|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования FPMI_ngtu_neti_rgb_polya«Новосибирский государственный технический университет» | | | |
|  | | | |
| Кафедра прикладной математики | | | |
| Практическое задание № 1 | | | |
| по дисциплине «Структуры данных и алгоритмы» | | | |
| **«Линейные списки»** | | | |
|  | | | |
|  | Группа | ПМ-21 | |
| Бригада 2 | Егупов иван | |
|  | Щербакова Мария | |
| Сагыдыев Данияр | |
|  | |
|  | |
| Преподаватель | ХИЦЕНКО ВАЛЕНТИНА ПАВЛОВНА | |
| Дата | 19.02. | |
| Новосибирск,2023 | | | |

1. **Условие задачи**

Дана последовательность символов, оканчивающаяся точкой. Написать подпрограмму, которая в двусвязном кольцевом списке с фиктивным звеном удаляет все символы, у которых равные соседи (первый и последний элемент считать соседями). Используя эту подпрограмму, удалить символы, у которых равные соседи.

1. **Анализ задачи**

Входные данные: S = {ai| ai ∈ G, i >=1}, G = **{**`~!@#$%^&\*()-\_+=qwertyuiop{}|asdfghjklzxcvbnm,.<>/,**}**

Выходные данные: S = {ai| ai ∈ G, i >=1 }, G = **{**`~!@#$%^&\*()-\_+=qwertyuiop{}|asdfghjklzxcvbnm,.<>/,**} или** ∅

Решение: **при** x = 1-й элемент S, y = последний элемент   
 если х – единсвенный, то удалить х и S = ∅

Иначе

Если предыдущий от y символ равен x, то удалить y

Иначе Если следующий за х равен y, то удалить x  
 Иначе

**повторять**

если предыдущий символ от x равен следующему за x, то удалить x;  
 иначе х = следущий элемент S  
 **пока** существует х

Подзадачи:

readList – считывание последовательности из файла

входные данные: информация о файле  
выходные данные: последовательность представленная в списке.

deleteElems – удаление элементов, у которых равные соседи.

Входные данные: S

Выходные данные: S или ∅

writeList – Запись измененной последовательности в новый файл

входные данные: видоизмененная последовательность, информация о файле

**3.Структуры данных**

Внешнее представление входных и выходных данных представляет собой последовательность символов, заверщающаяся точкой.

Внутреннее представление: Двусвязный кольцевой список, с фиктивным звеном.

Elem 2

Elem K

L

Elem 1



**Внутреннее представление элементов списка представлено в виде структуры**

struct list {

char elem;

struct list\* next, \* prev;

};

**4.Алгоритм решения задачи**



Ввод S

x = 1-й элемент списка ;y = последний элемент списка

Предыдущий от y равен x?

Удалить y

да

нет

Следующий за х равен y?

Удалить x

да

нет

x = следующий элемент

Если предыдущий символ от х равен следующему от х

нет

да

Существует х?

Удалить x

да

x = следующий элемент последовательности

нет

Вывод S

5.Структура программы

list\* readList(FILE\* file) назначение: считывание информации из файла

FILE \*file – указатель на файл

Возвращает указатель на начало считанной последовательности

void writeList(FILE\* file, list\* ch) назначение: запись измененной последовательности в файл

FILE \*file – указатель на файл

List \*ch – указатель на измененный список

list\* deleteElems(list\* ch) назначение: удаление элементов в поселодовательности, имеющих равных соседей

list \*ch – указатель на начало списка

Возвращает указатель на первый элемент( если удалили начальный первый)

6.Текст программы

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

struct list {

char elem;

struct list\* next, \* prev;

};

list\* readList(FILE\* file)

{

list\* ch=NULL, \* head;

char c;

head = new list;

head->next = NULL;

while ((c = fgetc(file)) != '.')

{

if (head->next == NULL)

{

head->next = new list;

head->next->next = head->next;

head->next->prev = head->next;

head->next->elem = c;

ch = head->next;

}

else

{

ch->next = new list;

ch->next->prev = ch;

ch->next->elem = c;

ch->next->next = head->next;

head->next->prev = ch->next;

ch = ch->next;

}

}

return head;

}

void writeList(FILE\* file, list\* head)

{

list\* p = head->next;

if (p->next == NULL) fprintf(file, " ");

else {

if (p->next == p) fprintf(file, " ");

else do {

fprintf(file, "%c", p->elem);

p = p->next;

} while (p != head->next);

}

}

void deleteElems(list\* head) {

list\* first = head->next, \* p = NULL, \* q = first;

if (first->next == first) return;

else {

do

{

if (q->prev->elem == q->next->elem)

{

if (q == first) first = q->next;

p = q;

q->next->prev = q->prev;

q->prev->next = q->next;

q = q->prev;

delete p;

p = NULL;

}

q = q->next;

} while (q->next!= first);

}

head->next = first;

}

int main()

{

FILE\* read, \* write;

list\* qList;

setlocale(0, "");

fopen\_s(&read, "input.txt", "r");

if (read == NULL) printf\_s("Ошибка");

qList = readList(read);

fclose(read);

fopen\_s(&write, "output.txt", "w");

deleteElems(qList);

writeList(write, qList);

fclose(write);

return 0;

}

7.Набор тестов

|  |  |  |
| --- | --- | --- |
| Входные данные | Выходные данные | назначение |
| S = Qwtyroip. | S = Qwtyroip | Последовательноть латинских букв, среди которых не было удалено ни одного символа |
| S = aaaaaaaaaaaaaa. | S = **∅** | Последовательность состоит из одинаковых символов. |
| S = ubbbereyuyoooptutup. | S = ubbeeyyoopttu | Последовательность с некоторым количеством удалений |
| S = tututututututu. | S = uuuuuuu | Последовательность, чередующихся символов. |
| S = bvhf v reiruei f. | S = bvhf reiruei f | Результат работы программы при вводе последовательности, содержащей пробелы. |
| S = a. | S = **∅** | Простейший тест. |
| S = $%#o#^&\*)\*(>p</cxq. | S = $%##^&\*\*(>p</cxq | Последовательность различных символов и букв. |
| S = abiouirtrab. | S = bioirra | Удаление первого и последнего элемента одновременно |
| S = abaaaaaababbabacccab. | S = aaabbbacct | Последовательность из большого количества подряд идущих символов. |