**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР**

отчет

**по лабораторной работе №3**

**по дисциплине «Программирование»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 1302 |  | Новиков Г.В. |
| Преподаватель |  | Калмычков В.А. |

Санкт-Петербург

2022

Оглавление

[**Исходная формулировка** 3](#_Toc114203549)

[**Контрольный пример** 3](#_Toc114203550)

[**Формальная постановка задачи** 3](#_Toc114203551)

[**Формат хранения данных** 3](#_Toc114203552)

[**Ограничения, обусловленные компьютером** 3](#_Toc114203553)

[**Макет ввода/вывода** 3](#_Toc114203554)

[**Алгоритм решения** 4](#_Toc114203555)

[**Реализация списка** 5](#_Toc114203556)

[**Текс программы** 5](#_Toc114203557)

[**Результат работы программы** 6](#_Toc114203558)

[**Вывод** 7](#_Toc114203559)

## **Исходная формулировка**

Вставить заданное число элементов перед последним элементом.

# **Контрольный пример**

Изначально:

toot tto

tot to

ot

Вставить:

tototo

Результат:

toot tto

tot to

tototo

ot

# **Формальная постановка задачи**

Дано: список

Вставить: элементы перед последним элементом списка

# **Формат хранения данных**

|  |  |  |
| --- | --- | --- |
| Идентификатор типа | Тип | Назначение |
| File | fstream | Файловые переменные |
| Next, prev, curr, el, last, prelast | Elem | Пользовательский тип |
| stroka | char | Строки |
| Letters, t | char | Символы |
| Num, i | int | Счётчик |
| len | int | Длинна строки |
| in | Elem\* | Функция ввода |
| Out, printel | void | Функция вывода |
| insert\_before\_last | Elem\* | Функция обработки |
| main | int | Основная функция |
| Del, desintegrate | void | Функция удаления |

# **Ограничения, обусловленные компьютером**

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

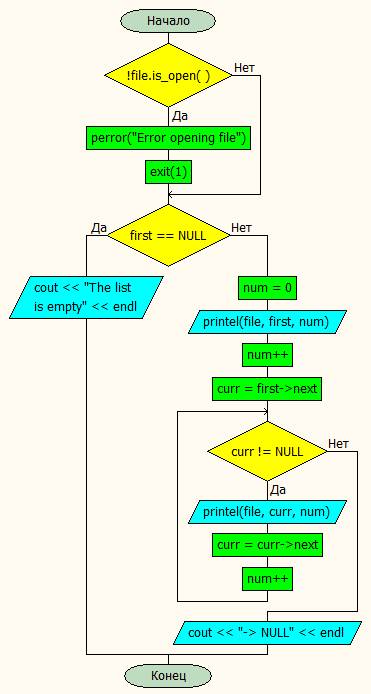
Символьный тип char. Минимальное значение -128, максимальное значение 128.

# **Макет ввода/вывода**

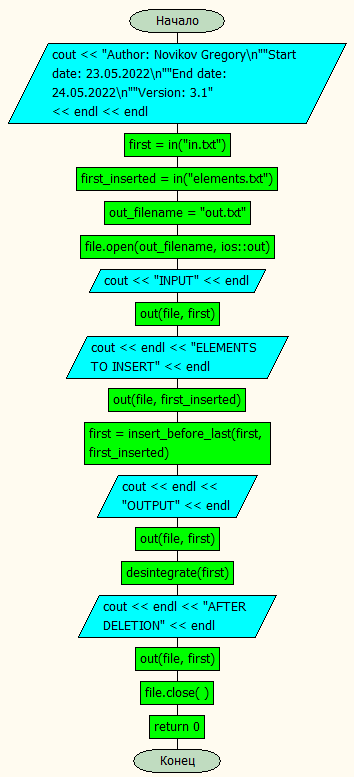
|  |  |
| --- | --- |
| Ввод из файла | file >> |
| Вывод из файла | file << |
| Ввод с клавиатуры | cin >> |
| Вывод на экран | cout << |

# **Алгоритм решения**

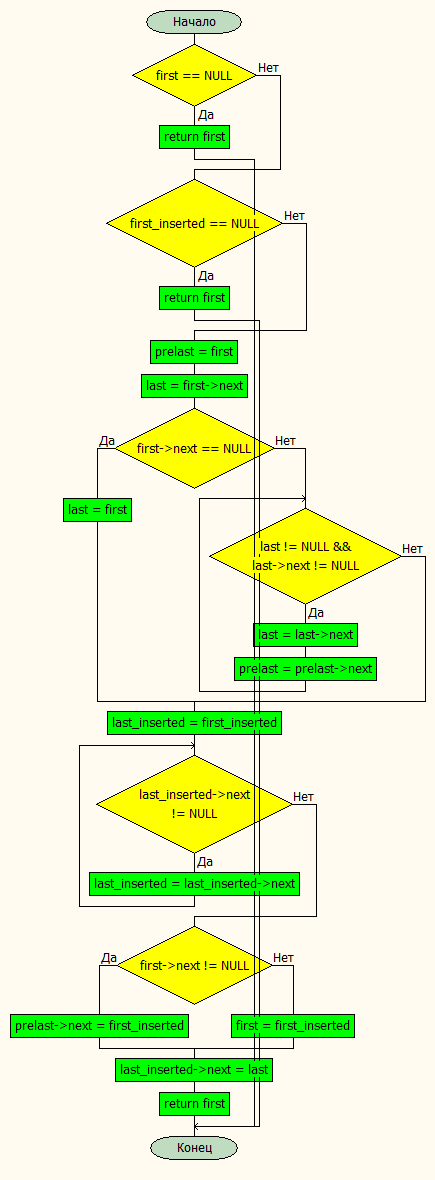
Out()



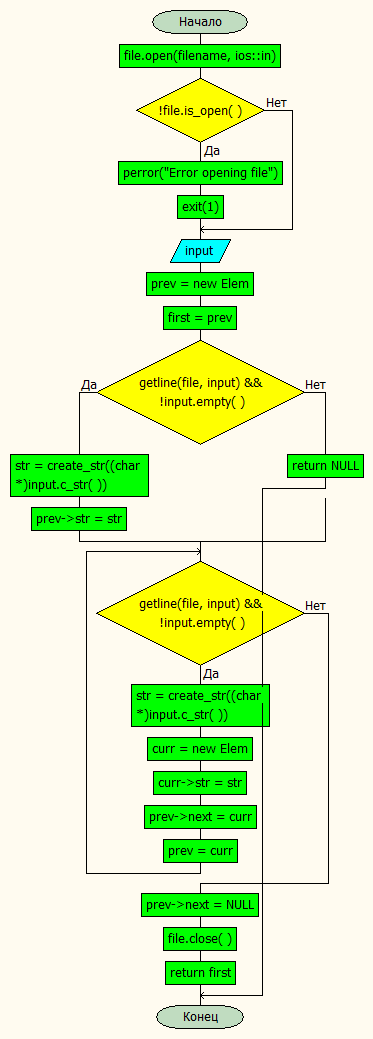
main()



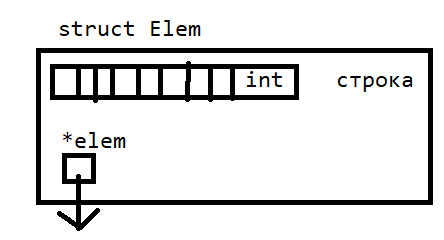
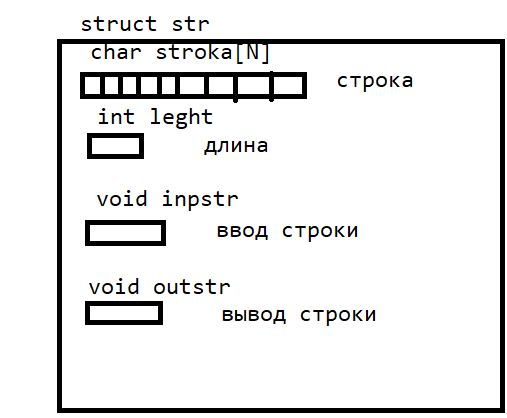
insert\_before\_last ()



Inp()



# **Реализация списка**



# **Текс программы**

|  |  |  |  |
| --- | --- | --- | --- |
| Str.h() | Elem.h | Functions.h | Del.cpp |
| #pragma once  #include <iostream>  #include <fstream>  #include <string>  using namespace std;  struct Str {  char letters[100];  int len = 0;  void del();  }; | #pragma once  #include "Str.h"  struct Elem {  Str str;  Elem\* next;  void del();  }; | #pragma once  #include "Elem.h"  Elem\* in(string filename);  Str create\_str(char\* word);  void out(fstream& file, Elem\* first);  void printel(fstream& file, Elem\* el, int num);  Elem\* insert\_before\_last(Elem\* first, Elem\* first\_inserted);  void desintegrate(Elem\*& first); | #include "Elem.h"  #include "Str.h"  #include "functions.h"  void Str::del() {  delete[] letters;  }  void Elem::del() {  if (next != NULL) {  next->del();  }  delete next;  next = NULL;  }  void desintegrate(Elem\*& first) {  if (first != NULL) {  first->del();  delete first;  first = NULL;  }  } |

|  |  |  |  |
| --- | --- | --- | --- |
| In.cpp | Out.cpp | Insert.cpp | Main.cpp |
| #include "functions.h"  Elem\* in(string filename) {  fstream file;  file.open(filename, ios::in);  if (!file.is\_open()) {  perror("Error opening file");  exit(1);  }  string input;  Elem\* prev = new Elem;  Elem\* first = prev;  if (getline(file, input) && !input.empty()) {  Str str = create\_str((char\*)input.c\_str());  prev->str = str;  }  else {  return NULL;  }  while (getline(file, input) && !input.empty()) {  Str str = create\_str((char\*)input.c\_str());  Elem\* curr = new Elem;  curr->str = str;  prev->next = curr;  prev = curr;  }  prev->next = NULL;  file.close();  return first;  }  Str create\_str(char\* word) {  Str\* str = new Str;  for (char\* t = word; \*t != '\0'; t++) {  str->len++;  str->letters[t - word] = \*t;  }  return \*str;  } | #include "functions.h"  void out(fstream& file, Elem\* first) {  if (!file.is\_open()) {  perror("Error opening file");  exit(1);  }  if (first == NULL) {  file << "The list is empty" << endl;  cout << "The list is empty" << endl;  }  else {  int num = 0;  printel(file, first, num);  num++;  Elem\* curr = first->next;  while (curr != NULL) {  printel(file, curr, num);  curr = curr->next;  num++;  }  file << "-> NULL" << endl;  cout << "-> NULL" << endl;  }  }  void printel(fstream& file, Elem\* el, int num) {  if (num != 0) {  file << "-> ";  cout << "-> ";  }  file << num << ". ";  cout << num << ". ";  for (int i = 0; i < el->str.len; i++) {  file << el->str.letters[i];  cout << el->str.letters[i];  }  file << endl;  cout << endl;  } | #include "functions.h"  Elem\* insert\_before\_last(Elem\* first, Elem\* first\_inserted) {  if (first == NULL) {  return first;  }  if (first\_inserted == NULL) {  return first;  }  Elem\* prelast = first;  Elem\* last = first->next;  if (first->next == NULL) {  last = first;  }  else {  while (last != NULL && last->next != NULL) {  last = last->next;  prelast = prelast->next;  }  }  Elem\* last\_inserted = first\_inserted;  while (last\_inserted->next != NULL) {  last\_inserted = last\_inserted->next;  }  if (first->next != NULL) {  prelast->next = first\_inserted;  }  else {  first = first\_inserted;  }  last\_inserted->next = last;  return first;  } | #include <iostream>  #include "functions.h"  int main()  {  cout << "Author: Novikov Gregory\n"  "Start date: 23.05.2022\n"  "End date: 24.05.2022\n"  "Version: 3.1" << endl << endl;  Elem\* first = in("in.txt");  Elem\* first\_inserted = in("elements.txt");  string out\_filename = "out.txt";  fstream file;  file.open(out\_filename, ios::out);  file << "INPUT" << endl;  cout << "INPUT" << endl;  out(file, first);  file << endl << "ELEMENTS TO INSERT" << endl;  cout << endl << "ELEMENTS TO INSERT" << endl;  out(file, first\_inserted);  first = insert\_before\_last(first, first\_inserted);  file << endl << "OUTPUT" << endl;  cout << endl << "OUTPUT" << endl;  out(file, first);  desintegrate(first);  file << endl << "AFTER DELETION" << endl;  cout << endl << "AFTER DELETION" << endl;  out(file, first);  file.close();  return 0;  } |

# **Результат работы программы**

Вывод в консоль

Вывод в файл

Ввод из файлов

# **Вывод**

В результате выполнения этой программы ознакомился с реализацией списков в языке программирования С++.