**Министерство общего и профессионального образования Российской Федерации**

**Уральский государственный университет**

**Кафедра Вычислительной техники**

**Программирование на языке С++**

**Лабораторный практикум**

**Вариант №5**

**Преподаватель: Томашевич В.Г.**

**Студент: Молодых А.А.**

**Группа: Фт-330801**

**Екатеринбург, 2015**

Оглавление

[1. Лабораторная работа №1. «Знакомство с C++. Выполнение программы простой структуры» 5](#_Toc435481414)

[1.1. Постановка задачи 5](#_Toc435481415)

[1.2. Программа решения задания 1 5](#_Toc435481416)

[1.3. Результаты работы программы для данных типа float 5](#_Toc435481417)

[1.4. Объяснение результатов 5](#_Toc435481418)

[1.5. Программа решения задания 2 5](#_Toc435481419)

[1.6. Результаты работы программы 5](#_Toc435481420)

[1.7. Объяснение результатов 5](#_Toc435481421)

[2. Лабораторная работа №2. «Использование основных операторов языка Си» 5](#_Toc435481422)

[2.1. Постановка задачи 5](#_Toc435481423)

[2.2. Программа решения задания 6](#_Toc435481424)

[2.3. Результат решения 6](#_Toc435481425)

[3. Лабораторная работа №3. «Вычисление функций с использованием их разложения в степенной ряд» 7](#_Toc435481426)

[3.1. Постановка задачи 7](#_Toc435481427)

[3.2. Программа решения задания 7](#_Toc435481428)

[3.3. Полученные результаты 9](#_Toc435481429)

[4. Лабораторная работа №4. «Работа с одномерными массивами» 9](#_Toc435481430)

[4.1. Постановка задачи 9](#_Toc435481431)

[4.2. Программа решения задания 9](#_Toc435481432)

[4.3. Программа решения задания 14](#_Toc435481433)

[5. Лабораторная работа №5. «Функции и массивы» 15](#_Toc435481434)

[5.1. Постановка задачи 15](#_Toc435481435)

[5.2. Программа решения задания 15](#_Toc435481436)

[5.3. Результат решения 17](#_Toc435481437)

[6. Лабораторная работа № 6. «Строки» 17](#_Toc435481438)

[6.1. Постановка задачи 17](#_Toc435481439)

[6.2. Программа решения задания 17](#_Toc435481440)

[6.3. Результаты выполнения программы 20](#_Toc435481441)

[7. Лабораторная работа № 7.1. «Перегрузка функций в Си++» 20](#_Toc435481442)

[7.1. Постановка задачи 20](#_Toc435481443)

[7.2. Программа решения задания 20](#_Toc435481444)

[7.3. Программа решения задания 22](#_Toc435481445)

# Лабораторная работа №1. «Знакомство с C++. Выполнение программы простой структуры»

## Постановка задачи

2. Вычислить значение выражения при различных вещественных типах данных (float и double). Вычисления следует выполнять с использованием промежуточных переменных. Сравнить и объяснить полученные результаты.

3. Вычислить значения выражений. Объяснить полученные результаты.

|  |  |  |
| --- | --- | --- |
| № | Задание 1 | Задание 2 |
| 5 | ,  при а=1000, b=0.0001 | 1. - -m-++n 2. m\*n<n++ 3. n-- > m++ |

## Программа решения задания 1

## Результаты работы программы для данных типа float

## Объяснение результатов

## Программа решения задания 2

## Результаты работы программы

## Объяснение результатов

# Лабораторная работа №2. «Использование основных операторов языка Си»

## Постановка задачи

Используя оператор цикла, найти сумму элементов, указанных в конкретном варианте. Результат напечатать, снабдив соответствующим заголовком.

Найти сумму ряда с точностью ε=10-4, общий член которого



## Программа решения задания

1. // Александр Молодых, ФТ-330801
2. // Лабораторная работа №2. Использование основных операторов языка Си
3. // Вариант - 5
4. #include <iostream>
5. #include <math.h>
6. #include <locale>
7. **using** **namespace** std;
8. #define EPS 0.0001
9. //Найти сумму ряда с точностью epsilon=10-4, общий член которого
10. //an = ((-1)^(n-1)) / (n^n)
11. **int** **main**()
12. {
13. setlocale(LC\_ALL, "Russian");
14. **double** summand = **0**, //слагаемое
15. diff, //разность
16. res = **0**; //результат
17. **int** n = **1**; //счётчик
18. cout << "Лабраторная работа 2: Использование основных операторов языка Си" << endl;
19. cout << "----------------------------------------------" << endl;
20. **do**
21. {
22. diff = summand;
23. n % **2** ? res -= summand : res +=summand;
24. summand = **1**/pow(**double**(n),**double**(n));
25. n++;
26. cout << abs(diff - summand) << endl;
28. } **while** (abs(diff - summand) > EPS); //пока значение n-го слагаемого больше точности
29. cout << "Результат вычислений: " << res << endl;
30. cout << "----------------------------------------------" << endl;
31. cout << "Итераций выполнено: " << n << endl;
32. **return** **0**;
33. }

## Результат решения

|  |
| --- |
| X:\\_URFU\TECHPROG\lab2\ex1\Release>ex1  Лабораторная работа 2: Использование основных операторов языка Си  ----------------------------------------------  1  0.75  0.212963  0.0331308  0.00358625  0.000298567  2.02192e-005  Результат вычислений: 0.783429  ----------------------------------------------  Итераций выполнено: 8 |

Рис 2. Результат решения задания лабораторной работы №2.

# Лабораторная работа №3. «Вычисление функций с использованием их разложения в степенной ряд»

## Постановка задачи

Для х, изменяющегося от a до b с шагом (b-a)/k, где (k=10), вычислить функцию f(x), используя ее разложение в степенной ряд в двух случаях:

а) для заданного n;

б) для заданной точности (=0.0001).

Для сравнения найти точное значение функции.

|  |  |  |  |
| --- | --- | --- | --- |
| Функция | Диапазон изменения аргумента | N | Сумма |
|  |  | 15 |  |

## Программа решения задания

1. // Александр Молодых, ФТ-330801
2. // Лабораторная работа №3. Вычисление функций с использованием их разложения в степенной ряд
3. // Вариант - 5
4. #include <iostream>
5. #include <iomanip> //для setw()
6. #include <locale> //для кириллицы
7. #include <math.h> //для pow() и abs()
8. **using** **namespace** std;
9. #define EPS 0.0001
10. #define N 15
11. #define k 10
12. **double** **calcExpAriphm**(**double**);
13. **double** **calcExpIter**(**double**);
14. /\*Для х изменяющегося от a до b с шагом (b-a)/k, где (k=10), вычислить функцию f(x),
15. используя ее разложение в степенной ряд в двух случаях:
16. а) для заданного n;
17. б) для заданной точности epsilon (epsilon=0.0001).
18. Для сравнения найти точное значение функции.
19. \*/
20. **int** **main**()
21. {
22. setlocale(LC\_ALL,"Russian");
23. **double** a = **1**;
24. **double** b = **2**;
25. cout << "Лабораторная работа 3: Вычисление функций с использованием их разложения в степенной ряд" << endl;
26. cout << "----------------------------------------------" << endl;
27. cout << "Результат вычислений:" << endl;
29. **for** (**double** x = a; x <= b + **0.001**; x += (b-a)/k) //b+0.001 связано с ошибкой при округлении
30. {
31. cout << "X=" << setw(**3**) << x << " SN=" << setw(**8**) << calcExpAriphm(x) << " SE=" << setw(**8**) << calcExpIter(x) << " Y=" << setw(**8**) << exp(x) <<endl;
32. }
33. cout << "----------------------------------------------" << endl;
34. **return** **0**;
35. }
36. **double** **calcExpAriphm**(**double** x)
37. {
38. **int** i = **0**; //счётчик
39. **double** resAriphm = **0**; //результат арифметического цикла
40. **double** summand = **1**; //первый член ряда, указан в условии
41. //Цикл арифметический:
42. **do**
43. {
44. resAriphm += summand;
45. summand \*= x / (i + **1**); //рекуррентное соотношение
46. i++;
47. } **while** (i < N);
48. **return** resAriphm;
49. }
50. **double** **calcExpIter**(**double** x)
51. {
52. **int** i = **0**; //счётчик
53. **double** resIter = **0**; //результат итерационного цикла
54. **double** summand = **1**; //первый член ряда, указан в условии
55. //Цикл итерационный:
56. **do**
57. {
58. resIter += summand;
59. summand \*= x / (i + **1**); //рекуррентное соотношение
60. i++;
61. } **while** (summand > EPS);
62. **return** resIter;
63. }

## 3.3. Полученные результаты

|  |
| --- |
| X:\\_URFU\TECHPROG\lab3\ex1\Release>ex1  Лабораторная работа 3: Вычисление функций с использованием их разложения в степенной ряд  ----------------------------------------------  Результат вычислений:  X= 1 SN= 2.71828 SE= 2.71825 Y= 2.71828  X=1.1 SN= 3.00417 SE= 3.00411 Y= 3.00417  X=1.2 SN= 3.32012 SE= 3.3201 Y= 3.32012  X=1.3 SN= 3.6693 SE= 3.66926 Y= 3.6693  X=1.4 SN= 4.0552 SE= 4.05513 Y= 4.0552  X=1.5 SN= 4.48169 SE= 4.48167 Y= 4.48169  X=1.6 SN= 4.95303 SE= 4.953 Y= 4.95303  X=1.7 SN= 5.47395 SE= 5.47388 Y= 5.47395  X=1.8 SN= 6.04965 SE= 6.04953 Y= 6.04965  X=1.9 SN= 6.68589 SE= 6.68586 Y= 6.68589  X= 2 SN= 7.38906 SE= 7.38899 Y= 7.38906  ---------------------------------------------- |

# Лабораторная работа №4. «Работа с одномерными массивами»

## Постановка задачи

1) Сформировать одномерный массив целых чисел, используя датчик случайных чисел.

2) Распечатать полученный массив.

3) Удалить элементы кратные 7.

4) Добавить после каждого нечетного элемента массива элемент со значением 0.

5) Распечатать полученный массив.

## 4.2. Программа решения задания

1. // Александр Молодых, ФТ-330801
2. // Лабораторная работа №4. Работа с одномерными массивами
3. // Вариант - 5
4. #define S 100
5. #include <iostream>
6. #include <iomanip> //для setw()
7. #include <locale> //для кириллицы
8. **using** **namespace** std;
9. /////////////////////////////////////////////////////////////////////////////
10. **void** **clearScreen**(); //очистка экрана с приглашением продолжения работы
11. /////////////////////////////////////////////////////////////////////////////
12. //Безопасный массив
13. **class** **Array**
14. {
15. **private:**
16. **int** \*arr; //сам массив
17. **int** size; //пользовательский размер
18. **public:**
19. //конструктор массива по размеру s
20. Array (**int** s) : size(s)
21. {
22. //допускается изменение размеров в пределах до S
23. arr = **new** **int**[S];
24. **for**(**int** i = **0**; i < size; i++)
25. {
26. arr[i] = rand() % **100** + **1**; //каждый элемент - от 1 до 100
27. }
28. }
29. //Удалить элементы кратные 7
30. **void** delDiv7()
31. {
32. **for** (**int** i = **0**; i < size; i++)
33. {
34. **if** (arr[i] % **7** == **0**)
35. {
36. **for** (**int** j = i; j < size; j++)
37. {
38. arr[j] = arr[j+**1**];
39. }
40. size--;
41. }
42. }
43. }
44. //Добавить после каждого нечетного элемента массива элемент со значением 0
45. **void** addZero()
46. {
47. **for** (**int** i = **0**; i < size; i++) //для элементов массива из пользовательской области
48. {
49. **if** (arr[i] % **2** != **0** && (size++) < S) //если элемент нечётный и не пересекается граница массива
50. {
51. **for** (**int** j = size; j >= i; j--) //для всех последующих элементов
52. {
53. arr[j + **1**] = arr[j]; //выполнить смещение вверх по индексу
54. }
55. arr[++i] = **0**; //нечётный элемент (который продублирован) заменить 0
56. }
57. **else** **if** (size > S)
58. {
59. clearScreen();
60. cout << "Переполнение массива!" << endl;
61. exit(**1**);
62. }
63. }
64. }
65. //Вывести массив на экран
66. **void** print()
67. {
68. **for** (**int** i = **0**; i < size; i++)
69. {
70. cout << "arr[" << i << "] = " << setw(**3**) << arr[i] << endl;
71. }
72. }
73. };
74. /////////////////////////////////////////////////////////////////////////////
75. **int** **main**()
76. {
77. setlocale(LC\_ALL,"Russian");
78. cout << "Лабораторная работа 4: Работа с одномерными массивами" << endl;
79. cout << "----------------------------------------------" << endl;
80. **int** userSize; //пользовательский размер массива
81. cout << "Введите размер массива (не больше " << S << "): ";
82. cin >> userSize;
83. //Генерация массива с пользовательским размером
84. Array arr(userSize);
85. cout << "----------------------------------------------" << endl;
86. cout << "Сформированный массив:" << endl;
87. //Вывод массива
88. arr.print();
89. //Удаление элементов кратных 7
90. arr.delDiv7();
91. clearScreen();
92. cout << "----------------------------------------------" << endl;
93. cout << "Удалены элементы кратные 7:" << endl;
94. //Вывод массива
95. arr.print();
96. //Добавление элементов со значением 0 после каждого нечётного
97. arr.addZero();
98. clearScreen();
99. cout << "----------------------------------------------" << endl;
100. cout << "Добавлены элементы со значением '0' после каждого нечётного:" << endl;
101. //Вывод массива
102. arr.print();
103. **return** **0**;
104. }
105. //Функция очистки экрана с приглашением для продолжения работы
106. **void** **clearScreen**()
107. {
108. cout << "Нажмите любую кнопку..." << endl;
109. system("pause>nul");
110. system("cls");
111. }

//Добавить после каждого нечетного элемента массива элемент со значением 0

1. **void** addZero()
2. {
3. **for** (**int** i = **0**; i < size; i++) //для элементов массива из пользовательской области
4. {
5. **if** (arr[i] % **2** != **0** && (size++) < S) //если элемент нечётный и не пересекается граница массива
6. {
7. **for** (**int** j = size; j >= i; j--) //для всех последующих элементов
8. {
9. arr[j + **1**] = arr[j]; //выполнить смещение вверх по индексу
10. }
11. arr[++i] = **0**; //нечётный элемент (который продублирован) заменить 0
12. }
13. **else** **if** (size > S)
14. {
15. clearScreen();
16. cout << "Переполнение массива!" << endl;
17. exit(**1**);
18. }
19. }
20. }
21. //Вывести массив на экран
22. **void** print()
23. {
24. **for** (**int** i = **0**; i < size; i++)
25. {
26. cout << "arr[" << i << "] = " << setw(**3**) << arr[i] << endl;
27. }
28. }
29. };
30. /////////////////////////////////////////////////////////////////////////////
31. **int** **main**()
32. {
33. setlocale(LC\_ALL,"Russian");
34. cout << "Лабораторная работа 4: Работа с одномерными массивами" << endl;
35. cout << "----------------------------------------------" << endl;

**int** userSize; //пользовательский размер массива

1. cout << "Введите размер массива (не больше " << S << "): ";
2. cin >> userSize;
3. //Генерация массива с пользовательским размером
4. Array arr(userSize);
5. cout << "----------------------------------------------" << endl;
6. cout << "Сформированный массив:" << endl;
7. //Вывод массива
8. arr.print();
9. //Удаление элементов кратных 7
10. arr.delDiv7();
11. clearScreen();
12. cout << "----------------------------------------------" << endl;
13. cout << "Удалены элементы кратные 7:" << endl;
14. //Вывод массива
15. arr.print();
16. //Добавление элементов со значением 0 после каждого нечётного
17. arr.addZero();
18. clearScreen();
19. cout << "----------------------------------------------" << endl;
20. cout << "Добавлены элементы со значением '0' после каждого нечётного:" << endl;
21. //Вывод массива
22. arr.print();
23. **return** **0**;
24. }
25. //Функция очистки экрана с приглашением для продолжения работы
26. **void** **clearScreen**()
27. {
28. cout << "Нажмите любую кнопку..." << endl;
29. system("pause>nul");
30. system("cls");
31. }

## 4.3. Программа решения задания

|  |
| --- |
| X:\\_URFU\TECHPROG\lab4\ex1\Release>ex1  Лабораторная работа 4: Работа с одномерными массивами  ----------------------------------------------  Введите размер массива (не больше 100): 15  ----------------------------------------------  Сформированный массив:  arr[0] = 42  arr[1] = 68  arr[2] = 35  arr[3] = 1  arr[4] = 70  arr[5] = 25  arr[6] = 79  arr[7] = 59  arr[8] = 63  arr[9] = 65  arr[10] = 6  arr[11] = 46  arr[12] = 82  arr[13] = 28  arr[14] = 62  Нажмите любую кнопку... |

|  |
| --- |
| ----------------------------------------------  Удалены элементы кратные 7:  arr[0] = 68  arr[1] = 1  arr[2] = 25  arr[3] = 79  arr[4] = 59  arr[5] = 65  arr[6] = 6  arr[7] = 46  arr[8] = 82  arr[9] = 62  Нажмите любую кнопку... |

|  |
| --- |
| ----------------------------------------------  Добавлены элементы со значением '0' после каждого нечётного:  arr[0] = 68  arr[1] = 1  arr[2] = 0  arr[3] = 25  arr[4] = 0  arr[5] = 79  arr[6] = 0  arr[7] = 59  arr[8] = 0  arr[9] = 65  arr[10] = 0  arr[11] = 6  arr[12] = 46  arr[13] = 82  arr[14] = 62 |

# Лабораторная работа №5. «Функции и массивы»

## Постановка задачи

Используя функции, решить указанную в варианте задачу. Массив должен передаваться в функцию как параметр.

Определить, является ли матрица ортонормированной, т. е. такой, что скалярное произведение каждой пары различных строк равно 0, а скалярное произведение строки самой на себя равно 1.

## Программа решения задания

1. // Александр Молодых, ФТ-330801
2. // Лабораторная работа №5. Функции и массивы
3. // Вариант - 5
4. #include <iostream>
5. #include <locale> //для кириллицы
6. **using** **namespace** std;
7. #define SX 3
8. #define SY 3
9. **void** **clearScreen**();
10. **void** **arrPrint**(**int**\* arr, **int** sx, **int** sy);
11. **int** **dotProduct**(**int**\* arr, **int** A, **int** B);
12. **bool** **isOrthonormal**(**int**\* arr, **int** sx, **int** sy);
13. //Определить является ли матрица ортонормированной, т. е. такой, что скалярное произведение каждой пары
14. //различных строк равно 0, а скалярное произведение строки самой на себя равно 1.
15. **int** **main**()
16. {
17. setlocale(LC\_ALL,"Russian");
18. cout << "Лабораторная работа 5: Функции и массивы" << endl;
19. cout << "----------------------------------------------" << endl;
20. cout << "Исходная матрица: " << endl;
21. **int** a[SX][SY] = {
22. {**1**,**0**,**0**},
23. {**0**,**1**,**0**},
24. {**0**,**0**,**1**}
25. };
26. **int**\* ptr = **reinterpret\_cast**<**int**\*>(a);
27. //Вывод
28. arrPrint(ptr, SX, SY);
30. //Проверка на ортонормированность
31. **if** (isOrthonormal(ptr, SX, SY))
32. cout << "Матрица ортонормирована" << endl;
33. **else**
34. cout << "Матрица не ортонормирована" << endl;
35. **return** **0**;
36. }
37. /\*
38. Проверка массива на ортонормированность.
39. Вычисляет dotProduct - скалярное произведение пар строк массива arr
40. Возвращает 1 в случае, когда матрица ортонормирована.
41. \*/
42. **bool** **isOrthonormal**(**int**\* arr, **int** sx, **int** sy)
43. {
44. //Результат, который может быть опровергнут циклом условий
45. **bool** res = **1**;
46. //Два цикла для 2 строк
47. **for** (**int** i = **0**; i < sx; i++)
48. **for** (**int** j = **0**; j < sy; j++)
49. {
50. //для произведений 2-х разных строк
51. **if** (i != j)
52. {
53. **if** (dotProduct(arr, i, j) != **0**)
54. res = **0**; //опровержение результата
55. }
56. //для произведений строки самой на себя
57. **else**
58. {
59. **if** (dotProduct(arr, i, j) != **1**)
60. res = **0**; //опровержение результата
61. }
62. }
63. **return** res;
64. }
65. /\*
66. Вычисление скалярного произведения для пар строк - A и B
67. Возвращает целое число
68. \*/
69. **int** **dotProduct**(**int**\* arr, **int** A, **int** B)
70. {
71. //Результат
72. **int** res = **0**;
73. **for** (**int** i = **0**; i < SX; i++) //для всех элементов строки
74. res += \*(arr + A\*SX + i) \* \*(arr + B\*SX + i); //складывать их произведения
76. //Результат - сумма произведений всех элементов строки
77. **return** res;
78. }
79. /\*
80. Вывод массива arr на печать
81. \*/
82. **void** **arrPrint**(**int**\* arr, **int** sx, **int** sy)
83. {
84. **for** (**int** i = **0**; i < SX; i++) {
85. **for** (**int** j = **0**; j < SY; j++) {
86. cout << \*(arr + i\*SY + j) << ' ';
87. }
88. cout << endl;
89. }
90. }
91. //Функция очистки экрана с приглашением для продолжения работы
92. **void** **clearScreen**()
93. {
94. cout << "Нажмите любую кнопку..." << endl;
95. system("pause>nul");
96. system("cls");
97. }

## Результат решения

|  |
| --- |
| X:\\_URFU\TECHPROG\lab5\ex1\Release>ex1  Лабораторная работа 5: Функции и массивы  ----------------------------------------------  Исходная матрица:  1 0 0  0 1 0  0 0 1  Матрица ортонормирована |

# Лабораторная работа № 6. «Строки»

## Постановка задачи

Задана строка, состоящая из символов. Символы объединяются в слова. Слова друг от друга отделяются одним или несколькими пробелами. В конце текста ставится точка. Текст содержит не более 255 символов. Выполнить ввод строки, используя функцию Gets(s) и обработку строки в соответствии со своим вариантом.

Напечатать все слова, которые совпадают с ее первым словом.

## Программа решения задания

1. // Александр Молодых, ФТ-330801
2. // Лабораторная работа №6. Строки
3. // Вариант - 5
4. #include <iostream>
5. #include <stdio.h> //для gets()
6. #include <iomanip> //для setw()
7. #include <locale> //для кириллицы
8. #include <string> //для работы со строками
9. **using** **namespace** std;
10. #define BUF\_SIZE 256
11. **int** **wordCount**(**char**\* sentence);
12. **char**\*\* **wordList**(**char**\* sentence, **int** itemsNum);
13. **int** **wordEntry**(**char**\*\* list, **char**\* word, **int** itemsNum);
14. **int** **main**()
15. {
16. setlocale(LC\_ALL,"Russian");
17. cout << "Лабораторная работа 6: Строки" << endl;
18. cout << "----------------------------------------------" << endl;
19. //Исходное предложение:
20. **char**\* paragraph = " трижды два - шесть, трижды три - девять, трижды Герой Советского Союза .";
21. cout << "Вы ввели: " << paragraph << endl;
22. cout << "----------------------------------------------" << endl;
23. //Подсчёт количества слов:
24. **int** words = wordCount(paragraph);
25. //Форирование списка слов:
26. **char**\*\* list = wordList(paragraph, words);
27. //Число совпадений слов списка с первым словом из него:
28. **int** entries = wordEntry(list, list[**0**], words);
29. cout << "Число слов, совпадающих с первым словом '" << list[**0**] << "' строки: " << entries << endl;
30. **return** **0**;
31. }
32. /\*
33. Подсчитывает число слов в предложении
34. \*/
35. **int** **wordCount**(**char**\* sentence)
36. {
37. **int** count = **0**; //счётчик слов
39. //Цикл по всему предложению:
40. **int** i = **0**; //счётчик цикла
41. **do**
42. {
43. //Условие проверки конца слова:
44. **if** ((sentence[i+**1**] == ' ' && (sentence[i] != '.') && (sentence[i] != ' ')) || (sentence[i+**1**] == '.' && (sentence[i] != '.') && (sentence[i] != ' ')))
45. {
46. count++;
47. }
48. i++;
49. } **while** (sentence[i] != '.' && sentence[i+**1**] != '\0' );
50. **return** count;
51. }
52. /\*
53. Формирует список слов из одного преложения
54. \*/
55. **char**\*\* **wordList**(**char**\* sentence, **int** itemsNum)
56. {
57. **char**\*\* list; //список результатов
58. list = **new** **char**\* [itemsNum]; //выделяем память под сам список
60. //Цикл по всему предложению:
61. **int** i = **0**; //счётчик цикла
62. **int** count = **0**; //счётчик слов
63. **char**\* buf = **new** **char** [BUF\_SIZE]; //выделяем буфер для слова
64. **do**
65. {
66. //Если слово началось:
67. **if** (sentence[i] != ' ' && sentence[i] != '.')
68. {
69. **int** j = i; //фиксируем позицию начала слова
70. //Цикл по одному слову:
71. **do**
72. {
73. buf[j-i] = sentence[j]; //заполняем буфер посимвольно
74. j++; //смещаем позицию
75. } **while** (sentence[j] != ' ' && sentence[j] != '.');
76. buf[j - i] = **0**; //j-i - последняя позиция в слове
77. list[count] = **new** **char** [j - i + **1**]; //выделяем память для слова в списке:
78. //j-i+1 - длина слова с учётом терминального нуля
79. strcpy(list[count++], buf); //копируем из буфера в список
80. i = j; //"проматываем" счётчик на следующий последний символ слова
81. }
82. i++;
83. } **while** (sentence[i] != '.' && sentence[i+**1**] != '\0' );
84. //возврат указателя на список
85. **return** list;
86. }
87. /\*
88. Подсчитывает количество совпадений слова в списке
89. \*/
90. **int** **wordEntry**(**char**\*\* list, **char**\* word, **int** itemsNum)
91. {
92. **int** entryCount = **0**; //счётчик количества совпадений
93. //Цикл по всему списку:
94. **for** (**int** i = **0**; i < itemsNum; i++)
95. {
96. //Если слова равны
97. **if** (!strcmp(list[i], word))
98. {
99. entryCount++; //увеличить счётчик
100. }
101. }
102. **return** entryCount;
103. }

## Результаты выполнения программы

|  |
| --- |
| X:\\_URFU\TECHPROG\lab6\ex1\Release>ex1  Лабораторная работа 6: Строки  ----------------------------------------------  Вы ввели: трижды два - шесть, трижды три - девять, трижды Герой Советского Союза .  ----------------------------------------------  Число слов, совпадающих с первым словом 'трижды' строки: 3 |

# Лабораторная работа № 7.1. «Перегрузка функций в Си++»

## Постановка задачи

Написать перегруженные функции и основную программу, которая их вызывает:

а) для умножения вещественных чисел;

б) для умножения комплексных чисел.

## 7.2. Программа решения задания

1. // Александр Молодых, ФТ-330801
2. // Лабораторная работа №7.1. Перегрузка функций
3. // Вариант - 5
4. /\*
5. Написать перегруженные функции и основную программу, которая их вызывает.
6. а) для умножения вещественных чисел;
7. б) для умножения комплексных чисел.
8. \*/
9. #include <iostream>
10. #include <stdio.h> //для gets()
11. #include <locale> //для кириллицы
12. #include <string> //для работы со строками
13. **using** **namespace** std;
14. **struct** Complex
15. {
16. **public:**
17. **double** re;
18. **double** im;
19. Complex() : re(**0**), im(**0**)
20. {};
21. Complex(**double** R, **double** I) : re(R), im(I)
22. {};
23. };
24. **void** **print**(**char**\* title, Complex digit);
25. **void** **print**(**char**\* title, **double** digit);
26. Complex **multiply**(Complex x, Complex y);
27. **double** **multiply**(**double** x, **double** y);
28. **int** **main**()
29. {
30. setlocale(LC\_ALL,"Russian");
31. cout << "Лабораторная работа 7: Перегрузка функций" << endl;
32. cout << "----------------------------------------------" << endl;
34. cout << "Вещественные числа:" << endl;
35. //Вещественные:
36. **double** d1 = **12**;
37. **double** d2 = **13**;
38. **double** d3 = multiply(d1, d2);
39. print("d1", d1);
40. print("d2", d2);
41. print("d1 + d2", d3);
42. cout << "Комплексные числа:" << endl;
43. //Комплексные:
44. Complex c1(**12**, **13**);
45. Complex c2(-**5**, -**6**);
46. Complex c3 = multiply(c1, c2);
47. print("c1", c1);
48. print("c2", c2);
49. print("c1 + c2", c3);
50. **return** **0**;
51. }
52. Complex **multiply**(Complex x, Complex y)
53. {
54. Complex c;
55. c.re = x.re \* y.re;
56. c.im = x.im \* y.im;
57. **return** c;
58. }
59. **double** **multiply**(**double** x, **double** y)
60. {
61. **return** x \* y;
62. }
63. **void** **print**(**char**\* title, Complex digit)
64. {
65. cout << title << " = " << digit.re << " + " << digit.im << "i" << endl;
66. }
67. **void** **print**(**char**\* title, **double** digit)
68. {
69. cout << title << " = " << digit << endl;
70. }

## 7.3. Результат выполнения программы

|  |
| --- |
| X:\\_URFU\TECHPROG\lab7\ex1\Release>ex1  Лабораторная работа 7: Перегрузка функций  ----------------------------------------------  Вещественные числа:  d1 = 12  d2 = 13  d1 + d2 = 156  Комплексные числа:  c1 = 12 + 13i  c2 = -5 + -6i  c1 + c2 = -60 + -78i |

# Лабораторная работа № 7.2. «Функции с переменным числом параметров»

## 8.1.Постановка задачи

Решить указанную в варианте задачу, используя функции с переменным числом параметров.

Написать функцию sum с переменным числом параметров, которая находит сумму чисел типа int по формуле:

S=a1\*a2-a2\*a3+a3\*a4-. . . . .

Написать вызывающую функцию main, которая обращается к функции sum не менее трех раз с количеством параметров 5, 10, 12.

## 8.2. Программа решения задания

1. // Александр Молодых, ФТ-330801
2. // Лабораторная работа №7.2. Функции с переменным числом параметров
3. // Вариант - 5
4. #include <iostream>
5. #include <stdio.h> //для gets()
6. #include <iomanip> //для setw()
7. #include <locale> //для кириллицы
8. #include <string> //для работы со строками
9. **using** **namespace** std;
10. **int** **sum**(**int** k, ...);
11. **void** **sumprint**(**int** k, ...);
12. **int** **main**()
13. {
14. setlocale(LC\_ALL,"Russian");
15. **int** g;
16. cout << "Лабораторная работа 7.2: Функции с переменным числом параметров" << endl;
17. cout << "----------------------------------------------" << endl;
18. cout << "Вызов с числом параметров 5...**\n**";
19. g = sum(**5**,**1**,**1**,**1**,**1**,**1**);
20. cout << "Ответ: ";
21. sumprint(**5**,**1**,**1**,**1**,**1**,**1**);
22. cout << "=" << g << endl;
23. cout << "----------------------------------------------" << endl;
24. cout << "Вызов с числом параметров 10...**\n**";
25. g = sum(**10**,**13**,**12**,**20**,**3**,**4**,**6**,**88**,**102**,**3**,**33**);
26. cout << "Ответ: ";
27. sumprint(**10**,**13**,**12**,**20**,**3**,**4**,**6**,**88**,**102**,**3**,**33**);
28. cout << "=" << g << endl;
29. cout << "----------------------------------------------" << endl;
30. cout << "Вызов с числом параметров 15...**\n**";
31. g = sum(**15**,**1**,**2**,**3**,**4**,**5**,**6**,**7**,**8**,**9**,**10**,**11**,**12**,**13**,**14**,**15**);
32. cout << "Ответ: ";
33. sumprint(**15**,**1**,**2**,**3**,**4**,**5**,**6**,**7**,**8**,**9**,**10**,**11**,**12**,**13**,**14**,**15**);
34. cout << "=" << g << endl;
35. **return** **0**;
36. }
37. **void** **sumprint**(**int** k, ...)
38. {
39. **int** \*p = &k; //настроили указатель на параметр k
40. **short** sign = **1**; //знак: 1 - плюс, -1 - минус
41. **while**(k!=**1**)
42. {
43. **if** (sign == **1**)
44. cout << "+";
45. **if** (sign == -**1**)
46. cout << "-";
47. cout << (\*(++p)) << "\*" << (\*(p+**1**));
48. sign \*= -**1**;
49. k--;
50. }
51. }
52. **int** **sum** (**int** k, ...)
53. {
54. **int** \*p = &k; //настроили указатель на параметр k
55. **int** sum = **0**;
56. **short** sign = **1**; //знак: 1 - плюс, -1 - минус
57. **while**(k!=**1**)
58. {
59. sum += sign \* ((\*(++p)) \* (\*(p+**1**)));
60. sign \*= -**1**;
61. k--;
62. }
63. **return** sum;
64. }

## 8.3. Результат выполнения программы

|  |
| --- |
| X:\\_URFU\TECHPROG\lab7\ex2\Release>ex2  Лабораторная работа 7.2: Функции с переменным числом параметров  ----------------------------------------------  Вызов с числом параметров 5...  Ответ: +1\*1-1\*1+1\*1-1\*1=0  ----------------------------------------------  Вызов с числом параметров 10...  Ответ: +13\*12-12\*20+20\*3-3\*4+4\*6-6\*88+88\*102-102\*3+3\*33=8229  ----------------------------------------------  Вызов с числом параметров 15...  Ответ: +1\*2-2\*3+3\*4-4\*5+5\*6-6\*7+7\*8-8\*9+9\*10-10\*11+11\*12-12\*13+13\*14-14\*15=-112 |

# Лабораторная работа №8. «Блоковый ввод-вывод»

## Постановка задачи

Сформировать двоичный файл из элементов, заданной в варианте структуры, распечатать его содержимое, выполнить удаление и добавление элементов в соответствии со своим вариантом, используя для поиска удаляемых или добавляемых элементов функцию. Формирование, печать, добавление и удаление элементов оформить в виде функций. Предусмотреть сообщения об ошибках при открытии файла и выполнении операций ввода/вывода.

Структура "Человек":

* фамилия, имя, отчество;
* год рождения;
* рост;
* вес.

Удалить все элементы с указанным ростом и весом, добавить элемент после элемента с указанной фамилией.

## Программа решения задачи

1. // Александр Молодых, ФТ-330801
2. // Лабораторная работа №8. Блоковый ввод-вывод
3. // Вариант - 5
4. #include <iostream>
5. #include <stdio.h> //для gets()
6. #include <iomanip> //для setw()
7. #include <locale> //для кириллицы
8. #include <string> //для работы со строками
9. **using** **namespace** std;
10. //Структура "Человек"
11. **struct** human
12. {
13. **char** first\_name[**40**]; //имя
14. **char** last\_name[**40**]; //фамилия
15. **char** middle\_name[**40**]; //отчество
16. **unsigned** **int** year\_of\_birth; //год рождения
17. **unsigned** **short** height; //рост
18. **unsigned** **short** weight; //вес
20. **public:**
21. human()
22. { }
23. };
24. //ФУНКЦИИ ФОРМИРОВАНИЯ И ВЫВОДА
25. //Формирование массива экземпляров структуры
26. human\* **setHuman**(**int**);
27. //Формирование одного экземпляра структуры
28. human **setOneHuman**(**int** a);
29. //Вывод массива экземпляров стркутуры на печать
30. **int** **getHuman**(human\*, **int**);
31. //ФУНКЦИИ ОБРАБОТКИ
32. //Удаление экземпляра структуры по указанным свойствам "Рост" и "Вес"
33. human\* **delHW**(human\*, **int**&, **int**, **int**);
34. //Добавление одного экземпляра после указанного свойства "Фамилия"
35. human\* **addAfter**(human\*, **int**&, **char**[]);
36. //ФУНКЦИИ РАБОТЫ С ДВОИЧНЫМИ ДАННЫМИ
37. //Запись в файл массива экземпляров структуры в форме двоичных данных
38. **int** **writeHuman**(human, **FILE**\*);
39. //Чтение из двоичного файла данных в массив эеземпляров структуры
40. human **readHuman**(**FILE**\*);
41. **int** **main**()
42. {
43. setlocale(LC\_ALL,"Russian");
44. cout << "Лабораторная работа 8: Блоковый ввод-вывод" << endl;
45. cout << "----------------------------------------------" << endl;
46. **FILE**\* f; //указатель, связанный с файлом
48. //Открытие файла
49. f = fopen("f.dat", "w+b");
50. **if** (f == NULL) //если ошибка
51. exit(**1**);
52. **int** amount;
53. cout << "Введите количество человек на добавление: ";
54. cin >> amount;
55. human\* arr = **new** human [amount];
57. //Запрашиваем данные
58. arr = setHuman(amount);
59. //Запись в файл
60. **for** (**int** i = **0**; i < amount; i++)
61. {
62. **if** (!writeHuman(arr[i], f))
63. exit(**1**);
64. }
65. //Сброс указателя
66. rewind(f);
67. **int** i = **0**;
69. //Чтение из файла в массив
70. **while**(!feof(f) && i < amount)
71. {
72. arr[i] = readHuman(f);
73. i++;
74. }
75. //Вывод данных из массива
76. getHuman(arr, amount);
78. **unsigned** **int** height; //вес
79. **unsigned** **int** weight; //рост
80. cout << "----------------------------------------------" << endl;
81. //Запрос роста и веса
82. cout << "Укажите рост и вес человека, которого нужно удалить**\n**: " << endl;
83. cout << "Рост: "; cin >> height;
84. cout << "**\n**Вес: "; cin >> weight;
85. arr = delHW(arr, amount, height, weight);
87. cout << "----------------------------------------------" << endl;
88. //Вывод данных из массива
89. getHuman(arr, amount);
90. **char** last\_name[**40**];
91. //Запрос фамилии
92. cout << "Укажите фамилию человека, после которого нужно добавить запись**\n**: " << endl;
93. cin >> last\_name;
94. arr = addAfter(arr, amount, last\_name);
96. //Вывод данных из массива
97. getHuman(arr, amount);
98. //Закрываем файл
99. fclose(f);
100. **return** **0**;
101. }
102. human **setOneHuman**(**int** i)
103. {
104. human obj;
105. cout << "----------------------------------------------**\n**";
106. cout << "Человек (ID = " << i << "): " << endl;
107. cout << "----------------------------------------------**\n**";
108. cout << "**\n**Имя: "; cin >> obj.first\_name;
109. cout << "Фамилия: "; cin >> obj.last\_name;
110. cout << "Отчество: "; cin >> obj.middle\_name;
111. cout << "Год рождения: "; cin >> obj.year\_of\_birth;
112. cout << "Рост: "; cin >> obj.height;
113. cout << "Вес: "; cin >> obj.weight;
114. cout << endl;
115. **return** obj;
116. }
117. human\* **setHuman**(**int** a)
118. {
119. human\* obj;
120. **for** (**int** i = **0**; i < a; i++)
121. {
122. cout << "----------------------------------------------**\n**";
123. cout << "Человек (ID = " << i << "): " << endl;
124. cout << "----------------------------------------------**\n**";
125. cout << "**\n**Имя: "; cin >> obj[i].first\_name;
126. cout << "Фамилия: "; cin >> obj[i].last\_name;
127. cout << "Отчество: "; cin >> obj[i].middle\_name;
128. cout << "Год рождения: "; cin >> obj[i].year\_of\_birth;
129. cout << "Рост: "; cin >> obj[i].height;
130. cout << "Вес: "; cin >> obj[i].weight;
131. cout << endl;
132. }
133. **return** obj;
134. }
135. **int** **getHuman**(human\* obj, **int** a)
136. {
137. **if** (a == **0**)
138. {
139. cout << "----------------------------------------------**\n**";
140. cout << "Объектов нет" << endl;
141. cout << "----------------------------------------------**\n**";
142. **return** **0**;
143. }
144. **for** (**int** i = **0**; i < a; i++)
145. {
146. cout << "----------------------------------------------**\n**";
147. cout << "Человек (ID = " << i << "): " << endl;
148. cout << "----------------------------------------------**\n**";
149. cout << "**\n**Имя: "; cout << obj[i].first\_name;
150. cout << "**\n**Фамилия: "; cout << obj[i].last\_name;
151. cout << "**\n**Отчество: "; cout << obj[i].middle\_name;
152. cout << "**\n**Год рождения: "; cout << obj[i].year\_of\_birth;
153. cout << "**\n**Рост: "; cout << obj[i].height;
154. cout << "**\n**Вес: "; cout << obj[i].weight;
155. cout << endl;
156. }
157. **return** **0**;
158. }
159. **int** **writeHuman**(human h, **FILE**\* f)
160. {
161. fwrite(&h, **sizeof**(human), **1**, f);
162. **if** (ferror(f) == NULL)
163. **return** **1**;
164. **return** **0**;
165. }
166. human **readHuman**(**FILE**\* f)
167. {
168. human h;
170. **if** (f == NULL)
171. {
172. cout << "ERR: возникла ошибка при чтении файла**\n**";
173. **return** human();
174. }
175. fread(&h, **sizeof**(human),**1**,f);
177. **return** h;
178. }
179. human\* **delHW**(human\* arr, **int**& a, **int** h, **int** w)
180. {
181. **for** (**int** i = **0**; i < a; i++)
182. {
183. **if** (arr[i].weight == w && arr[i].height == h)
184. {
185. **for** (**int** j = i; j < a - **1**; j++)
186. {
187. arr[j] = arr[j+**1**];
188. }
189. //delete [] arr[a];
190. a--;
191. }
192. }
194. **return** arr;
195. }
196. human\* **addAfter**(human\* arr, **int**& a, **char** ln[])
197. {
198. human\* res = **new** human [a+**1**];
199. **int** j = **0**;
201. **if** (a == **0**)
202. {
203. res[**0**] = setOneHuman(**0**);
204. a++;
205. **return** res;
206. }
207. **for** (**int** i = **0**; i < a; i++)
208. {
209. **if** (strcmp(arr[i].last\_name, ln) == **0**)
210. {
211. res[i] = setOneHuman(i);
212. a++;
213. **for** (**int** j = i + **1**; j < a; j++)
214. {
215. res[j] = arr[j - **1**];
216. }
217. **return** res;
218. }
219. **else**
220. {
221. res[i] = arr[i];
222. }
223. }
225. **delete** &res[a+**1**];
226. **return** res;
227. }

## Результат выполнения программы

|  |
| --- |
| X:\\_URFU\TECHPROG\lab8\ex1\Release>ex1  Лабораторная работа 8: Блоковый ввод-вывод  ----------------------------------------------  Введите количество человек на добавление: 5  ----------------------------------------------  Человек (ID = 0):  ----------------------------------------------  Имя: Svyatoslav  Фамилия: Taisky  Отчество: Svyatoslavovich  Год рождения: 1994  Рост: 70  Вес: 70  ----------------------------------------------  Человек (ID = 1):  ----------------------------------------------  Имя: Lyubov  Фамилия: Baszhenova  Отчество: Tarasovna  Год рождения: 1968  Рост: 160  Вес: 68  ----------------------------------------------  Человек (ID = 2):  ----------------------------------------------  Имя: Vladlen  Фамилия: Gulevich  Отчество: Grigorievich  Год рождения: 1971  Рост: 170  Вес: 75  ----------------------------------------------  Человек (ID = 3):  ----------------------------------------------  Имя: Dobromysl  Фамилия: Ichyotkin  Отчество: Nikolaevich  Год рождения: 1995  Рост: 180  Вес: 85  ----------------------------------------------  Человек (ID = 4):  ----------------------------------------------  Имя: Katerina  Фамилия: Volkova  Отчество: Gennadievna  Год рождения: 1968  Рост: 177  Вес: 77 |

|  |
| --- |
| Укажите рост и вес человека, которого нужно удалить  :  Рост: 180  Вес: 85  ----------------------------------------------  ----------------------------------------------  Человек (ID = 0):  ----------------------------------------------  Имя: Svyatoslav  Фамилия: Taisky  Отчество: Svyatoslavovich  Год рождения: 1994  Рост: 70  Вес: 70  ----------------------------------------------  Человек (ID = 1):  ----------------------------------------------  Имя: Lyubov  Фамилия: Baszhenova  Отчество: Tarasovna  Год рождения: 1968  Рост: 160  Вес: 68  ----------------------------------------------  Человек (ID = 2):  ----------------------------------------------  Имя: Vladlen  Фамилия: Gulevich  Отчество: Grigorievich  Год рождения: 1971  Рост: 170  Вес: 75  ----------------------------------------------  Человек (ID = 3):  ----------------------------------------------  Имя: Katerina  Фамилия: Volkova  Отчество: Gennadievna  Год рождения: 1968  Рост: 177  Вес: 77 |

|  |
| --- |
| Укажите фамилию человека, после которого нужно добавить запись  :  Baszhenova  ----------------------------------------------  Человек (ID = 2):  ----------------------------------------------  Имя: Vasilisa  Фамилия: Kulagina  Отчество: Maksimovna  Год рождения: 1983  Рост: 170  Вес: 58 |

|  |
| --- |
| ----------------------------------------------  Человек (ID = 0):  ----------------------------------------------  Имя: Svyatoslav  Фамилия: Taisky  Отчество: Svyatoslavovich  Год рождения: 1994  Рост: 70  Вес: 70  ----------------------------------------------  Человек (ID = 1):  ----------------------------------------------  Имя: Lyubov  Фамилия: Baszhenova  Отчество: Tarasovna  Год рождения: 1968  Рост: 160  Вес: 68  ----------------------------------------------  Человек (ID = 2):  ----------------------------------------------  Имя: Vasilisa  Фамилия: Kulagina  Отчество: Maksimovna  Год рождения: 1983  Рост: 170  Вес: 58  ----------------------------------------------  Человек (ID = 3):  ----------------------------------------------  Имя: Vladlen  Фамилия: Gulevich  Отчество: Grigorievich  Год рождения: 1971  Рост: 170  Вес: 75  ----------------------------------------------  Человек (ID = 4):  ----------------------------------------------  Имя: Katerina  Фамилия: Volkova  Отчество: Gennadievna  Год рождения: 1968  Рост: 177  Вес: 77 |

1. // Александр Молодых, ФТ-330801
2. // Лабораторная работа №9. Строковый ввод-вывод
3. // Вариант - 5
4. #include <iostream>
5. #include <vector> //для работы с векторами
6. #include <conio.h> //для getche()
7. #include <locale> //для кириллицы
8. #include <fstream> //для работы с файлами
9. #include <string>
10. **using** **namespace** std;
11. **int** **form** (string);
12. **int** **display** (string);
13. **int** **copy\_K** (string, string, **int**);
14. **int** **score** (string);
15. **int** **main**()
16. {
17. setlocale(LC\_ALL,"Russian");
18. cout << "Лабораторная работа 9: Строковый ввод-вывод" << endl;
19. cout << "----------------------------------------------" << endl;
20. form("f1.txt");
21. cout << "----------------------------------------------" << endl;
23. display("f1.txt");
24. cout << "----------------------------------------------" << endl;
26. **int** K;
27. cout << "Введите число K: ";
28. cin >> K;
30. **if** (copy\_K("f1.txt", "f2.txt", K))
31. {cout << "Копирование не удалось: слишком мало элементов или завышено число K.**\n**";}
33. cout << "----------------------------------------------" << endl;
34. cout << "Число гласных букв в файле F2: " << score("f2.txt") << endl;
35. **return** **0**;
36. }
37. //////////////////////////////////////////////////////////////////////////
38. **int** **form** (string fn)
39. {
40. cout << "----------------------------------------------**\n**";
41. cout << "Введите элементы массива для файла F1: **\n**";
42. cout << "----------------------------------------------**\n**";
44. vector<string> lines;
45. **char** confirm = '\0';
47. **int** i = **0**;
48. **for** (**int** i = **0**; confirm != 'n'; i++)
49. {
50. string data;
51. cout << i << ": ";
52. cin >> data;
53. lines.push\_back(data);
54. cout << "Продолжить(y/n)?";
55. cin >> confirm;
56. }
57. ofstream fout(fn);
58. **for** (**int** i = **0**; i < lines.size(); i++)
59. fout << lines[i] << '\n';
61. fout.close();
62. **return** **0**;
63. }
64. **int** **display** (string fn)
65. {
66. cout << "----------------------------------------------**\n**";
67. cout << "Элементы массива из файла F1: **\n**";
68. cout << "----------------------------------------------**\n**";
69. ifstream fin(fn);
70. vector<string> lines;
71. string data;
73. **while**(!fin.eof())
74. {
75. getline(fin, data);
76. **if** (data != "**\0**")
77. lines.push\_back(data);
78. }
79. **for** (**int** i = **0**; i < lines.size(); i++)
80. cout << i << ": " << lines[i] << '\n';
81. fin.close();
82. **return** **0**;
83. }
84. **int** **copy\_K** (string fn1, string fn2, **int** k)
85. {
86. ifstream fin (fn1);
87. ofstream fout (fn2);
88. vector<string> lines;
89. string data;
91. **while**(!fin.eof())
92. {
93. getline(fin, data);
94. **if** (data != "**\0**")
95. lines.push\_back(data);
96. }
97. fin.close();
98. **if** (k + **5** >= lines.size() && k >= **0**)
99. **return** **1**;
100. **for** (**int** i = k; i < k + **5**; i++)
101. fout << lines[i] << '\n';
103. fout.close();
104. **return** **0**;
105. }
106. **int** **score**(string fn)
107. {
108. **char**\* vowels = "aeiouyAEIOUYаоуыэяёюиеАОУЫЭЯЁЮИЕ";
109. ifstream fin (fn);
110. vector<string> lines;
111. **int** counter = **0**;
112. **while**(!fin.eof())
113. {
114. string data;
115. getline(fin, data);
116. **if** (data != "**\0**")
117. lines.push\_back(data);
118. }
119. **for** (**int** k = **0**; k < lines.size(); k++)
120. {
121. string data = lines[k];
122. **for** (**int** i = **0**; i < strlen(data.c\_str()); i++)
123. **for** (**int** j = **0**; j < strlen(vowels); j++)
124. **if** (data.c\_str()[i] == vowels[j])
125. counter++;
126. }
127. **return** counter;
128. }

http://hilite.me/