



Домашно 2

курс Увод в програмирането
за специалност Компютърни науки
зимен семестър 2021/2022 г

Важно: Решенията ви трябва да отговарят и на дадените по-долу изисквания:

1. Решете задачите на C++;
2. Решенията ви трябва да се изграждат успешно и да работят с поне един от компилаторите GCC 8.1 или MSVC 19 (Visual Studio 2019) или по-стари техни версии.
3. Ако не е посочено друго, в решението можете да използвате готови функционалности от езика, които са включени в заглавните файлове `<iostream>`, `<cmath>`, `<cstdint>`, `<cstring>` и `<limits>`.
4. Където не е посочено друго, трябва да осигурите коректни входни данни.
5. Типовете на използваните променливи трябва да са съобразени с условието и със семантиката на програмата, която реализирате.
6. Решенията на ВСИЧКИ задачи трябва да се предадат в един .cpp файл с име `group_fn_homework2.cpp`.
Например:
`group6_3MI0800383_homework2.cpp`
7. Работата с паметта е изцяло ваша отговорност
8. Оценката се получава по формулата $2 + \text{<брой точки>}$

Задача 0 (2,0 т.) Аня има проблем с правописа и често допуска грешки, пишейки документация на софтуера, който разработва. Затова ѝ е хрумнала добрата идея да напише софтуер, който да ги поправя вместо нея. Възможни автоматично поправими грешки са:

- пропуснат интервал около препинателен знак:
 - интервал се слага след всеки препинателен знак с изключение на отваряща скоба и отварящи кавички;
 - интервал се слага преди тире, аритметична операция, отваряща скоба и отварящи кавички;

- сгрешен правопис на дума или фраза;
 - за корекция на сгрешени думи или фрази, софтуерът използва помощен речник;
 - речникът представлява поредица от двойки от сгрешена дума **или фраза** и правилното ѝ изписване, разделени със знак за тире;
- изпуснат знак за край на изречението:
 - софтуерът определя дали е достигнат краят на изречението тогава и само тогава, когато следващата дума започва на нов ред или няма следваща дума;
 - приемаме, че всяко такова изречение би завършило с точка;
 - ако изречението завършва със затварящи кавички, точката се слага преди тях.

Да се напише функция `autoCorrect`, която по зададен текст за редакция и речник, както е описан по-горе, извежда на екрана редактирания текст. Функцията да приема параметри с подходящ тип. Да се реализира подходяща програма, която демонстрира работата на функцията като въвежда текст за редакция и речник и ги подава като параметри.

Допуска се използването на помощни функции.

Ограничения

- Максималният брой елементи в речника е 128.
- Максималната дължина на оригиналния текст е 1024 символа.
- Реалната дължина на речника и оригиналния текст се подават от клавиатурата и трябва да бъдат валидирани, че са в нужните граници.
- Записите в речника имат максимална дължина от 100 символа.
- Дължината на текста след извършените корекции не може да надвишава 2048 символа.

Пример:

Вход:

534

Once upon a time there lived in a certain village a little country girl, the pritiest creature who was ever seen. Her mother was excessively fond of her; and her grandmother doted on her still more. This good woman had a little red riding hood made for her. It suited the girl so extremely well that everybody called her little red riding hood.

One day her mother, having made some cakes, said to her, "Go, my dear, and see how your grandmother is doing, for I hear she has been very ill. Take her a cake, and this little pot of butter"

4

stil-still

pritiest-prettiest

little red riding hood-Little Red Riding Hood

leafes-leaves

Изход:

Once upon a time there lived in a certain village a little country girl, the prettiest creature who was ever seen. Her mother was excessively fond of her; and her grandmother doted on her

still more. This good woman had a Little Red Riding Hood made for her. It suited the girl so extremely well that everybody called her Little Red Riding Hood.

One day her mother, having made some cakes, said to her, "Go, my dear, and see how your grandmother is doing, for I hear she has been very ill. Take her a cake, and this little pot of butter."

Задача 1 (1,0 т.)

Да се напише функцията `transformMatrix`, която приема две целочислени матрици $A_{N_1 \times M_1}$ и $B_{N_2 \times M_2}$ и извършва следните преобразувания върху първата матрица:

- ако $N_2 < N_1$, то броят на редовете ѝ N_1 се намалява на N_2 ;
- ако $M_2 < M_1$, то броят на колоните ѝ M_1 се намалява на M_2 ;
- ако $A[i][j] = B[i][j]$, то $A[i][j]$ не се променя;
- ако $A[i][j] < B[i][j]$, то $A[i][j]$ приема стойността на $B[i][j]$;
- ако $A[i][j] > B[i][j]$, то $A[i][j]$ си сменя знака;

Функцията да приема подходящи параметри. Да се реализира примерна програма, която въвежда двете матрици и размерите им от клавиатурата и след извикване на функцията извежда на стандартния изход променената матрица A .

** Двете матрици трябва да бъдат динамично заделени, а направените промени действително да променят матрицата A , включително и да заделят нова памет с правилния размер, ако се налага промяна на броя редове или колони.*

Допуска се използването на помощни функции.

Пример

Примерът работи с две матрици със следните размери:

N1: 3

M1: 4

N2: 3

M2: 5

Вход:

A:

1	2	3	-4
10	19	2	0
-5	0	2	1

B:

1	-3	8	10	34
4	7	13	2	9
3	0	-4	5	2

Изход:

1	-2	8	10
-10	-19	13	2
3	0	-2	5

Задача 2: по избор

Всеки студент може да реши една от предложените задачи 2', 2'' или 2'''.

Задача 2' (1.0 т.)

Робот е разположен на правоъгълна дъска с размери $N \times M$ (N реда, M колони), разграфена на квадратчета. Редовете са номерирани от 0 до $N-1$ отгоре надолу, а колоните от 0 до $M-1$ отляво надясно. Всяко квадратче има написан в него символ – една от буквите 'L', 'R', 'D' или 'U', които обозначават посоката в която роботът следва да се придвижи попадайки на съответното квадратче – ляво, дясно, долу или горе, съответно.

Роботът може да се движи от текущото квадратче, на което се намира, до всяко от четирите съседни квадратчета (с обща страна).

Роботът може да започне движението си от произволна клетка. След това се придвижва към съседна клетка в посока, индикирана върху текущото квадратче, на една стъпка, при спазване на следните ограничения:

- Ако роботът се придвижи (излезе) извън ръба на дъската, той пада от нея и приключва своята работа.
- Ако роботът попадне в клетка, която вече е посетил, той спира да се движи и се самоизключва (приключва своята работа).

Роботът има поглед върху цялата дъска и написаните върху всяко квадратче символи и може да избере от кое квадратче да започне своето движение. Неговата цел е да извърши максимален брой стъпки (придвижвания от квадратче в квадратче) преди да падне от дъската или да се самоизключи.

Напишете програма, която определя от кое квадратче роботът трябва да започне своето движение, за да може да изпълни максимален брой команди (придвижвания). Командата се счита за успешно изпълнена, ако роботът се е придвижил от квадратчето, върху което е написана командата (без значение дали е отишъл на друго квадратче или е паднал от дъската).

Примерен вход:

```
4 4
RRRD
RUUD
URUD
UL LR
```

Изход: 3 2

Задача 2'' (1,0 т.) Езикът на една нова планетата се състои от думи, които се образуват по следните правила:

1. Празният низ е дума от езика;

2. Ако $\langle w \rangle$ е дума от езика, то $AA\langle w \rangle B$ също е дума от езика;
3. Ако $\langle w_1 \rangle$ и $\langle w_2 \rangle$ са думи от езика, то $\langle w_1 \rangle C \langle w_2 \rangle$ също е дума от езика;
4. Ако $\langle w \rangle$ е дума от езика, то $\langle w \rangle D \langle w \rangle$ също е дума от езика;
5. Ако $\langle w \rangle$ се състои от повече от 64 символа, то $\langle w \rangle$ не е дума от езика.

Да се напише програма, която отговаря на въпроси. От стандартния вход се въвежда цяло положително число n и n реда, на всеки от които има по един низ – поредната дума, която програмата трябва да провери дали е от гореописания език (празният низ се задава с празен ред). Програмата трябва да извежда на стандартния изход n реда, всеки от които съдържа по една цифра: 1, ако съответната на входа дума е от езика, или 0, в противен случай.

Ограничения

$$1 \leq n \leq 1024$$

Дължината никой от въведените низове не надвишава 256.

За решението на задачата използвайте функции.

Примери:

Вход	Изход
5 AABCAAB DD AABD CAAB DDD	1 0 0 1 1
10 AAAABB AACB CCCC CCDCCCAAB AABAAB CAABDAAB AABCDAAAB CXC AABAAAABB	1 1 1 1 0 1 0 1 0 0

Задача 2''' (1,0 т.)

Да се реализира функция sumK , която събира цели положителни числа с произволна дължина, записани в k -ична бройна система, където $2 \leq k \leq 36$. За цифри със стойности от 10 до 35 се използват съответно латинските букви от A до Z, като е допустимо използването както на главни, така и на малки букви. Функцията да приема като параметри двете числа m и n (зададени като последователности от цифри) и основата на бройната система k , в която са записани m и n . На стандартния изход да

се извежда сумата на двете числа, отново записана в **k**-ична бройна система. Ако поне едно от числата **m** или **n** не е валидно число в посочената система, да се изведе низът "Bad input data!". Да се реализира примерна програма, която демонстрира работата на функцията **sumK**.

Ограничения

$$2 \leq k \leq 36$$

m и **n** съдържат не повече от 1000 цифри

Примери:

Вход	Изход
250 340 10	590
110 11 2	1001
54 453 8	527
421 340 5	1311
645 342 6	Bad input data!
HELLO WORLD 25	Bad input data!
Hello World 36	1E3D71
ABCDEFGHIJKLM MLKJIHGFEDCBA 34	WWWWWWWWWWWWWW
ABCDEFGHIJKLM MLKJIHGFEDCBA 32	11111111111110