//C PROGRAM TO PRINT THE WORDS WITH SIMILAR LETTERS TOGETHER

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

// duplicating each word in the main using structure

struct Word

{

char\* str;

int id;

};

// representing the duplicate

struct DupArray

{

struct Word\* array;

int n; // Size of array

};

struct DupArray\* createDupArray(char\* str[], int n)

{

struct DupArray\* dupArray =

(struct DupArray\*) malloc( sizeof(struct DupArray) );

dupArray->n = n;

dupArray->array =

(struct Word\*) malloc( dupArray->n \* sizeof(struct Word) );

int i;

for (i = 0; i < n; ++i)

{

dupArray->array[i].id = i;

dupArray->array[i].str = (char\*) malloc( strlen(str[i]) + 1 );

strcpy(dupArray->array[i].str, str[i] );

}

return dupArray;

}

//for character compare

int compChar(const void\* a, const void\* b)

{

return \*(char\*)a - \*(char\*)b;

}

// for word comparing

int compStr(const void\* a, const void\* b)

{

struct Word\* a1 = (struct Word \*)a;

struct Word\* b1 = (struct Word \*)b;

return strcmp(a1->str, b1->str);

}

void printAnagramsTogether(char\* wordArr[], int n)

{

struct DupArray\* dupArray = createDupArray(wordArr, n);

int i;

for (i = 0; i < n; ++i)

qsort(dupArray->array[i].str,

strlen(dupArray->array[i].str), sizeof(char), compChar);

// sorting the array using qsort()

qsort(dupArray->array, n, sizeof(dupArray->array[0]), compStr);

for (i = 0; i < n; ++i)

printf("%s ", wordArr[dupArray->array[i].id]);

}

// The main program to test all the functionalities

int main()

{

char\* wordArr[] = {"how", "who", "paw", "awp", "wap", "here"};

int n = sizeof(wordArr) / sizeof(wordArr[0]);

printAnagramsTogether(wordArr, n);

return 0;

}