|  |  |  |
| --- | --- | --- |
|  | | |
| 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.

1. **Алгоритм решения задачи**

Считать исходный символ и текст. Составить по нему список. Затем разбить список на сегменты, разделенные пробелом. Создать дополнительный пустой список. Для каждого сегмента выполнить следующие действия:

1. Попытаться найти в сегменте элемент с искомым символом. Если такового в данном сегменте нет, то перейти к шагу 6.
2. Добавить найденный элемент в конец дополнительного списка.
3. Для элемента, предшествующего найденному, если такой имеется, назначить следующим элементом тот, что следует после найденного.
4. Убрать связь между найденным элементом и следующим после него элементом, тем самым превратив найденный элемент в конечный элемент дополнительного списка.
5. Перейти к шагу 1.
6. Если дополнительный список пуст, перейти к следующему сегменту.
7. Связать последний элемент дополнительного списка со следующим после конечного элемента данного сегмента элементом.
8. Связать конечный элемент данного сегмента с первым элементом дополнительного списка.
9. Убрать связь дополнительного списка с его первым элементом, тем самым опустошив его.
10. Перейти к следующему сегменту.
11. **Программа**

#include <stdio.h>

const char separator = ' ';

const char endchar = '.';

char c;

struct list {

char elem;

list \*next;

bool isHead;

list(bool \_isHead = false, list\* \_next = NULL, char \_elem = 0) :

elem(\_elem), next(\_next), isHead(\_isHead) {}

list\* FindElem(char \_elem, char stop) {

list\* current = isHead ? next : this;

while (current != NULL && current->elem != \_elem && current->elem != stop)

{

current = current->next;

}

return current == NULL || current->elem == stop ? NULL : current;

}

list\* GetStringEnd()

{

list \*current = this;

while (current->next != NULL &&

current->next->elem != separator &&

current->next->elem != endchar) {

current = current->next;

}

return current;

}

list\* GetPrevious(list \*head)

{

list \*current = head;

while (current != NULL && current->next != this)

{

current = current->next;

}

return current;

}

void Print() {

list \*current = this->isHead ? next : this;

while (current != NULL) {

printf\_s("%c", current->elem);

current = current->next;

}

}

} \*p = new list(true, new list);

int Input()

{

FILE\* pFile;

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

if (pFile == 0) {

return 1;

}

list\* current = p->next;

if (pFile != NULL)

{

c = fgetc(pFile);

fgetc(pFile);

while (fread(&current->elem, sizeof(char), 1, pFile) == 1) {

current->next = new list;

current = current->next;

}

}

fclose(pFile);

return 0;

}

int main()

{

if (Input() != 0)

return 0;

list \*matched = new list(true);

list \*prev = p;

list \*current = p->FindElem(c, endchar);

while (current != NULL && current->elem != endchar) {

while (current != NULL) {

prev = current->GetPrevious(p);

prev->next = current->next;

current->next = NULL;

matched->GetStringEnd()->next = current;

current = prev->FindElem(c, separator);

}

matched->GetStringEnd()->next = prev->GetStringEnd()->next;

prev->GetStringEnd()->next = matched->next;

matched->next = NULL;

current = prev->GetStringEnd()->next->FindElem(c, endchar);

}

p->Print();

return getchar();

}

1. **Набор тестов**

|  |  |  |
| --- | --- | --- |
| **№** | **Входные данные** | **Назначение** |
| **1** | x  x193879183iufhwie. | Искомый символ стоит в начале строки и текста. |
| **2** | x  2948ytoijg34j0x. | Искомый символ стоит в конце строки и текста. |
| **3** | x  13df93x348iwueyf. | Искомый символ стоит в середине строки. |
| **4** | x  123xxxx850gdfjg. | Искомые символы стоят друг за другом. |
| **5** | x  iauhphq1398ut98j. | Искомого символа нет в тексте. |
| **6** | x  x. | В тексте нет символов кроме искомого символа. |
| **7** | x  . | Пустой текст. |
| **8** | x  x123123 x11iwueyt93 x234. | Искомый символ стоит в начале строки и текста.  В тексте больше одной строки. |
| **9** | x  2357x2iejo ijx1x1ixwuxeyt93 xx. | 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** | x  x123x xx12314x xxx23748xxx. | Искомые символы стоят в начале и в конце строки.  В тексте больше одной строки. |

1. **Результаты работы программы**

|  |  |
| --- | --- |
| **№** | **Ввод/вывод программы** |
| **1** | 193879183iufhwiex. |
| **2** | 2948ytoijg34j0x. |
| **3** | 13df93348iwueyfx. |
| **4** | 123850gdfjgxxxx. |
| **5** | iauhphq1398ut98j. |
| **6** | x. |
| **7** | . |
| **8** | 123123x 11iwueyt93x 234x. |
| **9** | 23572iejox ij11iwueyt93xxxx xx. |
| **10** | 123xx 12314xxx 23748xxxxxx. |