

ОТЧЕТ по лабораторной работе№5

Что делает программа?

Программа в зависимости от значения флага записывает отсортированные по длине предложения в файл `out.txt`. Самое большое предложение записывается в `NINE.txt` в перевернутом виде, не меняя слова.

Форматы ввода

```
| flag | correct/incorrect | Error |
| --tofile | cottect | - |
| --fromfile "filename.txt" | cottect | - |
| --fromfile "filename.txt" --tofile | cottect | - |
| else flags | incorrect | WRONG!! ENTER |
| nothing | incorrect | Nothing was entered |
```

О реализации

Если введен флаг `--tofile` программа будет ожидать ввода от пользователя до тех пор, пока не будет введен '/' и затем приступит к действию. Если введен флаг `--fromfile` и название файла, который надо обработать, и приступит к действию, пока не достигнет конца файла.

Сама реализация основана на посимвольном вводе из файла или из консоли:

```
35 c = cin.peek();
...
72 c = fin.get();
```

Программа всегда на каждой итерации превышает длины строки и их количества:

```

7 const int STR_MAX_LEN = 1024;
8 const int STR_MAX_QUANTITY = 100;
...
30 if (counter == STR_MAX_LEN || k == STR_MAX_QUANTITY)
31 {
32 out << "BUFFER OVERFLOW!";
33 return -1;
34 }
...
67 if (counter == STR_MAX_LEN || k == STR_MAX_QUANTITY)
68 {
69 out << "BUFFER OVERFLOW!";
70 return -1;
71 }

```

Далее идет обработка и результат записывается массив строк:

```

53 std::copy(str, str + counter+1, a[k]);
...
90 std::copy(str, str + counter+1, a[k]);

```

Вовремя ввода учитывается, что после точки обязательно должен стоять пробел, перенос строки или завершения файла, иначе бан:

```

36 if (flag == 1 && c != '\n' && c != ' ' && c != '/') {
37 cout << "Error of entry!";
38 return -1;
39 }
...
73 if (flag == 1 && c != '\n' && c != ' ' && c != '/') {
74 cout << "Error of entry!";
75 return -1;
76 }

```

Сортировка массива строк осуществляется сортировкой пузырьком, где 'id[i]' - порядковый номер предложения, 'mas[i]' - длина предложения:

```
103   for (int i = 0; i < k; i++)
104       {
105           id[i] = i;
106       }
107   for(int i = 0; i < k-1; i++)
108       for(int j = 0; j < k - i - 1; j++)
109           {
110               if(mas[j] > mas[j+1])
111                   {
112
113                       t1 = mas[j];
114                       mas[j] = mas[j + 1];
115                       mas[j + 1] = t1;
116                       t2 = id[j];
117                       id[j] = id [j+1];
118                       id[j+1] = t2;
119                   }
120       }
```

Затем предложения записываются в `out.txt` при помощи отсортированного массива их номеров, а последняя строка в перевернутом виде записывается в `NINE.txt` :

```
121     std::ofstream fout("out.txt");
122     char nine[STR_MAX_QUANTITY][STR_MAX_LEN];
123     int sun = 0;
124     int j =0;
125     int q = 0;
126     for (int i = 0; i< k; i++)
127     {
128         fout<< a[id[i]] << endl;
129         if (i == (k-1))
130         {
131             while(a[id[i]][j]!='.')
132             {
133                 if(a[id[i]][j] != ' ' && a[id[i]][j] != '.'){
134                     nine[sun][q] = a[id[i]][j];
135                     q+=1;
136                 }
137                 else {sun+=1; q=0;}
138                 j+=1;
139             }
140         }
141     }
142 }
143 }
144 fout.close();
145 std::ofstream out("NINE.txt");
146 for (int q=sun; q>=0;q--)
147 {
148     out << nine[q];
149     if (q == 0) out << '.';
150     else out << ' ';
151 }
152 }
153 out.close();
```

Преимущества и ошибки

- Если вы введете '\n', то данному символу будет присвоен пробел
 - Если после точки у вас символ, отличный от пробела, переноса строки или '/', то выведет ошибку
 - Флаги записываются в строгом порядке
 - У массивов строк и массива символов есть свои ограничения
 - '/' - не эпишется, как символ
-