|  |  |  |
| --- | --- | --- |
|  | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования «Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра прикладной математики | | |
| Практическое задание № 1 | | |
| по дисциплине «Структуры данных и алгоритмы» | | |
| **Линейные списки** | | |
|  | | |
|  | Бригада 7 | Дроздов даниил |
| Группа ПМ-15 | иванова дарья |
| Вариант 4 | Чепурная Алина |
|  |  |
|  |  |
|  |  |
| Преподаватель | Тракимус Юрий Викторович |
|  |  |
| Новосибирск, 2022 | | |

1. **Условие задачи**

Задан текст, состоящий из строк, разделенных пробелом и оканчивающийся точкой. Написать подпрограмму поиска заданного элемента в списке. Если в строке текста содержится заданный символ, то переместить его на место последнего символа в этой строке.

1. **Анализ задачи**

Входные данные: искомый символ и текст, состоящий из строк, разделенных пробелом и оканчивающийся точкой.

Выходные данные: список символов, составленный по исходному тексту, в котором все искомые символы перемещены в конец своих строк.

Для решения задачи определяем структуру элемента list, имеющую два поля: символ elem и указатель на следующий элемент next. Сначала объявляем указатель p, указывающий на первый элемент текста и символ target, который будет использоваться в качестве искомого символа. С помощью функции readFile считываем искомый символ и текст из файла в target и p соответственно. Затем с помощью функции moveToStringEnd переместим все вхождения искомого символа на место последнего символа их строк.

Принцип работы функции moveToStringEnd. Принимаемые аргументы – искомый символ t. Объявляем и инициализируем указатели на элементы: seqHead – первый элемент в последовательности искомых символов, seqEnd – последний элемент в последовательности искомых символов, strEnd – конец строки. Затем в цикле ищем искомый элемент

1. **Программа**

#include <stdio.h>

const char separator = ' ';

const char endChar = '.';

char target = NULL;

struct list {

char elem;

list\* next;

list(char \_elem = NULL, list\* \_next = NULL) : elem(\_elem), next(\_next) {}

list\* getStringEnd() {

if (this == NULL) return NULL;

list\* c = this;

while (c->next != NULL && c->next->elem != separator &&

c->next->elem != endChar)

c = c->next;

return c;

}

void print() {

list\* c = this;

while (c != NULL) {

printf("%c", c->elem);

c = c->next;

}

}

list\* findElem(char t) {

list\* c = this;

while (c != NULL && c->elem != t) c = c->next;

return c;

}

list\* findPrev(list\* h) {

list\* c = h;

if (h == this) return NULL;

while (c != NULL && c->next != this) c = c->next;

return c;

}

void moveTo(list\* t, list\* prev) {

if (prev != NULL) prev->next = next;

next = t->next;

t->next = this;

}

void moveSequenceTo(list\* sEnd, list\* t, list\* prev) {

if (this == sEnd) this->moveTo(t, prev);

else {

if (prev != NULL && sEnd->next != NULL) prev->next =

sEnd->next;

sEnd->next = t->next;

t->next = this;

}

}

list\* getSequenceEnd() {

if (this == NULL) return NULL;

list\* c = this;

while (c->next != NULL && c->next->elem == elem) c = c->next;

return c;

}

list\* moveToStringEnd(char t) {

if (next == NULL) return this;

list\* h = new list(NULL, this);

list\* first = h;

list\* seqHead = this->findElem(t);

list\* seqEnd = seqHead->getSequenceEnd();

list\* strEnd = seqEnd->getStringEnd();

while (seqHead != NULL) {

if (seqEnd != strEnd) {

seqHead->moveSequenceTo(seqEnd, strEnd,

seqHead->findPrev(first));

}

else {

first = strEnd->next;

}

seqHead = first->findElem(t);

seqEnd = seqHead->getSequenceEnd();

strEnd = seqEnd->getStringEnd();

}

first = h->next;

delete h;

return first;

}

};

list\* readFile() {

list\* p = NULL;

list\* current = NULL;

FILE\* fp;

fopen\_s(&fp, "test.txt", "r");

if (fp == 0)

return NULL;

target = fgetc(fp);

fgetc(fp);

char c = fgetc(fp);

while (c != EOF) {

if (current == NULL)

{

p = new list();

current = p;

}

else {

current->next = new list();

current = current->next;

}

current->elem = c;

c = fgetc(fp);

}

fclose(fp);

return p;

}

int main() {

readFile()->moveToStringEnd(target)->print();

return 0;

}

1. **Набор тестов**

|  |  |  |
| --- | --- | --- |
| **№** | **Входные данные** | **Назначение** |
| **1** | a  abbbbb. | Искомый символ стоит в начале строки и текста. |
| **2** | a  bbbbbbba. | Искомый символ стоит в конце строки и текста. |
| **3** | a  bbbbabbb. | Искомый символ стоит в середине строки. |
| **4** | a  bbaaabbb. | Искомые символы стоят друг за другом. |
| **5** | a  bbbbbb. | Искомого символа нет в тексте. |
| **6** | a  a. | В тексте нет символов кроме искомого символа. |
| **7** | a  . | Пустой текст. |
| **8** | a  abbbbb abbb abbbb. | Искомый символ стоит в начале строки и текста.  В тексте больше одной строки. |
| **9** | a  bbbabbbb bbababbabbabbb aa. | Искомый символ стоит в середине строки.  В тексте больше одной строки.  В строке несколько непоследовательных искомых символов.  В строке нет символов кроме последовательных искомых символов. |
| **10** | A  abbba aabbbba aaabbbbbaaa. | Искомые символы стоят в начале и в конце строки.  В тексте больше одной строки. |

1. **Результаты работы программы**

|  |  |
| --- | --- |
| **№** | **Ввод/вывод программы** |
| **1** | bbbbba. |
| **2** | bbbbbbba. |
| **3** | bbbbbbba. |
| **4** | bbbbbaaa. |
| **5** | bbbbbb. |
| **6** | a. |
| **7** | . |
| **8** | bbbbba bbba bbbba. |
| **9** | bbbbbbba bbbbbbbbbbaaaa aa. |
| **10** | bbbaa bbbbaaa bbbbbaaaaaa. |