CPP Problem Design

Subject: Word game

Contributor: 謝公耀, 陳俊儒, 廖宣瑋

Main testing concept: File I/O

Basics	Functions
■ C++ BASICS	□ SEPARATE COMPILATION AND NAMESPACES
□ FLOW OF CONTROL	■ STREAMS AND FILE I/O
■ FUNCTION BASICS	■ RECURSION
□ PARAMETERS AND OVERLOADING	□ INHERITANCE
■ ARRAYS	□ POLYMORPHISM AND VIRTUAL FUNCTIONS
□ STRUCTURES AND CLASSES	□ TEMPLATES
□ CONSTRUCTORS AND OTHER TOOLS	□ LINKED DATA STRUCTURES
□ OPERATOR OVERLOADING, FRIENDS,AND REFERENCES	□ EXCEPTION HANDLING
■ STRINGS	□ STANDARD TEMPLATE LIBRARY
□ POINTERS AND DYNAMIC ARRAYS	□ PATTERNS AND UML

Description:

A popular word game involves finding words from a grid of randomly generated letters. Words must be at least three letters long and formed from adjoining letters. Letters may not be reused and it is valid to move across diagonals. As an example, consider the 4x4 grid of letters below.

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

The word "FAB" is valid (letters in the upper left corner) and the word "KNIFE" is valid. The word "BABE" is not valid because the "B" may not be reused. The word "MINE" is not valid because the "E" is not adjacent to the "N".

Write a program that uses a 4x4 two dimensional array to represent the game board. The program should read the words from the text file words.txt and then use a recursion algorithm to determine if the word may be formed from the letters on the game board. The program should output all valid words from the file that are on the game board.

Input:

See the Sample Input below, and note that all inputs are lower case letters.

Output:

See the Sample Output below, and please output to console.

Sample Input / Output：

Sample Input	Sample Output
abcd	abc

efgh	afb
ijkl	fie
mnop	fin
	fink
	glop
	ink
	jim
	knife
	lop
	min
	mink
	nim
	pkg
	plonk
	pol
	polk
aabc	abc
defg	abed
hijk	afb
llmn	baa
	baaed
	bad
	bade
	bead
	bed
	bedim
	dab
	deaf
	deb
	dei
	die
	dill
	dim
	fad
	fade
	fed
	fide
	fie
	fill
	film
	head
	heil
	hid
	hide
	hie
	hied
	hill
	him

	idea ill jill jim lid lie lied lief life lim limn mid mil mill

☐ Eazy,Only basic programming syntax and structure are required.
☐ Medium,Multiple programming grammars and structures are required.
■ Hard,Need to use multiple program structures or complex data types.

Expected solving time:
40 minutes

Other notes: