

Общи забележки:

- При четене на входа, приемерте, че потребителят въвежда данни в коректен формат и тип. Валидацията на стойностите им е ваша задача.
- Всички масиви трябва да са с точната големина (не по-големи от нужното).
- Разрешено е използването само на библиотеките **new** и **iostream**.

Задача 1

Напишете следните функции:

1. `input`, която въвежда от стандартния вход две цели числа `N` и `M`, след това създава в динамичната памет двумерен масив с `N` реда, `M` колони и елементи - цели числа и накрая прочита от стандартния вход елементите на масива.
2. `transform`, която въвежда цяло число `D` от стандартния вход и в подаден масив (създаден от горната функция) заменя всички елементи, които имат точно `D` делителя, с `0`.
3. `clear`, която премахва от масива, създаден чрез първата функция, всички редове и всички колони, съставени само от нулеви елементи, като премества оставащите на тяхно място и запазва относителния им ред.
4. `print`, която извежда на екрана двумерен масив.

Помислете какви трябва да са аргументите и връщаният резултат за всяка функция.

Демонстрирайте използването им в смислена цялостна програма.

Задача 2

В дните на засилена електронна комуникация е все по-важно да се използват надеждни средства за криптиране. Веско Синия, виден специалист по криптография, измислил следната процедура за криптиране – всяка буква (малка или главна латинска буква) се заменя с последователност от няколко (една или повече, но не повече от 10) букви според подадена таблица. Всички останали символи, както и буквите, които не са в таблицата, не се променят. При кодиране не се прави разлика между големи и малки букви.

Вашата задача е да му помогнете, като напишете функция, която по подаден символен низ и таблица със съответствия между букви и символни низове за кодиране според горния алгоритъм, създава динамично символен низ с криптираното съдържание. Помислете какъв е най-добрият начин да подадете данните и да получите резултата от тази функция.

Напишете също функция, която по подаден, криптиран според горния алгоритъм, символен низ и таблица на съответствията декриптира и връща в динамично заделен с точната големина низ оригиналното съобщение. Ако текстът не може коректно да се декриптира, да се върне празен резултат.

Напишете програма, която въвежда число **K**, следвано от последователност от **K** двойки символ (малка или главна латинска буква) и последователност от символи, с които този символ се криптира, двете разделени с интервал - по една такава двойка на ред.

Да се провери коректността на входа според условието.

След това трябва да се въвежда число **N**, следвано от **N** символни низа, по един на ред (всеки не по-дълъг от 1000 символа). За всеки да се изведе на екрана криптираният му вид.

Накрая се въвежда число **M**, следвано от **M** криптирани символни низа, по един на ред (всеки не по-дълъг от 1000 символа). За всеки да се изведе на екрана декриптираният му вид или текст “Bad string”, ако не може да се декриптира.

Пример:

Вход:

```
8
i ala
l bala
d cana
e h
t disco
o electro
c pro
a some
1
I l0ve to code!
1
ala somem discoelectroelectro proelectroelectrobala!
```

Изход:

```
ala bala0vh discoelectro proelectrocanah!
i am too cool!
```

Позволено е използването само на библиотеките `iostream` и `new`. Проверката на коректността на входа е ваша задача.

Задача 3

Напишете програма, която прочита от стандартния вход символен низ с произволна дължина. Низът може да е на много редове. Края ще разпознаете по символа #. След прочитането му трябва да:

- а) преброите и изведете колко са всички думи в текста;
- б) преброите и изведете колко са различните думи в текста;
- в) преброите по колко пъти се срещат различните думи в текста. Изведете на екрана различните думи и броя срещания за всяка, подредени по честота на срещане. Ако има две с еднаква честота, първо да е по-малката лексикографски.

Думите са последователност от малки и главни латински букви и тире '-', всички останали символи приемете за разделители. При броенето на думите не правете разлика между малки и главни букви.

Пример:

Вход:

```
Once upon a time there were three happy little pigs. The first little pig
played the flute, the second little pig played the violin and the third little
pig played the piano.#
```

Изход:

```
all words - 32
distinct words - 20
the - 6
little - 4
pig - 3
played - 3
a - 1
and - 1
first - 1
flute - 1
happy - 1
once - 1
piano - 1
pigs - 1
second - 1
there - 1
third - 1
three - 1
time - 1
upon - 1
violin - 1
were - 1
```