

Описание: Отваря се файла Input.txt прочита се първия ред и по броя на буквите се определя размера на матрицата. Вземат се редовете като стрингове и се слагат във лист. След това листа се преобразува до двумерен char масив. Останалите редове се слагат в друг лист, речник, от който се проверяват думите. След това се прави проверка за думите в матрицата и намерените думи се записват във файл output.txt. Проверката се извършва като се проверява дали всяка една дума е във матрицата. За всяка дума се взема първата ѝ буква и за всяко срещане в матрицата се вика рекурсивната функция findNeighbour, която проверява дали думата е част от матрицата.

struct Letter

Структ, който има две член данни – координати на буква в матрицата.

list<Letter> working;

Списък от координати на букви, на текущо проверявания път.

list<list<Letter>> final;

Списък съдържащ пътя на всички намерени думи.

string listToString(char matrix[8][8], list<Letter> otново)

По даден път на дума, създава стринг съдържащ думата.

void printResult(char matrix[8][8])

Записва намерените думи във файл output.txt.

void findNeighbour(char matrix[8][8], int msize, int i, int j, bool path[8][8], string word, int current)

Рекурсивна функция, която по дадени координати на буква от дума, функцията намира дали следващата буква в думата е част от матрицата. Прави се проверка дали е стигнат края на думата, ако е, листа със пътя ѝ се добавя в листа със намерени думи. Ако не е стигнат края се проверява дали следващата буква от думата е съседна на сегашната. Прави се проверка във всички посоки, като се проверява дали се излиза от матрицата и дали е минато вече през тази буква. Ако има валидна съседна буква, координатите ѝ се записват в листа working, променя се булевия масив, че вече е минато през тази буква и се вика същата функция с координатите на съседната буква и следващата буква от думата. След като се върне, трие старата буква от листа и се търси друга подходяща съседна буква.

void nullPath(bool path[8][8])

Прави всички стойности на булевия масив false.

void checkWord(char matrix[8][8], int msize, string word)

Намира всички срещания на първата буква от думата в матрицата и за всяко едно вика функцията findNeighbour със нейните координати.

void check(char matrix[8][8], int msize, list<string> dict)

Обхожда листа речник и подава всяка дума на функцията checkWord.