#include <stdio.h>

#include <stdlib.h>

#include <unistd.h> // To use read and write

#include <sys/stat.h>

unsigned int comparisons = 0;

int frobcmp(char const \* a, char const \* b){

comparisons++;

while (1){

if ( \*a == 32)

return 1;

if ( \*b == 32)

return -1;

if ( (\*a^42) > (\*b^42) )

return 1;

else if ( (\*a^42) < (\*b^42) )

return -1;

a++;

b++;

}

if ( \*a == 32 && \*b != 32)

return -1;

if ( \*a != 32 && \*b == 32)

return 1;

return 0;

}

int qsortwrap(const void \* a, const void \* b){

const char const\* word1 = \*(const char const\*\*) a;

const char const\* word2 = \*(const char const\*\*) b;

return frobcmp(word1, word2);

}

int main (void) {

int memsize = 100;

int i = 0;

char\* cin\_array;// = (char\*)malloc(sizeof(char)\*memsize);

int alpha = 0;

struct stat filestat;

fstat(0, &filestat);

int file\_size = filestat.st\_size;

if(S\_ISREG(filestat.st\_mode)) {

cin\_array = (char\*)malloc(sizeof(char)\*file\_size);

if(cin\_array == NULL) {

fprintf(stderr, "Error allocating memory.");

memsize = file\_size;

exit(1);

}

}

else{

cin\_array= (char\*)malloc(sizeof(char)\*memsize);

if(cin\_array == NULL) {

fprintf(stderr, "Error allocating memory.");

exit(1);

}

}

char character;

while (1) {

if (read(0, &character, 1) <= 0) {

if (cin\_array[i] != ' ' && i != 0){

cin\_array = (char\*)realloc(cin\_array, memsize + 1);

if ( cin\_array == NULL){

fprintf(stderr,"Memory allocation error");

exit(1);

}

cin\_array[i] = ' ';

i++;

}

break;

}

else{

cin\_array[i] = character;

if (cin\_array[i] != ' ')

{

alpha =1;

}

i++;

if (i == memsize){

cin\_array = (char\*)realloc(cin\_array, memsize \* 4);

memsize = memsize \* 4;

}

if (cin\_array == NULL){

fprintf(stderr,"Memory allocation error");

exit(1);

}

}

}//end of while

char\*\* word\_array = malloc(memsize);

int word\_count = 0;

for(int j = 0; j < i; j++){

if( j==0 || ( cin\_array[j-1] == ' ' && cin\_array[j] != ' ' ) ){

word\_array = realloc(word\_array, sizeof(char\*) \* (word\_count + 1));

if ( cin\_array == NULL){

fprintf(stderr,"Memory allocation error");

exit(1);

}

word\_array[word\_count] = &cin\_array[j];

word\_count++;

}

}

qsort(word\_array, word\_count, sizeof(char\*), qsortwrap);

if (alpha == 1) {

for ( int k = 0; k < word\_count; k++){

char \* fpointer = word\_array[k];

while ( \*fpointer != ' ' && \*fpointer != EOF){

write(1, (fpointer), 1);

fpointer++;

}

write(1, (fpointer), 1);

}

}

free(cin\_array);

free(word\_array);

fprintf(stderr, "Comparisons: %d\n", comparisons);

}//end of main