

# SWE2004: Principles in Programming(Spring 2015)

## Programming Lab #2

Due-date: March 19, 14:45PM

Q&A: hhkim at csl.skku.edu

### Description

Andy, 8, has a dream - he wants to produce his very own dictionary. This is not an easy task for him, as the number of words that he knows is, well, not quite enough. Instead of thinking up all the words himself, he has a brilliant idea. From his bookshelf he would pick one of his favourite story books, from which he would copy out all the distinct words. By arranging the words in alphabetical order, he is done! Of course, it is a really time-consuming job, and this is where a computer program is helpful.

You are asked to write a program that lists all the different words in the input text. In this problem, a word is defined as a consecutive sequence of alphabets, in upper and/or lower case. Words with only one letter are also to be considered. Furthermore, your program must be CaSe InSeNsItIvE. For example, words like "Apple", "apple" or "APPLE" must be considered the same.

### Input

The input file is a text with no more than 5000 lines. An input line has at most 200 characters. Input is terminated by EOF.

### Output

Your output should give a list of different words that appears in the input text, one in a line. The words should all be in lower case, sorted in alphabetical order. You can be sure that the number of distinct words in the text does not exceed 5000.

## Sample Input

Adventures in Disneyland

Two blondes were going to Disneyland when they came to a fork in the road. The sign read: "Disneyland Left."

So they went home.

## Sample Output

a  
adventures  
blondes  
came  
disneyland  
fork  
going  
home  
in  
left  
read  
road  
sign  
so  
the  
they  
to  
two  
went  
were  
when

<ctype.h>

int tolower(int ch)
---------------------

int isalpha(int c)
--------------------

<string.h>

char *strtok(char *str, const char *delim)
--

char str[] = "hello world";
-----------------------------

char *ptr ;
-------------

ptr = strtok(str, " ");
-------------------------

printf("%s\n", ptr);
----------------------

ptr = strtok(str, " ");
-------------------------

printf("%s\n", ptr);
----------------------

hello
-------

world
-------

char* strcmp(const char *s1, const char *s2)
--

char* strcpy(char *dest, const char *src)
---

<stdlib.h>

Void qsort(void *base, size_t nitems, size_t size, int (*compar) (const void*, const void*))
--

base – the pointer to the first element of the array to be sorted
---

nitems – the number of elements in the array pointed by base
--

size – the size in bytes of each elements in the array
--

compar – the function that compares two elements
--