



#include <stdio.h>

#include <string.h>

#define PATH "/home/consta/practice1/1/Test1/3.txt"

#define BUFFER\_SIZE 256

void qsortRecursive(char \*\*mas, int size)

{

// Указатели в начало и в конец массива

int i = 0;

int j = size - 1;

// Центральный элемент массива

int mid = size / 2;

// Делим массив

do

{

// Пробегаем элементы, ищем те, которые нужно перекинуть в другую часть

// В левой части массива пропускаем(оставляем на месте) элементы, которые меньше центрального

while (strcmp(mas[i], mas[mid]) < 0)

{

i++;

}

// В правой части пропускаем элементы, которые больше центрального

while (strcmp(mas[j], mas[mid]) > 0)

{

j--;

}

// Меняем элементы местами

if (i <= j)

{

char \*tmp = mas[i]; // тут mas[i]

mas[i] = mas[j]; // теперь тут mas[j]

mas[j] = tmp; // теперь тут mas[i]

// в mas[i] - mas[j], а в mas[j] - mas[i]

i++;

j--;

}

} while (i <= j);

// Рекурсивные вызовы, если осталось, что сортировать

if (j > 0)

{

//"Левый кусок"

qsortRecursive(mas, j + 1);

}

if (i < size)

{

//"Правый кусок"

qsortRecursive(&mas[i], size - i);

}

}

int main(int argc, char \*argv[])

{

FILE \*file;

const char \*fname = argv[1];

if(argc != 2)

{

return 0;

}

file = fopen(fname, "r");

if (file == NULL)

{

printf("не могу открыть файл для чтения '%s'", fname);

return 0;

}

int size = 0;

char\*\* file\_content = new char\*;

char line\_from\_file[BUFFER\_SIZE]{0};

while (fgets(line\_from\_file, sizeof(line\_from\_file), file))

{

file\_content[size] = new char[BUFFER\_SIZE];

strcpy(file\_content[size], line\_from\_file);

size += 1;

}

qsortRecursive(file\_content, size);

file = fopen(fname, "wb");

if (file == NULL)

{

printf("не могу открыть файл '%s'", fname);

return 0;

}

for (int line\_num = 0; line\_num < size; line\_num += 1) {

char\* line = file\_content[line\_num];

fwrite(line, sizeof(char), strnlen(line, BUFFER\_SIZE), file);

delete[] line;

}

fclose(file);

}