**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР**

отчет

**по лабораторной работе №2**

**по дисциплине «Программирование»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 1302 |  | Гордиенко М. Ю. |
| Преподаватель |  | Калмычков В.А. |

Санкт-Петербург

2022

Оглавление

[Исходная формулировка 3](#_Toc103721854)

[Контрольный пример 3](#_Toc103721855)

[Формальная постановка задачи 3](#_Toc103721856)

[Формат хранения данных 3](#_Toc103721857)

[Ограничения, обусловленные компьютером 4](#_Toc103721858)

[Макет ввода/вывода 4](#_Toc103721859)

[Алгоритм решения 4](#_Toc103721860)

[Пользовательский тип данных 5](#_Toc103721861)

[Текст программы 5](#_Toc103721862)

[Результат работы программы 7](#_Toc103721863)

[Вывод 7](#_Toc103721864)

# Исходная формулировка

Найти все предложения с отступами и посчитать длину между ними.

# Контрольный пример

w!

fewfrfer

fefrrefererf.

dwedewf

frefe

frefe

g.

f

w!

fewfrfer

fefrrefererf.

dwedewf

frefe

frefe

g.

f!

Ответ:

Расстояние между 1 и 2 - 1

Расстояние между 2 и 3 - 3

Расстояние между 3 и 4 - 0

Расстояние между 4 и 5 - 2

Расстояние между 5 и 6 - 3

Расстояние между 6 и 7 - 0

Всего отступов: 7

# Формальная постановка задачи

Дано: текст

Найти: отступы и расстояние между ними

# Формат хранения данных

|  |  |  |
| --- | --- | --- |
| Идентификатор типа | Тип | Назначение |
| f, f1 | fstream | Файловые переменные |
| stroka | char | Строки |
| j, i, count, sum | int | Счётчик |
| T[M] | str | Пользовательский тип |
| mark | char | Маркер |
| k, a, n, b | int | Дополнительные переменные |

# Ограничения, обусловленные компьютером

Целый тип int имеет размер 4 байта (32 бита). Минимальное значение -2 147 483 648, максимальное значение 2 147 483 647.

Символьный тип char. Минимальное значение -128, максимальное значение 128.

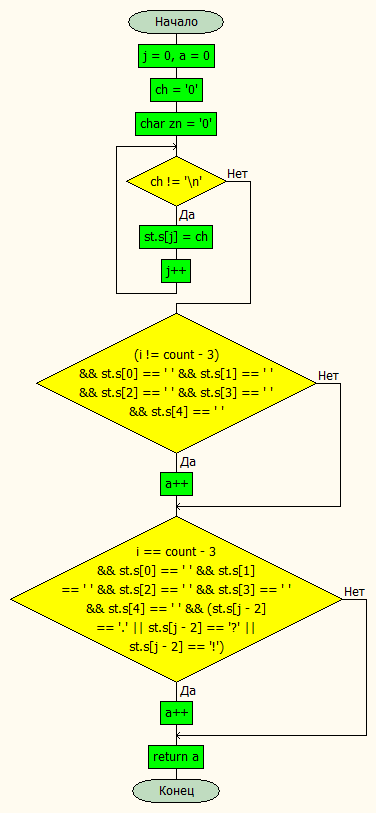
# Макет ввода/вывода

|  |  |
| --- | --- |
| Ввод из файла | f>> |
| Вывод из файла | f1<< |

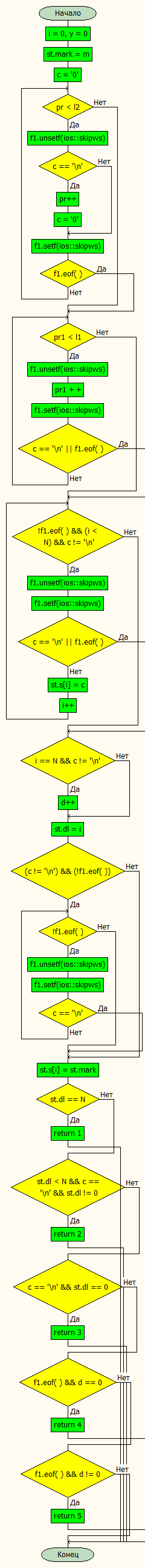
# Алгоритм решения

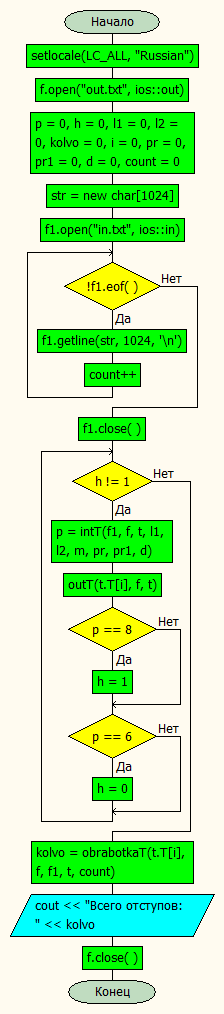
main()

obrabotkaStr()



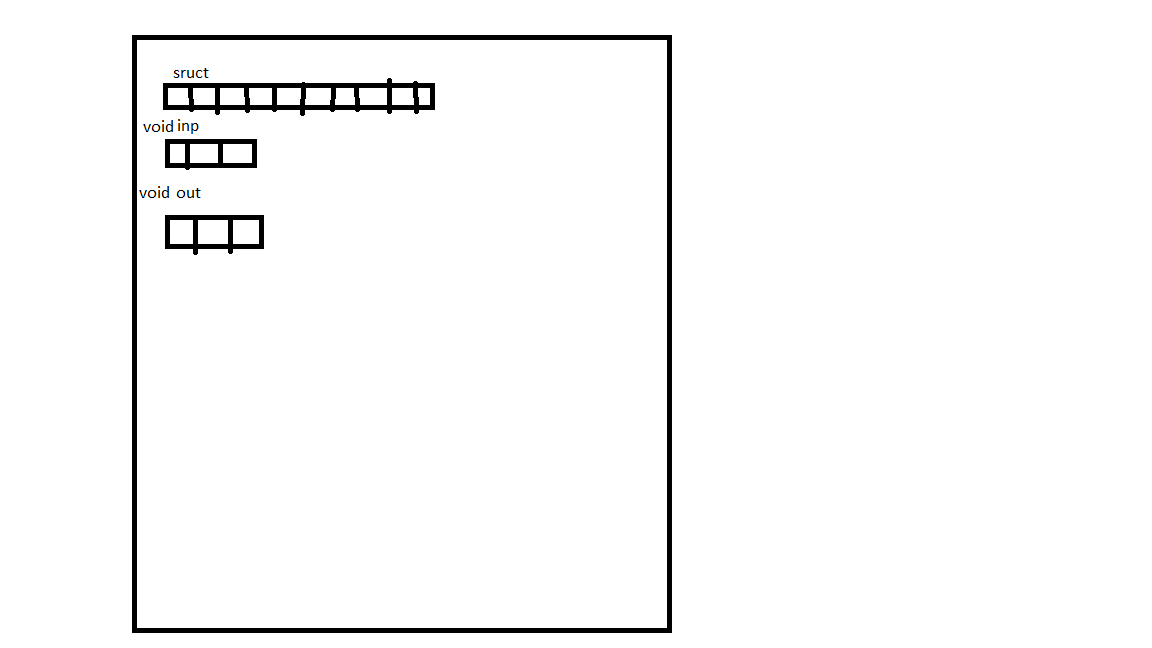
inpStr()



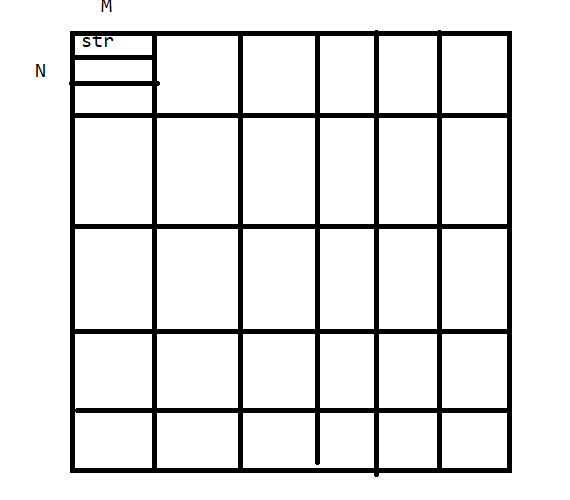


# Пользовательский тип данных

Представление строки:



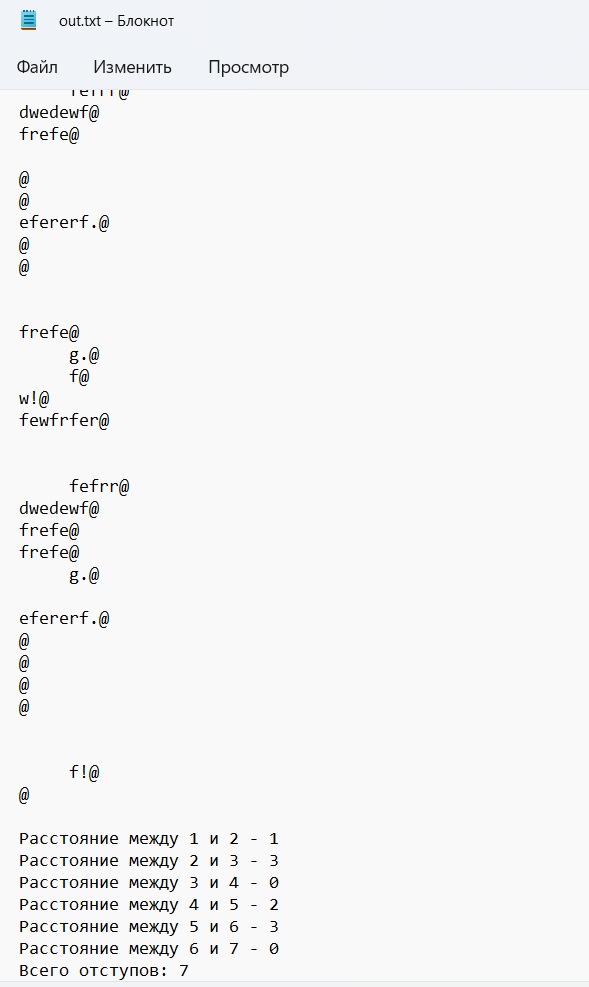
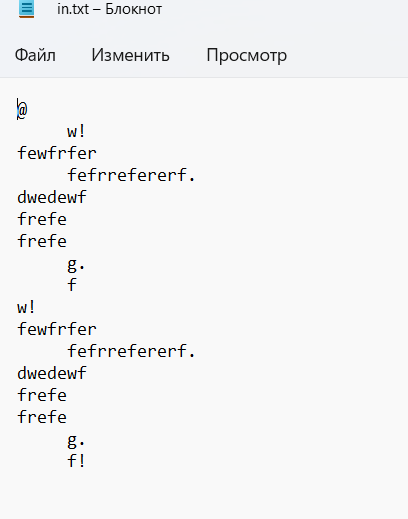
Представление текста:



# Текст программы

|  |  |  |
| --- | --- | --- |
| main() | inpT() | inpStr() |
| #include <iostream>  #include <iomanip>  #include <fstream>  #include "stroka.h"  #include "text.h"  #include "functions.h"  using namespace std;  int main()  {  setlocale(LC\_ALL, "Russian");  fstream f, f1;  f.open("out.txt", ios::out);  f << "Гордиенко Михаил; \n";  f << "группа 1302; \n";  Text t;  int p = 0, h = 0, l1 = 0, l2 = 0, kolvo = 0, i = 0, pr = 0, pr1 = 0, d = 0, count = 0;  char m;  char\* str = new char[1024];  f1.open("in.txt", ios::in);  while (!f1.eof())  {  f1.getline(str, 1024, '\n');  count++;  }  f1.close();  f << "Введённый текст: \n";  while (h != 1)  {  p = intT(f1, f, t, l1, l2, m, pr, pr1, d);  outT(t.T[i], f, t);  if (p == 8)  h = 1;  if (p == 6)  h = 0;  }  kolvo = obrabotkaT(t.T[i], f, f1, t, count);  f << "Всего отступов: " << kolvo;  f.close();  } | char r;  f1.open("in.txt", ios::in);  f1.unsetf(ios::skipws);  f1 >> m;  f1 >> r;  f.setf(ios::skipws);  int p = 0, k = 0, j1 = 0, i = 0;  pr = 0;  pr1 = 0;  d = 0;  while (i < M)  {  p = inStr(t.T[i], f, f1, l1, l2, m, pr, pr1, d);  i++;  if (p == 4)  {  k = M + 1;  break;  }  if (p == 3)  j1++;  pr1 = 0;  }  t.L = i;  l1 = l1 + N;  if (j1 == t.L)  {  l2 = l2 + t.L;  l1 = 0;  t.L = 0;  }  f1.close();  if (k == M + 1)  return 8;  if (k == 0)  return 6;  } | #include <iostream>  #include <iomanip>  #include <fstream>  #include "functions.h"  using namespace std;  unsigned inStr(str& st, fstream& f, fstream& f1, int& l1, int& l2, char& m, int& pr, int& pr1, int& d)  {  char c;  int i = 0, y = 0;  st.mark = m;  c = '0';    while (pr < l2)  {  f1.unsetf(ios::skipws);  f1 >> c;  if (c == '\n')  {  pr++;  c = '0';  }  f1.setf(ios::skipws);  if (f1.eof())  break;  }  while (pr1 < l1)  {  f1.unsetf(ios::skipws);  f1 >> c;  pr1++;  f1.setf(ios::skipws);  if (c == '\n' || f1.eof())  {  break;  }  }  while (!f1.eof() && (i < N) && c != '\n')  {  f1.unsetf(ios::skipws);  f1 >> c;  f1.setf(ios::skipws);  if (c == '\n' || f1.eof())  {  break;  }  st.s[i] = c;  i++;  }  if (i == N && c != '\n')  d++;  st.dl = i;  if ((c != '\n') && (!f1.eof()))  while (!f1.eof())  {  f1.unsetf(ios::skipws);  f1 >> c;  f1.setf(ios::skipws);  if (c == '\n')  break;  }  st.s[i] = st.mark;  if (st.dl == N)  return 1;  if (st.dl < N && c == '\n' && st.dl != 0)  return 2;  if (c == '\n' && st.dl == 0)  return 3;  if (f1.eof() && d == 0)  return 4;  if (f1.eof() && d != 0)  return 5;  } |
| outT() | outStr() | obrabotkaStr() |
| #include <iostream>  #include <fstream>  #include "functions.h"  using namespace std;  void outT(str& st, fstream& f, Text& t)  {  unsigned i;  for (i = 0; i < t.L; i++)  outStr(t.T[i], f, t);  f << endl;  } | #include <iostream>  #include <fstream>  #include "functions.h"  using namespace std;  void outStr(str& st, fstream& f, Text t)  {  int j;  for (j = 0; j <= st.dl; j++)  f << st.s[j];  f << endl;  } | #include <iostream>  #include <fstream>  #include "functions.h"  using namespace std;  unsigned obrabotkaStr(str& st, fstream& f1, Text& t, unsigned i, int count)  {  int j = 0, a = 0;  char ch = '0';  static char zn = '0';  while (ch != '\n')  {  f1.unsetf(ios::skipws);  f1 >> ch;  st.s[j] = ch;  f1.setf(ios::skipws);  j++;  }  if ((i != count - 3) && st.s[0] == ' ' && st.s[1] == ' ' && st.s[2] == ' ' && st.s[3] == ' ' && st.s[4] == ' ')  {  a++;  }  if (i == count - 3 && st.s[0] == ' ' && st.s[1] == ' ' && st.s[2] == ' ' && st.s[3] == ' ' && st.s[4] == ' ' && (st.s[j - 2] == '.' || st.s[j - 2] == '?' || st.s[j - 2] == '!'))  {  a++;  }  return a;  } |

# Результат работы программы



# Вывод

В результате выполнения работы научился разбивать текст по блокам, и улучшил свои навыки по работе со структурами и классами.