**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 |  | Гордиенко М. Ю. |
| Преподаватель |  | Калмычков В.А. |

Санкт-Петербург

2022

Оглавление

[**Исходная формулировка** 3](#_Toc101309100)

[**Контрольный пример** 3](#_Toc101309101)

[**Формальная постановка задачи** 3](#_Toc101309102)

[**Формат хранения данных** 3](#_Toc101309103)

[**Ограничения, обусловленные компьютером** 3](#_Toc101309104)

[**Макет ввода/вывода** 3](#_Toc101309105)

[**Алгоритм решения** 4](#_Toc101309106)

[**Реализация списка** 4](#_Toc101309107)

[**Текс программы** 5](#_Toc101309108)

[**Результат работы программы** 7](#_Toc101309109)

[**Вывод** 7](#_Toc101309110)

## **Исходная формулировка**

Удалить n элементов, начиная с элемента с номером k.

# **Контрольный пример**

\* - ограничитель

Удалить со 2 элемента 3 элемента.

fwefrwve

fws

adwfe

ccefefww

ddd

vdswwdsdc

Ответ:

fwefrwve

fws

vdswwdsdc

# **Формальная постановка задачи**

Дано: список

Удалить: n элементов, начиная с элемента с индексом k

# **Формат хранения данных**

|  |  |  |
| --- | --- | --- |
| Идентификатор типа | Тип | Назначение |
| f, f1 | fstream | Файловые переменные |
| s, next | Elem | Пользовательский тип |
| stroka | char | Строки |
| ogr | char | Ограничители |
| i | int | Счётчик |
| k, a, n | int | Дополнительные переменные |
| length | int | Длинна строки |

# **Ограничения, обусловленные компьютером**

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

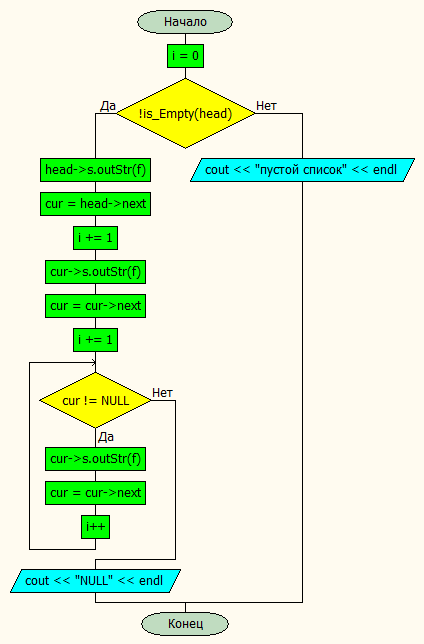
Символьный тип char. Минимальное значение -128, максимальное значение 128.

# **Макет ввода/вывода**

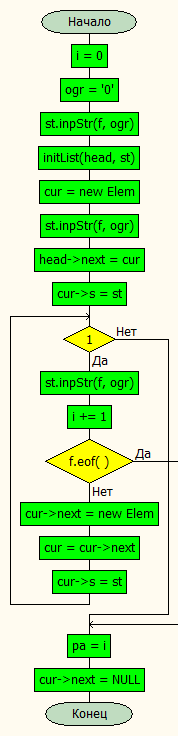
|  |  |
| --- | --- |
| Ввод из файла | f>> |
| Вывод из файла | f1<< |
| Ввод с клавиатуры | cin>> |
| Вывод на экран | cout<< |

# **Алгоритм решения**

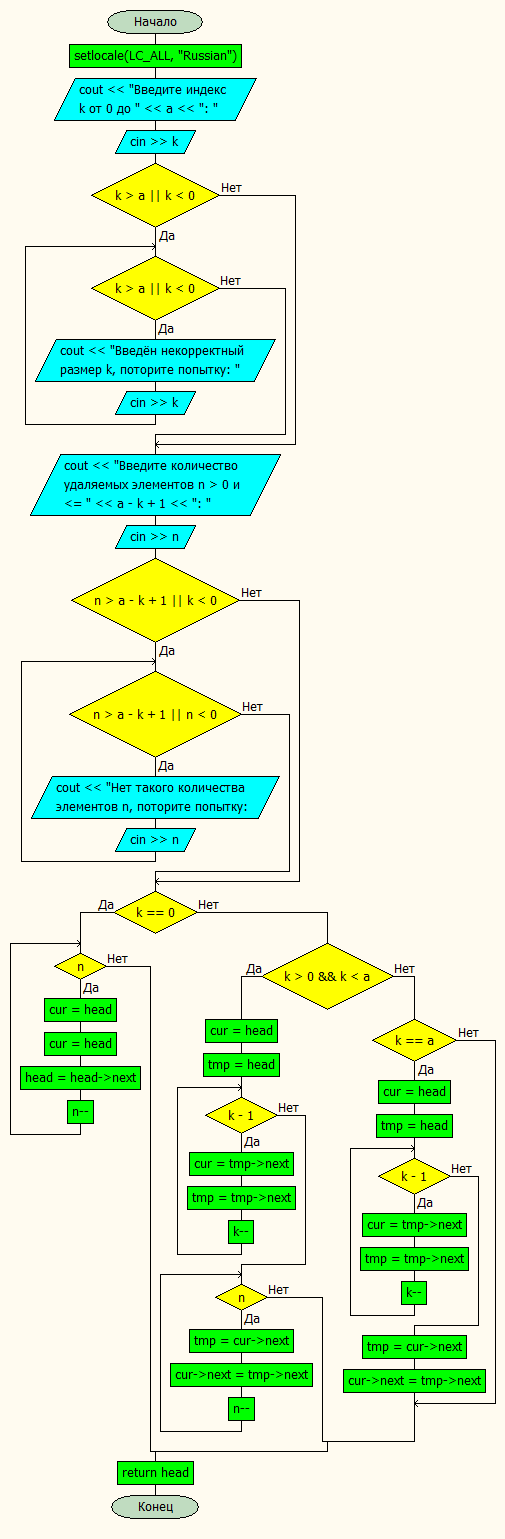
Out()



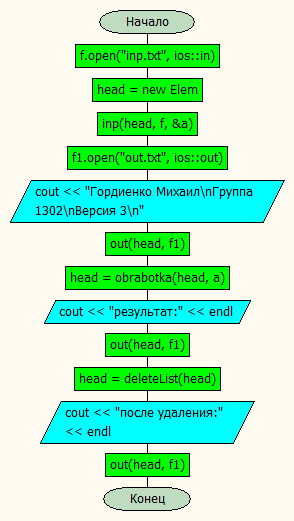
Inp()



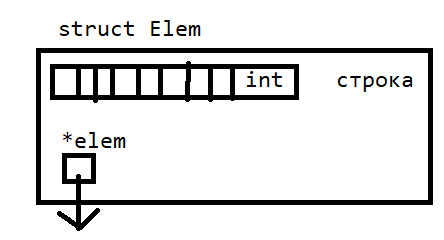
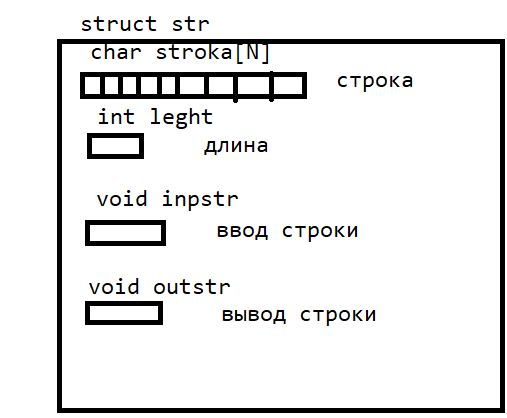
Obrabotka()

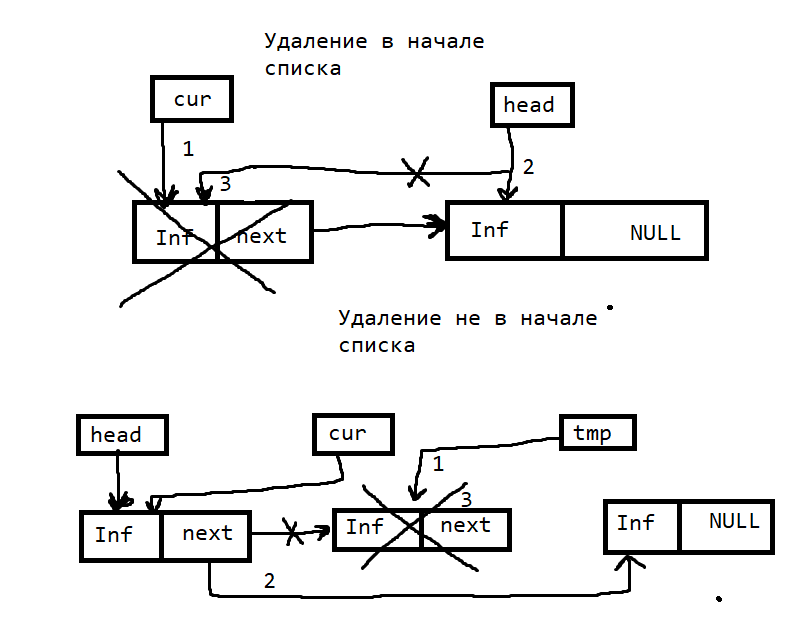
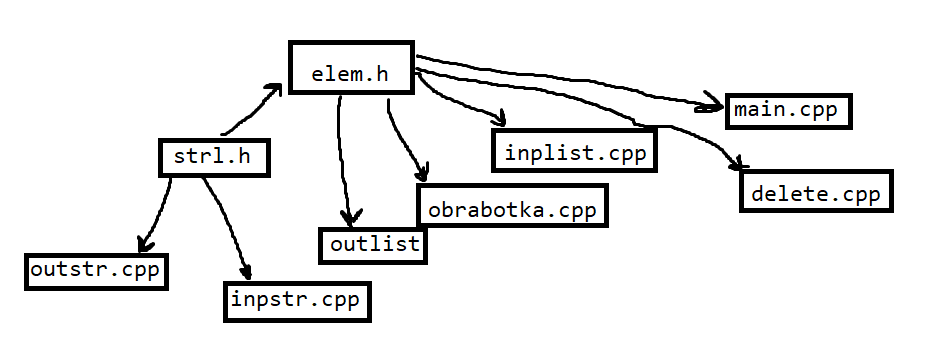


main()



# **Реализация списка**



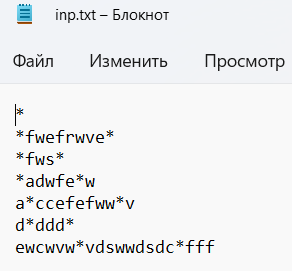
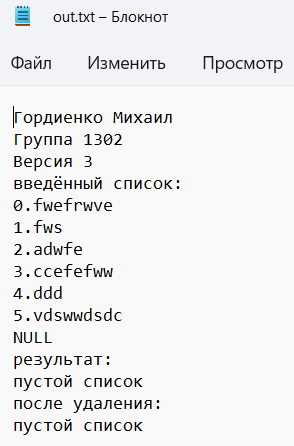


# **Текс программы**

|  |  |  |  |
| --- | --- | --- | --- |
| main | inpstr | inplist | initlist |
| #include"Elem.h"  #include <iostream>  int main()  {  int a;  fstream f, f1;  f.open("inp.txt", ios::in);  Elem\* head;  head = new Elem;  inp(head, f, &a);  f1.open("out.txt", ios::out);  f1 << "Гордиенко Михаил\nГруппа 1302\nВерсия 3\n";  f1 << "введённый список:" << endl;  out(head, f1);  head = obrabotka(head, a);  f1 << "результат:" << endl;  out(head, f1);  head = deleteList(head);  f1 << "после удаления:" << endl;  out(head, f1);  } | #include"strL.h"  void strL::inpStr(fstream& f, char ogr)  {  char c;  unsigned i = 0;  while (f.eof() == 0)  {  f >> c;  if (c == ogr)  {  f >> c;  while (c != ogr && c != '\n' && f.eof() == 0)  {  stroka[i] = c;  f.unsetf(ios::skipws);  f >> c;  i++;  if (i > N)  break;  f.setf(ios::skipws);  }  length = i;  break;  }  else  continue;  }  } | #include"Elem.h"  void inp(Elem\* head, fstream& f, int \*pa)  {  int i = 0;  char ogr = '0';  f >> ogr;  strL st;  st.inpStr(f, ogr);  initList(head, st);  Elem\* cur = new Elem;  st.inpStr(f, ogr);  head->next = cur;  cur->s = st;  while (1)  {  st.inpStr(f, ogr);  i += 1;  if (f.eof())  break;  cur->next = new Elem;  cur = cur->next;  cur->s = st;  }  \*pa = i;  cur->next = NULL;  } | #include"Elem.h"  void initList(Elem\* head, strL st) {  head->s = st;  head->next = NULL;  } |

|  |  |  |  |
| --- | --- | --- | --- |
| outstr | outlist | delete | obrabotka |
| #include"strL.h"  void strL::outStr(fstream& f)  {  if (length == 0)  f << "пустой элемент";  else  {  for (int i = 0; i < length; i++)  {  f << stroka[i];  }  }  f << endl;  } | #include"Elem.h"  void out(Elem\* head, fstream& f)  {  int i = 0;  if (!is\_Empty(head))  {  f << i << '.';  head->s.outStr(f);  Elem\* cur = head->next;  i += 1;  f << i << '.';  cur->s.outStr(f);  cur = cur->next;  i += 1;  while (cur != NULL)  {  f << i << '.';  cur->s.outStr(f);  cur = cur->next;  i++;  }  f << "NULL" << endl;  }  else  f << "пустой список" << endl;  } | #include"Elem.h"  bool is\_Empty(Elem\* head)  {  return head == NULL;  }  Elem\* DeleteHead(Elem\* head)  {  if (!is\_Empty(head))  {  Elem\* p = head->next;  delete head;  return(p);  }  }  Elem\* deleteList(Elem\* head)  {  if (!is\_Empty(head))  {  while (head != NULL)  head = DeleteHead(head);  }  return head;  } | #include"Elem.h"  #include <iostream>  Elem\* obrabotka(Elem\* head, int a)  {  int k, n;  setlocale(LC\_ALL, "Russian");  cout << "Введите индекс k от 0 до " << a << ": ";  cin >> k;  if (k > a || k < 0)  {  while (k > a || k < 0)  {  cout << "Введён некорректный размер k, поторите попытку: ";  cin >> k;  }  }  cout << "Введите количество удаляемых элементов n > 0 и <= "<< a-k+1 << ": ";  cin >> n;  if (n > a - k + 1 || k < 0)  {  while (n > a - k + 1 || n < 0)  {  cout << "Нет такого количества элементов n, поторите попытку: ";  cin >> n;  }  }  if (k == 0)  {  while (n)  {  Elem\* cur = head;  cur = head;  head = head->next;  delete cur;  n--;  }  }  else if (k > 0 && k < a)  {  Elem\* cur = head;  Elem\* tmp = head;  while (k-1)  {  cur = tmp->next;  tmp = tmp->next;  k--;  }  while (n)  {  tmp = cur->next;  cur->next = tmp->next;  delete tmp;  n--;  }  }  else if (k == a)  {  Elem\* cur = head;  Elem\* tmp = head;  while (k - 1)  {  cur = tmp->next;  tmp = tmp->next;  k--;  }  tmp = cur->next;  cur->next = tmp->next;  delete tmp;  }  return head;  } |

# **Результат работы программы**

# **Вывод**

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