

Написана программа, принимающая на вход текст. Она выводит все предложения в порядке неубывания длин ровно по одному на строку.

Решена дополнительная задача : вывод в файл *NINE.txt* перевернутого самого длинного предложения из текста.

При сборке проекта возможно использование трех флагов:

- `--tofile` (вывод результата в текстовый файл, его название будет запрошено)
- `--fromfile` (ввод данных из текстового файла, название будет запрошено)
- `--dop` (решение дополнительной задачи)

Возможно использование любой комбинации этих флагов.

Команды для работы с проектом:

<code>make</code>	сборка проекта
<code>make mytest</code>	сборка тестов
<code>make clean</code>	удаление всех объектных файлов
<code>make dist-clean</code>	удаление всех объектных файлов, тестов и текстовых файлов, созданных в процессе запуска программы
<code>./MyProg --tofile --fromfile --dop</code>	запуск с нужными флагами
<code>./mytest</code>	запуск тестов

Главная функция `int main(int argc, char* argv[])`; обрабатывает флаги, по необходимости просит пользователя ввести названия используемых файлов.

```
int main(int argc, char* argv[]){  
  
    char file_to_read[100];  
    char file_to_write[100];  
    int flag_dop = 0;  
  
    for(int i = 1; i<argc; i++){  
        if (argv[i][0]=='-' && argv[i][1]=='-' && argv[i][2]=='f' && argv[i][3]=='r' && argv[i][4]=='o' &&
```

```

argv[i][5]=='m' && argv[i][6]=='f' && argv[i][7]=='i' && argv[i][8]=='l' && argv[i][9]=='e') {
    std::cout << "Eter the file to read name" << std::endl;
    int name_len=0;
    char bukv;
    while (std::cin.get(bukv)) {
        if (bukv == '\n') {
            break;
        }
        file_to_read[name_len] = bukv;
        name_len++;
    }
}

if (argv[i][0]=='-' && argv[i][1]=='-' && argv[i][2]=='t' && argv[i][3]=='o' && argv[i][4]=='f' &&
argv[i][5]=='i' && argv[i][6]=='l' && argv[i][7]=='e') {
    std::cout << "Eter the file to write name" << std::endl;
    int name_len=0;
    char bukv;
    while (std::cin.get(bukv)) {
        if (bukv == '\n') {
            break;
        }
        file_to_write[name_len] = bukv;
        name_len++;
    }
}

if (argv[i][0]=='-' && argv[i][1]=='-' && argv[i][2]=='d' && argv[i][3]=='o' && argv[i][4]=='p') {
    flag_dop=1;
}

}

char *mas = prog(file_to_read);
if(file_to_write[0]) clear(file_to_write);
else std::cout<<"Answer: " <<std::endl;
int i = 0;
while(mas[i]){
    if(file_to_write[0]){
        std::ofstream wrfile(file_to_write, std::ios::app);
        wrfile << mas[i];
        wrfile.close();
    }
    else std::cout<<mas[i];
    i++;
}

delete[] mas;

if (flag_dop==1) {
    std::cout<<std::endl;
    reversemax();
}

return 0;
}

```

Далее она вызывает функцию `prog(char* file_to_read);` , которая, в свою очередь, благодаря переданному значению обрабатывает ввод, заполняя двумерный массив типов *char* и массив длин строк. После эта функция сортирует подмассивы (строки), переписывает их в новый массив типов *char* определенной длины и возвращает его.

Прописана функция `void clear(char* file_to_clear);` , очищающая текстовый файл, название которого в нее передается.

```
void clear(char* file_to_clear) {  
    std::ofstream file(file_to_clear);  
    file<<"";  
}
```

Функция `reversemax();` выполняет переворот самого длинного предложения из введенного текста и выводит его в файл под названием *NINE.txt*. Она вызывается в `int main();`.