изчистване на символите от конзолата, като например system(“cls”) за Windows или system(“clear”) за Linux/Unix. За тази цел също можете да включите подходящ заглавен файл.

Примерен изход в един момент, показващ прогрес бар със затапващи символи “[“ и “]”, запълващ символ “=”, празен символ интервал и показана стойност на процент на запълване.

[============ 30% ]

Примерен изход със затапващи символи “<” и “>”, запълващ символ “-”, празен символ “.” и показване на процентите:

<-------------------------60%--....................>

**Задача 2**

Трябва да направите система която симулира оттичането на вода в даден терен. Терена е представен като квадратна мрежа, за която е дадена височината във всеки квадрат. Симулацията започва като определено количество вода вали над целият регион. След това следват стъпки на оттичане. На всяка стъпка част от водата попива, а ако клетката има съседи с по-ниска височина, то останалата се оттича към тях пропорционално. Съседите на една клетка са максимално 4 на брой (горе, долу, ляво и дясно) Симулацията приключва когато водата спре да се движи, и всичката вода е попила.

Правилата за това как се оттича водата са следните.

1. Всяка клетка попива фиксирано количество вода.
2. От всяка клетка в която има вода която не е попила, и има съседи с по ниска височина, се оттича вода.
3. Разпределението на количеството вода което се оттича от всяка клетка към съседите ѝ е обратно пропорционално на височините им.
4. Водата в клетка която няма по-ниски съседи остава и попива там.

Накрая на симулацията трябва да изведете цялата квадратна мреже, като във всяка клетка изведете количеството попила вода.

За вход на вашата програма ще получите размерите на квадратната мрежа (2 цели числа W и H - брой клетки по ширина и височина). След това ще получите WxH реални числа които са височините на всяка клетка. Ще получите още WxH реални числа които представляват количеството вода което всяка клетка може да попие на стъпка от симулацията. Последното нещо на входа е реално число което е количеството вода с която ще започне всяка клетка в началото.

Пример за пропорционално оттичане:

Ако имаме клетка с 110 литра вода, с височина 10, 4 съседа с височини 20, 10, 7, 3, и попиване от 10 литра на стъпка. За една стъпка ще се случи следното:

1. 10 литра ще попият, и ще останат 100 литра
2. 2 от съседите са с по-ниска височина и вода ще прелее в тях
   1. 30 литра ще прелеят в съседа с височина 7
   2. 70 литра ще прелеят в съседа с височина 3

**Задача 3**

Имплементирайте функцията

char \*stringReplace(const char \*haystack, const char \*needle, const char \*replace);

Функцията замества всички срещания на needle в haystack с replace. Върнатият низ трябва да има памет с **точна дължина** за съдържанието си.

Пример:

stringReplace("the quick brown fox jumps over the brown fence", " ", "<SPACE>") -> "the<SPACE>quick<SPACE>brown<SPACE>fox<SPACE>jumps<SPACE>over<SPACE>the<SPACE>brown<SPACE>fence"

Покажете че имплементацията ви работи коректно.

**Задача 4**

Имате даден код: <https://gist.github.com/poseidon4o/5a22d1aee665c6b2b0494788ec52f929> Преработете кода, както намерите за добре за да го подобрите. В коментар опишете какво прави кода, как го постига