**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра Вычислительной техники**

**Курсовая работа**

**по дисциплине «Программирование»**

**Тема: Обработка текстовой информации**

|  |  |  |
| --- | --- | --- |
| Студент гр. 3312 |  | Шарапов И. Д. |
| Преподаватель |  | Аббас С. А. |

Санкт-Петербург

2023

**Содержание**

[Цель работы 3](#_Toc152880982)

[Задание (Вариант 51) 3](#_Toc152880983)

[Постановка задачи и описание решения 3](#_Toc152880984)

[Описание переменных 4](#_Toc152880985)

[Схема алгоритма 4](#_Toc152880986)

[Текст программы 5](#_Toc152880987)

[Формат ввода 7](#_Toc152880988)

[Контрольные примеры 7](#_Toc152880989)

[Содержимое файлов 8](#_Toc152880990)

[Примеры выполнения программы 10](#_Toc152880991)

[Выводы 11](#_Toc152880992)

# Цель работы

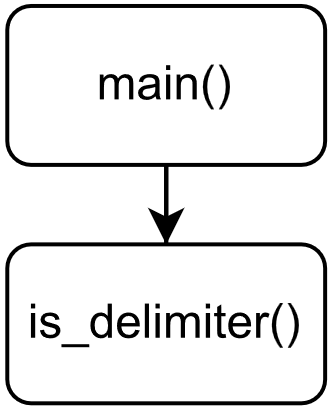
Законченное поэтапное решение содержательной задачи (постановка задачи, спецификация, выбор структур данных и разработка алгоритма, программная реализация, тестирование).

# Задание (Вариант 51)

Ввести заданное количество ключевых слова и строку символов-разделителей. Затем вводится текст с неизвестным количеством строк. Ввод текста заканчивается, если после ввода строки в тексте окажется в любой последовательности все ключевые слова. Из строк введённого текста, в которых встречается хотя бы одно ключевое слово, удалить слово, имеющее минимальную длину. Вывести преобразованный текст.

# Постановка задачи и описание решения

# Структура вызова функций



# Описание переменных

|  |  |  |  |
| --- | --- | --- | --- |
| Функция *int is\_delimiter(char c, char \*delimiters)* | | | |
| № | Имя переменной | Тип | Назначение |
| 1 | c | char |  |
| 2 | delimiters | char\* |  |
| 3 | x | char\* |  |
| 4 | ans | int |  |
| Функция *int main()* | | | |
| № | Имя переменной | Тип | Назначение |
| 1 | r | int |  |
| 2 | t | int |  |
| 3 | b | int |  |
| 4 | type\_of\_input | char |  |
| 5 | filename | char[] |  |
| 6 | f | file |  |
| 7 | cnt\_of\_keywords | int |  |
| 8 | keyword | char[] |  |
| 9 | x | char\* |  |
| 10 | hash\_of\_word1 | long long |  |
| 11 | hash\_of\_word2 | long long |  |
| 12 | keysh | int[][] |  |
| 13 | delimiters | char[] |  |
| 14 | flag | int |  |
| 15 | lines | int |  |
| 16 | text | char[][] |  |
| 17 | start | char\* |  |
| 18 | min\_line | int |  |
| 19 | cnt\_words\_bef | int |  |
| 20 | cnt\_words\_now | int |  |
| 21 | min\_word | char\*[][] |  |
| 22 | type\_of\_output | char |  |

# **Схема алгоритма**

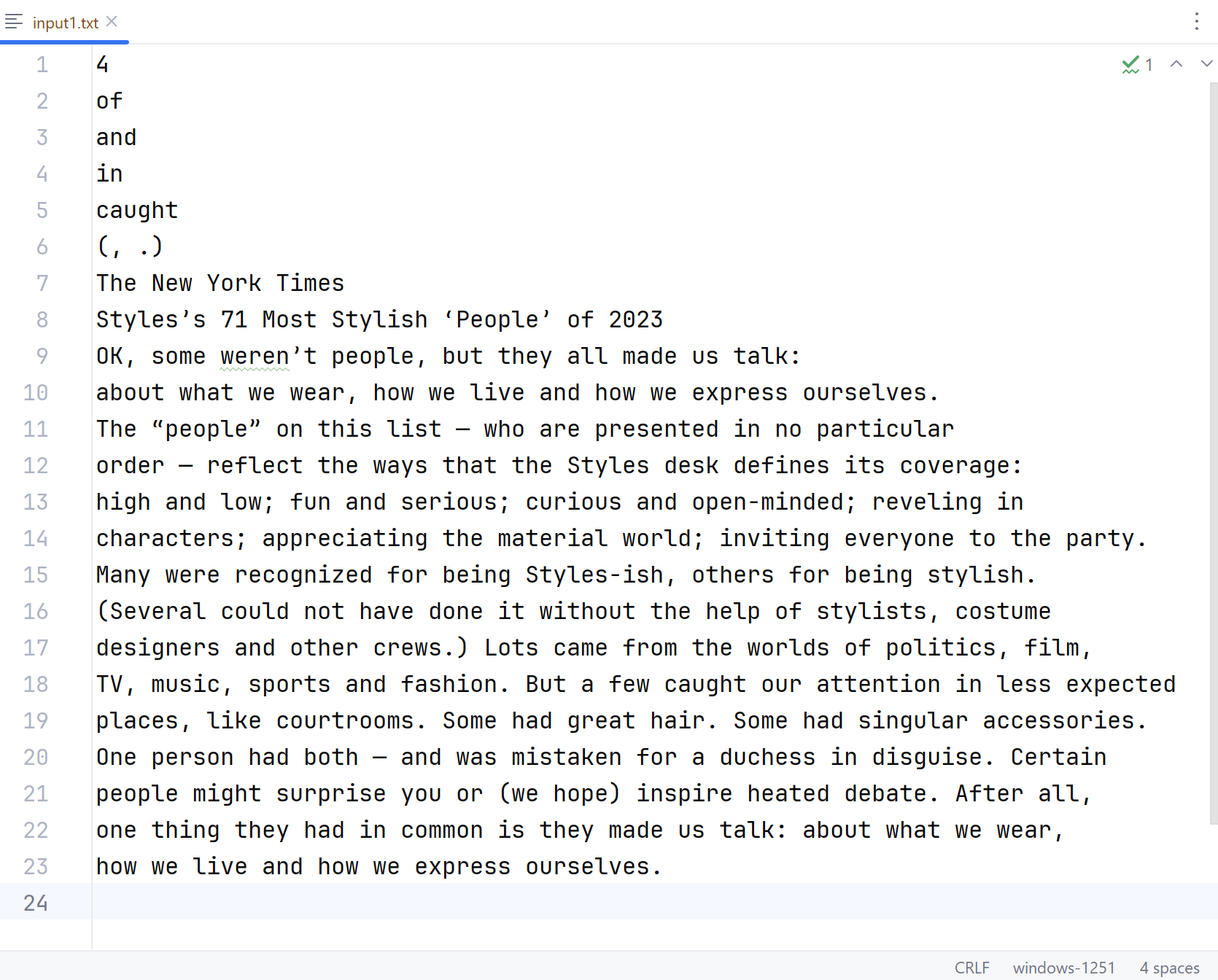
# Текст программы

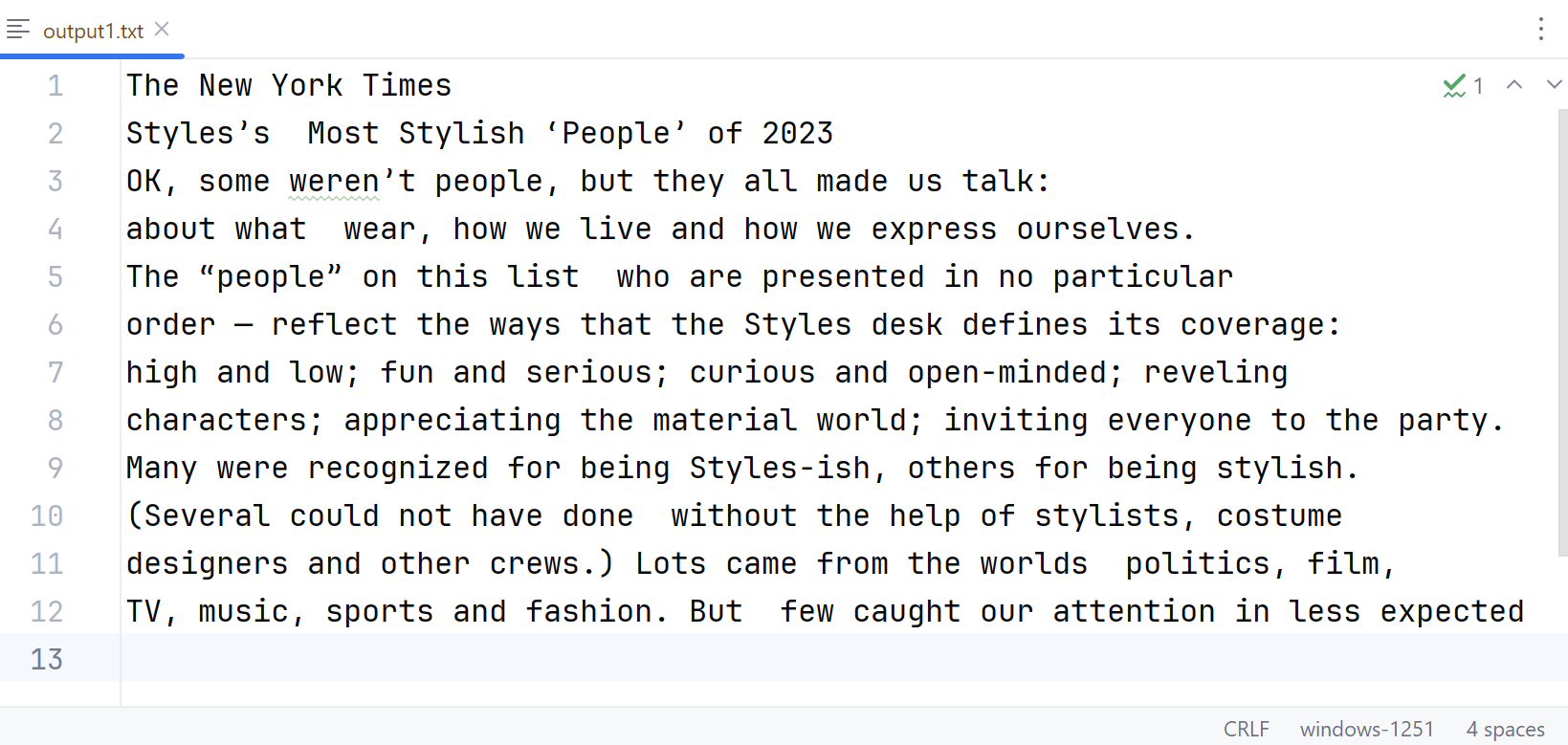
|  |
| --- |
| **#include <stdio.h>  #define HASH\_MOD 1000000009 *// hash module* #define HASH\_B1 43 *// hash base 1* #define HASH\_B2 47 *// hash base 2* #define MAXLEN\_N 128 *// file name* #define MAXLEN\_K 128 *// keyword len* #define MAXKEYS 256 *// keywords* #define MAXLEN\_D 64 *// delimiters* #define MAXLEN\_S 512 *// text line* #define MAXLINES 512 *// lines* int is\_delimiter(char c, char \*delimiters);  int main() {  *// Block of variables* int r;  int t;  int b;  char type\_of\_input;  char filename[MAXLEN\_N];  FILE \*f;  int cnt\_of\_keywords;  char keyword[MAXLEN\_K];  char \*x;  long long hash\_of\_word1;  long long hash\_of\_word2;  int keysh[MAXKEYS][3];  char delimiters[MAXLEN\_D];  int flag;  int lines;  char text[MAXLINES][MAXLEN\_S];  char \*start;  int min\_line;  int cnt\_words\_bef; *// count of found words before* int cnt\_words\_now; *//* char \*min\_word[MAXLINES][2];  char type\_of\_output;   *// Block of initialization* r = HASH\_MOD;  t = HASH\_B1;  b = HASH\_B2;  flag = 1;  lines = 0;  cnt\_words\_bef = 0;   *// Block of input* printf("Input from file or console? (f/c)\n");  type\_of\_input = getchar();  while (type\_of\_input != 'f' && type\_of\_input != 'c') {  printf("Something went wrong! Please enter \'f\' or \'c\':\n");  type\_of\_input = getchar();  }  if (type\_of\_input == 'f') {  printf("Please enter the file name (limit: %i chars):\n", MAXLEN\_N);  scanf("%s", filename);  f = fopen(filename, "r");  while (f == NULL) {  printf("Something went wrong! Perhaps such a file does not exist."  "\nPlease enter the file name again:\n");  scanf("%s", filename);  f = fopen(filename, "r");  }  } else {  f = stdin;  }   if (type\_of\_input == 'c') printf("Please enter the count of keywords (limit: %i):\n", MAXKEYS);  fscanf(f, "%i", &cnt\_of\_keywords);    if (type\_of\_input == 'c') printf("Please enter the keywords (limit: %i chars):\n", MAXLEN\_K);  for (int i = 0; i < cnt\_of\_keywords; ++i) {  fscanf(f, "%s", keyword);  x = keyword;  hash\_of\_word1 = 0;  hash\_of\_word2 = 0;  while (\*x != '\0') {  hash\_of\_word1 = (hash\_of\_word1 \* t + (int) \*x) % r;  hash\_of\_word2 = (hash\_of\_word2 \* b + (int) \*x) % r;  ++x;  }  keysh[i][0] = hash\_of\_word1;  keysh[i][1] = hash\_of\_word2;  keysh[i][2] = 0;  }   if (type\_of\_input == 'c') printf("Please enter the line of delimiter characters (limit: %i chars):\n", MAXLEN\_D);  fgets(delimiters, MAXLEN\_D, f);  fgets(delimiters, MAXLEN\_D, f);   if (type\_of\_input == 'c')  printf("Please enter the lines of text (limit of line\'s length: %i; limit\n"  "count of lines: %i):\n", MAXLEN\_S, MAXLINES);   *// Block of main logic* while (flag == 1 && fgets(text[lines], MAXLEN\_S, f)) {  x = text[lines];  start = text[lines];  hash\_of\_word1 = 0;  hash\_of\_word2 = 0;  min\_line = MAXLEN\_S;  while (\*x != '\0') {  if (is\_delimiter(\*x, delimiters) == 1) {  for (int i = 0; i < cnt\_of\_keywords; ++i) {  if (hash\_of\_word1 == keysh[i][0] && hash\_of\_word2 == keysh[i][1]) {  ++keysh[i][2];  }  }  if (x - start != 0 && min\_line > x - start) {  min\_line = x - start;  min\_word[lines][0] = start;  min\_word[lines][1] = x;  }  start = x + 1;  hash\_of\_word1 = 0;  hash\_of\_word2 = 0;  } else {  hash\_of\_word1 = (hash\_of\_word1 \* t + (int) \*x) % r;  hash\_of\_word2 = (hash\_of\_word2 \* b + (int) \*x) % r;  }  ++x;  }  flag = 0;  cnt\_words\_now = 0;  for (int i = 0; i < cnt\_of\_keywords; ++i) {  if (keysh[i][2] == 0) flag = 1;  cnt\_words\_now += keysh[i][2];  }  if (cnt\_words\_now == cnt\_words\_bef) {  min\_word[lines][0] = text[lines];  min\_word[lines][1] = text[lines];  }  cnt\_words\_bef = cnt\_words\_now;  lines++;  }  if (type\_of\_input == 'f') fclose(f);   *// Block of output* printf("Output to file or console? (f/c)\n");  if (type\_of\_input == 'f') getchar();  type\_of\_output = getchar();  while (type\_of\_output != 'f' && type\_of\_output != 'c') {  printf("Something went wrong! Please enter \'f\' or \'c\':\n");  type\_of\_output = getchar();  }  if (type\_of\_output == 'f') {  printf("Please enter the file name (limit: %i chars):\n", MAXLEN\_N);  scanf("%s", filename);  f = fopen(filename, "w");  while (f == NULL) {  printf("Something went wrong! Please enter the file name again:\n");  scanf("%s", filename);  f = fopen(filename, "w");  }  } else {  printf("Processed text:\n");  f = stdout;  }   for (int i = 0; i < lines; ++i) {  x = text[i];  while (\*x != '\0') {  if (x == min\_word[i][0]) {  x = min\_word[i][1];  min\_word[i][0] = NULL;  }  fprintf(f, "%c", \*x);  ++x;  }  }  if (type\_of\_output == 'f') {  printf("DONE");  fclose(f);  }  return 0; }  int is\_delimiter(char c, char \*delimiters) {  char \*x;  int ans = 0;  x = delimiters;  while (\*x != '\0') {  if (c == \*x) ans = 1;  ++x;  }  return ans; }** |

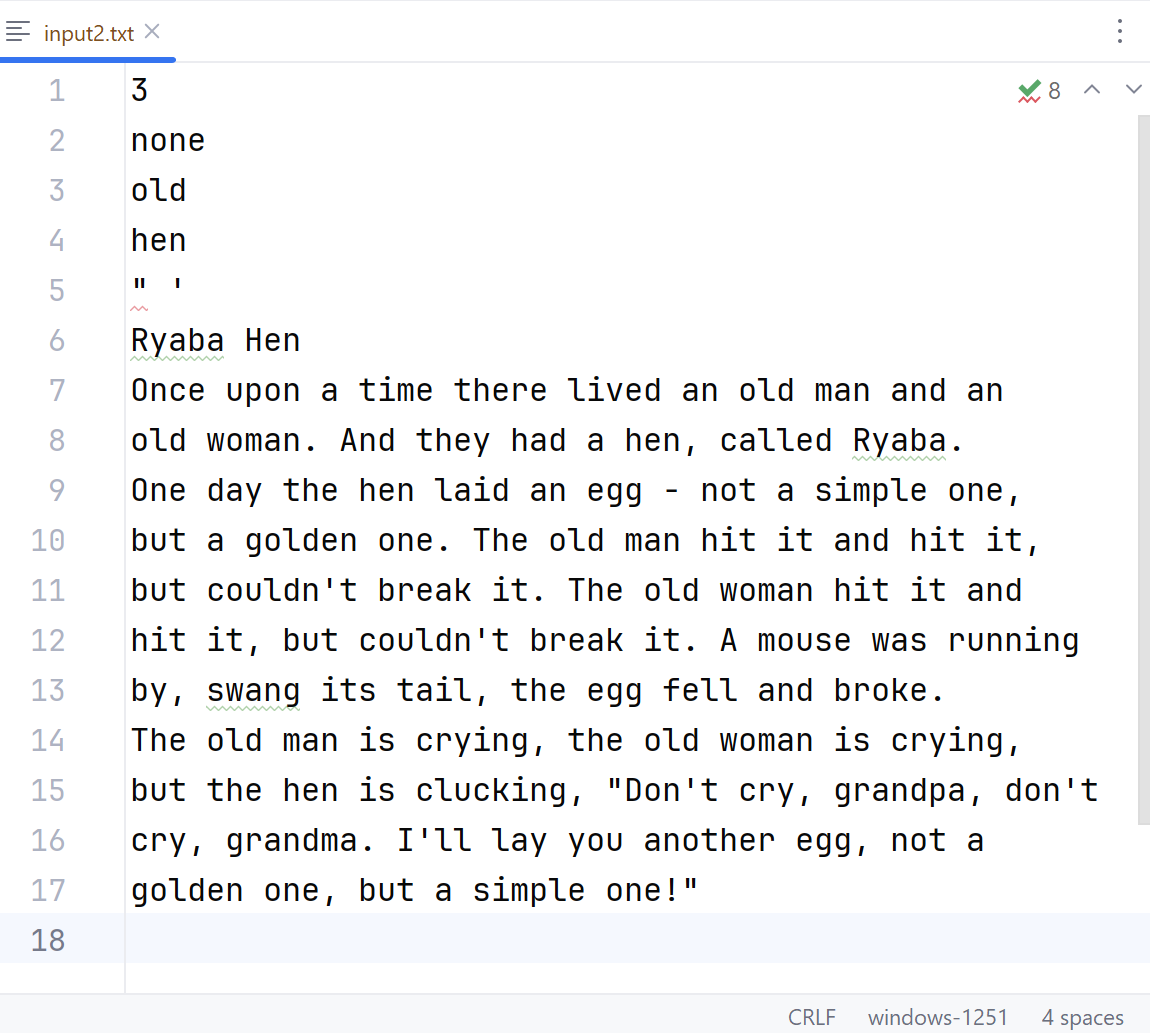
# Контрольные примеры

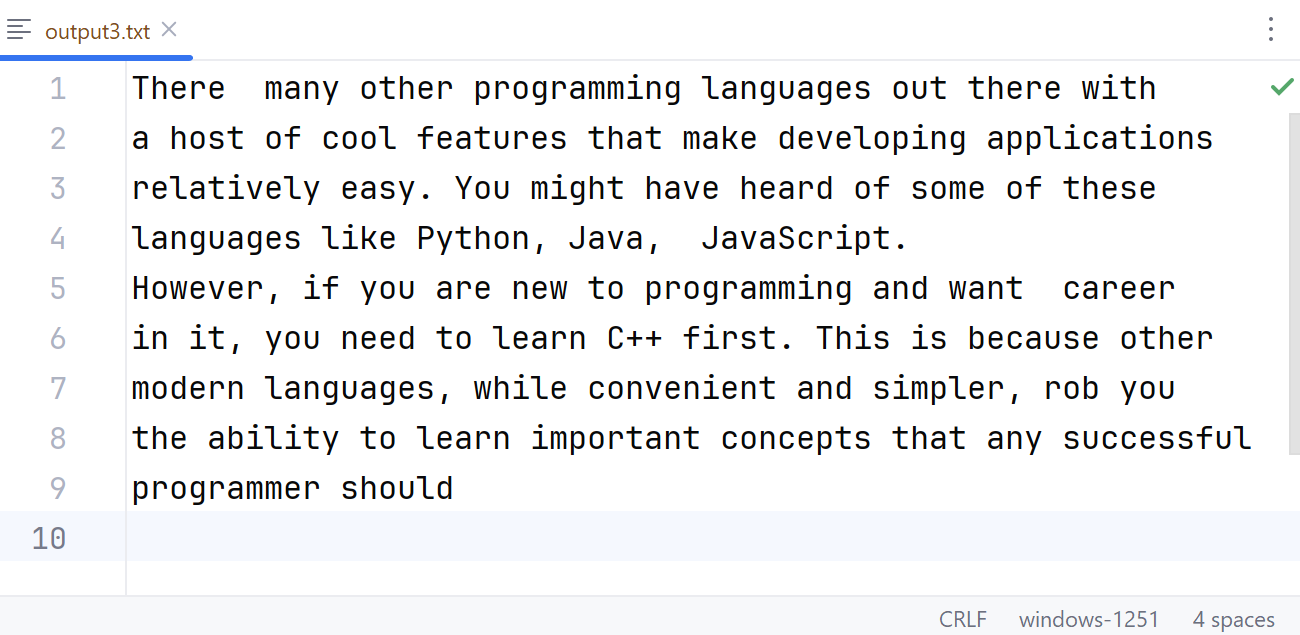
|  |  |  |
| --- | --- | --- |
| № | Исходные данные | Результаты |
| 1 | f  input1.txt  f  output1.txt | DONE |
| 2 | f  input2.txt  c | Processed text:  Ryaba Hen  Once upon time there lived an old man and an  old woman. And they had hen, called Ryaba.  One day the hen laid an egg not a simple one,  but golden one. The old man hit it and hit it,  but couldn' break it. The old woman hit it and  hit it, but couldn't break it. A mouse was running  by, swang its tail, the egg fell and broke.  The old man crying, the old woman is crying,  but the hen is clucking, "Don' cry, grandpa, don't  cry, grandma. I'll lay you another egg, not a  golden one, but a simple one!" |
| 3 | c  4  are  simpler,  programmer  Java,    There are many other programming languages out there with  a host of cool features that make developing applications  relatively easy. You might have heard of some of these  languages like Python, Java, and JavaScript.  However, if you are new to programming and want a career  in it, you need to learn C++ first. This is because other  modern languages, while convenient and simpler, rob you of  the ability to learn important concepts that any successful  programmer should know.  f  output3.txt | DONE |

# Содержимое файлов

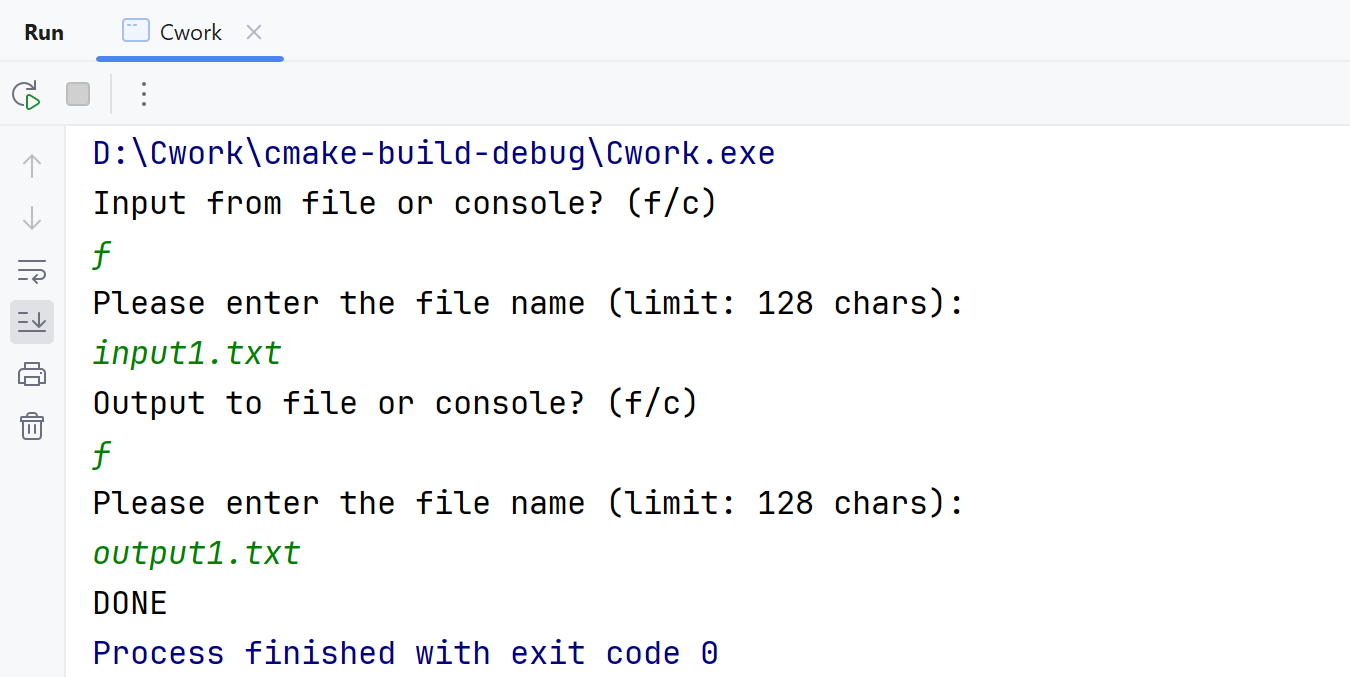


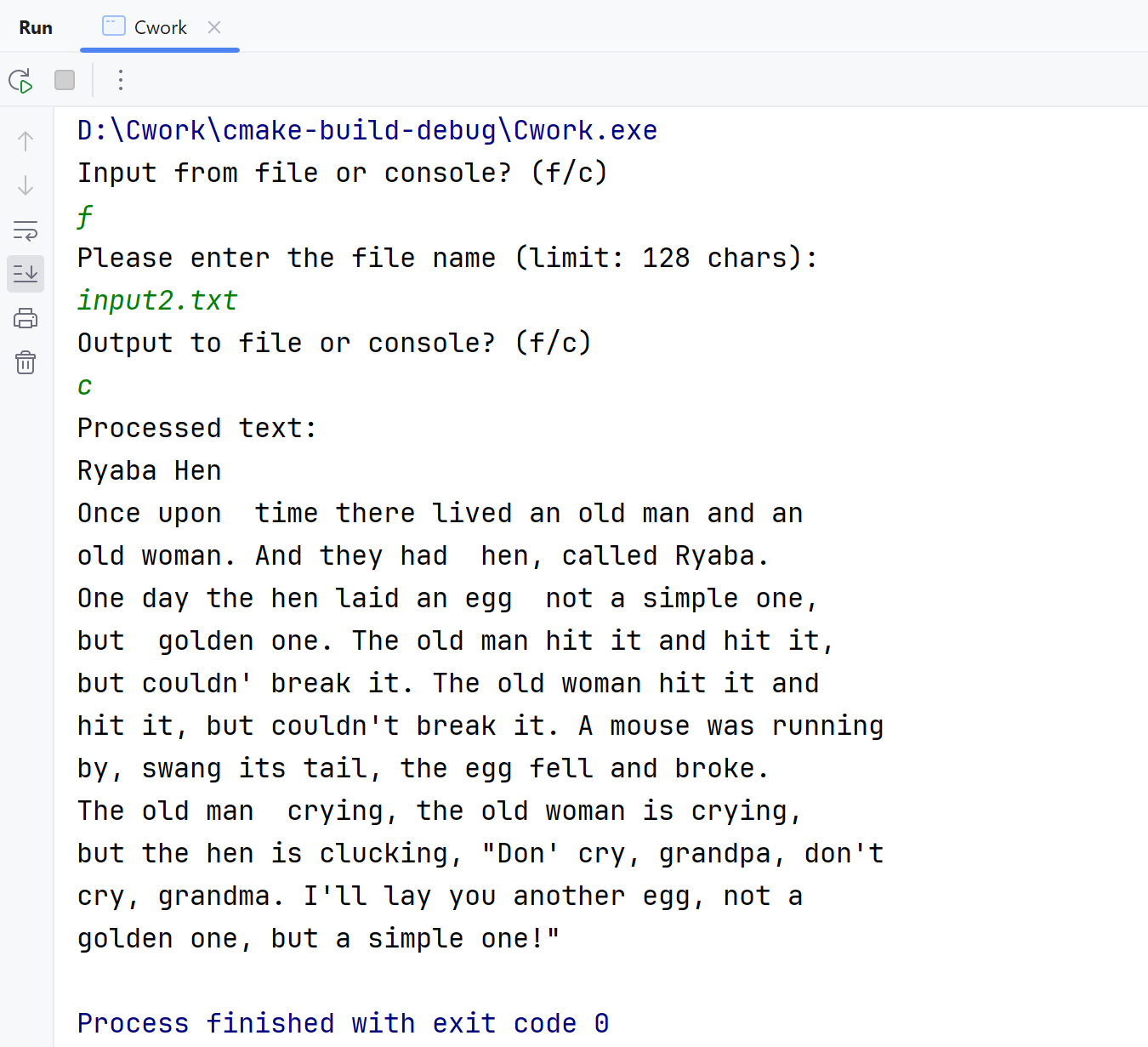


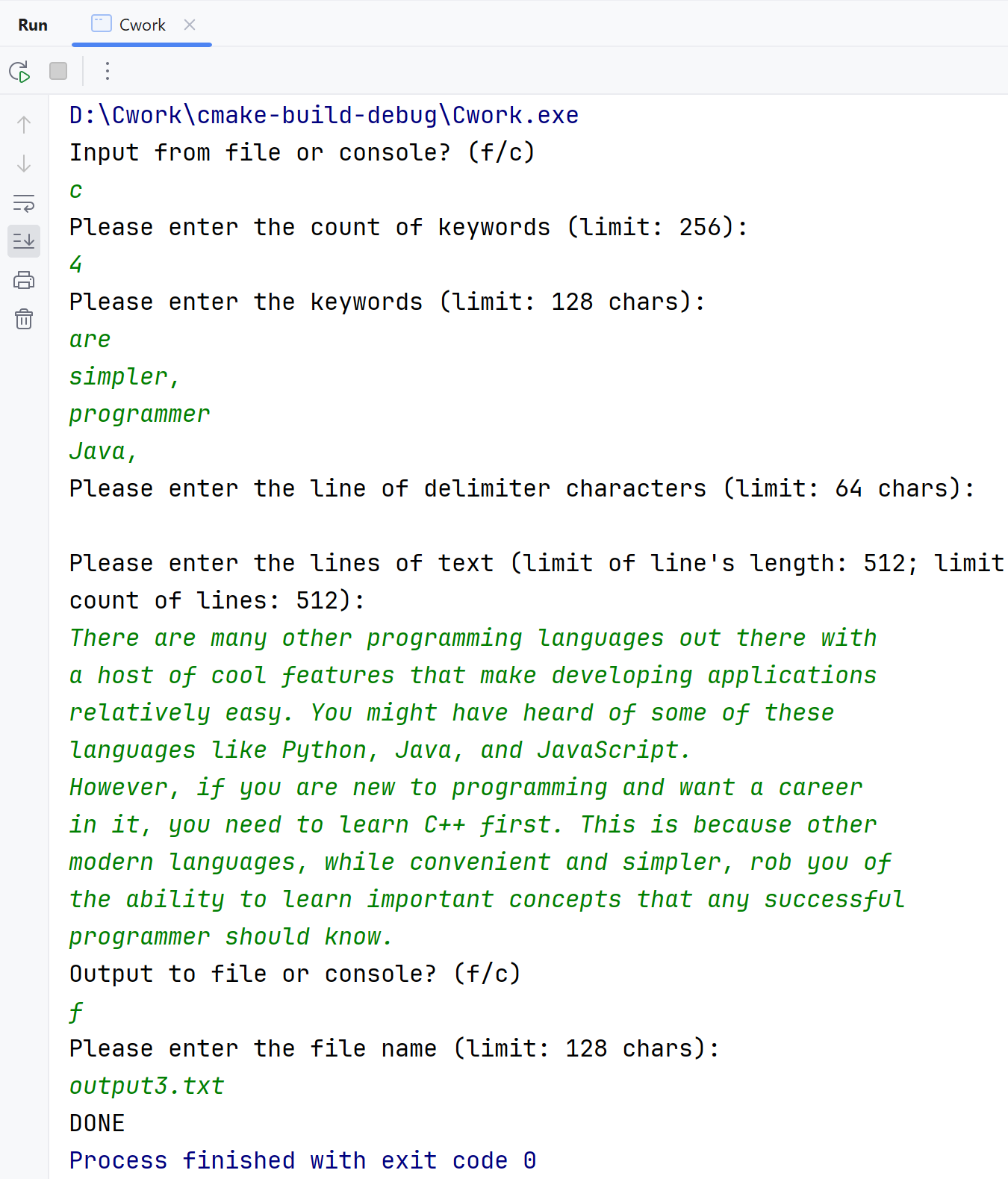




# Примеры выполнения программы







# Выводы

В работе использован только один заголовочный файл стандартной библиотеки. *<stdio.h>* используется для ввода и вывода из файла и консоли.