|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| 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 | |
| Студент | Егупов иван | |
|  |  | |
|  | |
|  | |
|  | |
| Преподаватель | ХИЦЕНКО ВАЛЕНТИНА ПАВЛОВНА | |
| Дата | 03.03. | |
| Новосибирск,2023 | | | |

1. **Условие задачи**

В текстовом файле дана последовательность чисел. Написать рекурсивную функцию, которая находит минимальный элемент непустого однонаправленного ациклического списка. Используя эту подпрограмму, удалить из заданной последовательности все элементы с минимальным значением.

1. **Анализ задачи.**

Входные данные: S = {ai| ai ∈ R, i >=1},

Выходные данные: S = {ai| ai ∈ R, i >=1 } или ∅

Решение: **min(x) =**

Подзадачи:

**readList** – считывание последовательности из файла

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

**deleteElem** – удаление необходимого элемента

Входные данные: адресс удаляемого элемента

Выходные данные: адресс нового заглавного звена, в случае его удаления

**writeList** – Запись измененной последовательности в новый файл

входные данные: видоизмененная последовательность, информация о файле

**minlist**– поиск минимального элемента числовой последовательности

Входные данные: S

Выходные данные: a ∈ R

1. **Структуры данных**

Внешнее представление входных и выходных данных представляет собой последовательность чисел

Внутреннее представление: односвязный ацикличесий список без фиктивного звена

Elem 2

Elem K

L

Elem 1



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

struct list {

float elem;

struct list\* next;

};

**4. Алгоритм решения задачи**



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)

list\* minList(list\* head) назначение: поиска элемента с наименьшим значением  
list \*head – указатель на начало списка  
Возвращает указатель на минимальный элемент

6.Текст программы

#include <stdio.h>

#include <stdlib.h>

struct list {

struct list\* next;

float elem;

};

FILE\* in, \* out;

list\* readList(FILE\* file)

{

list\* ch = NULL, \* head = NULL;

float a;

while (fscanf\_s(file, "%f", &a) != EOF) {

if (head == NULL)

{

head = new list;

head->elem = a;

ch = head;

}

else

{

ch->next = new list;

ch = ch->next;

ch->elem = a;

}

ch->next = NULL;

}

return head;

}

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

{

list\* head = ch;

if (ch == NULL) fprintf(file, " ");

else {

while (ch->next != NULL)

{

fprintf(file, "%.3f ", ch->elem);

ch = ch->next;

}

fprintf(file, "%.3f", ch->elem);

}

}

float minList(list\* start)

{

if (start->next == NULL) return start->elem;

float m = start->elem;

float t = minList(start->next);

return (m < t) ? m : t;

}

list\* Deleteelem(list\* q, list\* head) {

list\* p = q->next;

list\* ch = head;

if (q == head) {

head = p;

delete q;

}

else {

while (ch->next != q) ch = ch->next;

ch->next = p;

delete q;

}

return head;

}

int main() {

list\* head;

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

if (in == NULL) printf\_s("Error with file\n");

head = readList(in);

fclose(in);

list\* ch = head;

list\* p;

float mine = minList(head);

while (ch != NULL)

{

if (ch->elem == mine)

{

p = ch;

ch = ch->next;

head = Deleteelem(p, head);

}

else ch = ch->next;

}

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

writeList(out, head);

fclose(out);

return 0;

}

7.Набор тестов

|  |  |  |
| --- | --- | --- |
| Входные данные | Выходные данные | Назначение |
| 62.1 |  | Тест с вводом единственного элемента |
| -10 8.75 -65.43 999 -65.43 78.2 50 -65.43 | -10.000 8.750 999.000 78.200 50.000 | Тест с удалением минимального элемента, встречающегося несколько раз |
| -100 8.75 99.9 -65.43 785.2 50.23 -100 | 8.750 99.900 -65.430 785.200 50.230 | Минимальный элемент расположен в начале и в конце списка |
| 67 8.23 78.23 0 8.999 23 0 | 67.000 8.230 78.230 8.999 23.000 | Минимальный элемент расположен в конце и середине списка |