Министерство Образования и Науки Российской Федерации  
Новосибирский Государственный Технический Университет  
Кафедра Прикладной Математики

**Лабораторная работа № 1**

**по дисциплине «Структуры данных и алгоритмы»**

Факультет: ФПМИ

Группа: ПМ-53

Студент: Тябин Егор Алексеевич

Преподаватель: Еланцева Ирина Леонидовна

### Новосибирск 2015

# Условие задачи

Дана последовательность действительных чисел a1, a2, . . . , an ( n >= 2 и

заранее неизвестно). Если последовательность упорядочена по неубыванию, то оставить ее без изменения, иначе получить последовательность an , an-1 , . . . , a1 .

# 2. Анализ задачи

Дано: A = {ai∈R, i= 1̅,̅ ̅n, n∈N, n>=2} Где ai это вещественное число, а n-их количество, минимум 2 элемента.  
Результат: A = ai∈R, i= 1̅,̅ ̅n, n∈N, n2.  
   
Где ai это вещественное число, а n-их количество, минимум 2 элемента.

## Метод решения

Основная задача проверяет последовательность на упорядоченность по возрастанию.

Задача проверяет меньше ли следующий элемент. Если такое происходит, то данный момент запоминается и в последующем, запускается подзадача инвертирования последовательности

Введем подзадачу menyaem(), которая будет инвертировать последовательность.

**Для того, чтобы найти инвертированную последовательность** выполняются следующие действия:  
Работая с последовательностью, делаем первый элемент последним.  
Потом делаем 2 элемент предпоследним.  
Дальше делаем 3 элемент 3-им с конца.  
И так далее до тех пор пока не закончится последовательность.  
То есть последний элемент в итоге станет первым.

## Выделение основных подзадач

1. Ввод исходных данных;
2. Инвертирование списка;
3. Проверка на упорядоченность;
4. Вывод результата.

# 3. Структуры данных, используемые для представления исходных данных и результатов задачи

## Внешнее представление входных данных

Вещественные числа, которые находятся в файле input.txt.

## Внешнее представление выходных данных

Вещественные числа, которые находятся в файле output.txt.

# Внутреннее представление входных данных

Нециклический линейный однонаправленный список a. Без фиктивного звена. Каждое звено имеет 2 поля. 1 поле – вещественное число chislo, 2 поле next – адрес следующего звена. Struct a{float chislo; a \*next} – структурный тип a, где его составляющие chislo-Вещественное число. \*next-указатель.

## Внутреннее представление данных

\*L, \*curr – указатели структурного типа a. L-указывает на начало списка. Curr используется для обозначения текущего элемента.  
float t – вещественная переменная, хранящая значение элемента следующего после текущего.  
bool s – переменная типа bool хранящая результат true в случае если последовательность упорядочена, и false в случае если последовательность не упорядочена.  
FILE \*fp – указатель на поток файлов.  
\*sled, \*pred – указатели структурного типа a, где sled-следующий элемент, pred-предыдущий элемент. Область действия - подзадача menyaem().

Результат представлен в виде последовательности, где элементы этой последовательности – вещественные переменные. Количество элементов последовательности - n, где n – количество исходных вещественных чисел.   
Минимальное количество элементов последовательности - 2.

# 4. Укрупненный алгоритм решения задачи

Menyaem()

Начало

S=true;  
L = new a;

curr = L;  
t=curr->next

Curr=L; pred=NULL;

Sled=curr->next

curr->next = новый элемент

(curr->next) = pred;

pred = curr;

curr = sled;

sled = (sled->next);

Curr->chislo>t

Файл закончился?

S=false

sled->next = curr;

curr->next = pred;

L = sled;

curr = (curr->next);

curr->chislo = t;

Конец

Файл закончился?

curr->next = NULL;

S=false

Конец

Menyaem()

Вывод А

Этот алгоритм прост в понимании и не требует удаления лишних элементов так как их нет. Алгоритм действует по принципу: поменять направление списка, в этом заключается его легкость.

# Структура программы

## Взаимосвязь функций

menyaem()

Void main()

## Составные части программы

Наименование функции:   
menyaem  
Прототип функции:  
void menyaem();  
Данная подпрограмма инвертирует список, начало которого в L.

# Текст программы

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

struct a{ float chislo; a\*next; } \*L, \*curr;

void menyaem(){

a \*sled, \*pred;

curr = L;

pred = NULL;

sled = curr->next;

while (sled->next) {

(curr->next) = pred;

pred = curr;

curr = sled;

sled = (sled->next);

}

sled->next = curr;

curr->next = pred;

L = sled;

}

void main(){

FILE \*fp;

float t;

bool s = true;

if (fp = fopen("input.txt", "r")) {

L = new a;

curr = L;

fscanf(fp, "%f", &(curr->chislo));

while (fscanf(fp, "%f", &t) != EOF) {

curr->next = new a;

if (curr->chislo > t)

s = false;

curr = (curr->next);

curr->chislo = t;

}

curr->next = NULL;

fclose(fp);

if (s == false)

menyaem();

if (fp = fopen("output.txt", "w")) {

curr = L;

while (curr != NULL) {

fprintf(fp, "%f ", curr->chislo);

curr = (curr->next);

}

fclose(fp);

}

else perror("error");

}

else perror("error");

}

# 7.Тесты

|  |  |  |  |
| --- | --- | --- | --- |
| № | Входные данные | Результат | Примечания |
| 1 | 1 2 3 4 5 6 | 1.000000 2.000000 3.000000 4.000000 5.000000 6.000000 | Проверка на работоспособность. |
| 2 | 1 4 2 3 5 6 | 6.000000 5.000000 3.000000 2.000000 4.000000 1.000000 | Проверка на правильность переворота. |
| 3 | 1.5 1.6 1.7 -1.8 2 3.5 | 3.500000 2.000000 -1.800000 1.700000 1.600000 1.500000 | Проверка на работу с действительными числами. |

# 8. Результат работы программы

-Программа выдала верное решение на всех тестах и, следовательно, правильно работает.