

Write the program, invoked in a following way:

```
./sort infile outfile
```

infile is a text file, containing several lines, some of them can be empty. Read in the file, sort the lines lexicographically and write them to the file named **outfile**. The contents of **infile** is not changed. Maximum line length neither maximum number of lines are not defined, so fixed-size line buffer is not acceptable. You will get 0% for the implementation using fixed-size line buffer.

Do not seek in the input file, as it can be a special file, e.g. **/dev/tty**.

Programs generating errors or warnings when compiled with **-Wall -pedantic**, generating warnings when run with **valgrind**, not working or dumping core will not be graded.

Hint:

Maximum line length is not defined, so you will need to use dynamic memory allocation. Define the function

```
char* getline(FILE* infile)
```

reading entire line from **infile** character by character using **fgetc** and storing them in a dynamically allocated array. Expand the array as needed. Return the pointer to the beginning of the line or **NULL**, if end of file has been encountered before any character has been read. Beware of the final line not ending with **'\n'**.

Example:

file1 contains the following:

```
That Katharina and Petruchio should be married,  
And yet we hear not of our son-in-law.  
What will be said? what mockery will it be,  
To want the bridegroom when the priest attends  
To speak the ceremonial rites of marriage!  
What says Lucentio to this shame of ours?
```

after invoking

```
./sort file1 file2
```

file2 contains:

```
And yet we hear not of our son-in-law.  
That Katharina and Petruchio should be married,  
To speak the ceremonial rites of marriage!  
To want the bridegroom when the priest attends  
What says Lucentio to this shame of ours?  
What will be said? what mockery will it be,
```