```
// $Id: countwords.cpp,v 1.1 2020-06-27 19:59:24-07 - - $

#include <cerrno>
#include <cstring>
#include <fstream>
#include <iostream>
#include <map>
#include <regex>
#include <string>
#include <vector>
using namespace std;

using wordcount_type = map<string,size_t>;

void scan (wordcount_type& words, istream& infile) {
   static const regex word_rx {"[[:alpha:]]+"};
   for (;;) {
      string line;
      getline (infile, line);
      if (infile.eof()) break;
      for (auto& chr: line) chr = tolower (chr);
      auto itor = sregex_iterator (line.begin(), line.end(), word_rx);
      for (; itor != sregex_iterator(); ++itor) {
         ++words[itor->str()];
      }
   }
}

int main (int argc, char** argv) {
   wordcount_type words;
   string exec_name {basename (argv[0])};
   int exit_status = EXIT_SUCCESS;
   vector<string> filenames (&argv[1], &argv[argc]);
   if (filenames.size() == 0) filenames.push_back ("-");
   for (const auto& filename: filenames) {
      if (filename == "-") scan (words, cin);
      else {
         ifstream infile (filename);
         if (infile) scan (words, infile);
         else {
            exit_status = EXIT_FAILURE;
            cerr << exec_name << ": " << filename << ": "
                 << strerror (errno) << endl;
         }
      }
   }
   for (const auto& word: words) {
      cout << word.first << " " << word.second << endl;
   }
   return exit_status;
}
```

```
# $Id: Makefile,v 1.1 2020-06-27 19:59:24-07 - - $


GARN    = -Wall -Wextra -Wpedantic -Wshadow -Wold-style-cast
GOPTS   = ${GWARN} -fdiagnostics-color=never
GPP     = g++ -std=gnu++2a -g -O0 ${GOPTS}
GRIND   = valgrind --leak-check=full --show-reachable=yes
NODEPS  = ${filter ci clean spotless tar, ${MAKECMDGOALS}}
MKTAR   = gtar --create --verbose --gzip

H_FILES =
C_FILES = countwords.cpp
OBJECTS = ${C_FILES:.cpp=.o}
EXECBIN = countwords
SOURCES = ${H_FILES} ${C_FILES} Makefile


all : ${EXECBIN}

${EXECBIN} : ${OBJECTS}
        ${GPP} -o $@ $^

%.o : %.cpp
        - checksource $<
        ${GPP} -c $<

ci : ${SOURCES}
        - checksource $^
        cid -is $^

clean :
        - rm --force ${OBJECTS} test.log test.out test.err

lis : ${SOURCES} Makefile.deps
        mkpspdf Listing.ps $^

spotless : clean
        - rm --force ${EXECBIN} Listing.{ps,pdf} Makefile.deps

tar : ${SOURCES}
        ${MAKE} --no-print-directory spotless
        ( DIRNAME=$$(basename $$(pwd)) \
        ; cd .. \
        ; ${MKTAR} --exclude=RCS --file=countwords.tar.gz $$DIRNAME \
        )

test : ${EXECBIN}
        ${GRIND} --log-file=test.log \
                 ${EXECBIN} ${SOURCES} 1>test.out 2>test.err

Makefile.deps :
        ${GPP} -MM ${C_FILES} >Makefile.deps

ifeq (${NODEPS}, )
include Makefile.deps
endif

```

```
countwords.o: countwords.cpp
```