|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования «Новосибирский государственный технический университет» | | | |
|  | | | |
| Кафедра прикладной математики | | | |
|  | | | |
| Практическое дополнительное задание № 3 | | | |
| по дисциплине «Структура данных» | | | |
| **Определение времени работы прикладных программ** | | | |
|  | | | |
|  | Бригада 1 | ПМ-31 Крашенин Александр |
|  | пм-31 Никулищев егор |
|  | ПМ-31 коцелап екатерина |
|  |  |
|  |  |
|  |  |
| Преподаватель | Хиценко Валентина павловна |
|  | | | |
| Новосибирск | | | |

**1 Условие задачи**

Даны действительные числа x1, x2, ... , xn (n ⩾ 2 и заранее неизвестно). Вычислить: x1xn + x2xn - 1 + ... + xnx1;

**2 Анализ задачи**

* 1. **Входные данные:**

A = {xi ***R***| i = 1, n; n⩾2}

* 1. **Выходные данные:**

Y  *R*

* 1. **Решение задачи:**

*При*: x1 – первый элемент

xn – последний элемент

y = 0

*Повторять*:

y = y + 2\* x1 \* xn

x1 – следующий элемент

xn – предыдущий элемент

i = i +1

*пока* (следующий за x1 ≠ xn) и (x1 ≠ xn)

*если* x1 = xn,

то y = y + x1 \* xn.

**Основные подзадачи:**

1. Подзадача input

Назначение: чтение последовательности из файла, создание соответствующего списка

Входные данные: имя текстового файла  
Выходные данные: A – последовательность n в списке

1. Подзадача calculate

Назначение: Вычисление выражения

Входные данные: список А с последовательностью  
Выходные данные: y  ***R***

**3 Структуры основных входных и выходных данных**

* 1. **Внешнее представление:**

Последовательность действительных чисел разделённых пробелов, оканчивающаяся концом строки

* 1. **Внутреннее представление:**

цикличный двухсвязный линейный список, с указанием от первого к последнему

struct list

{

list\* next, \* prev;

float elem;

};

**4 Алгоритм решения:**

**Подзадача ввода:**

**Назначение:** Чтение полиномов из файла, создание соответствующих списков. **Входные данные:** файл, содержащий последовательность А = { xi | I = 1, n; n >= 2}

**Выходные данные:** А = { xi | i = 1, n; n >= 2}

**Прототип:**

{

s = первое звено;

t = последнее звено

n = s;

while ()

{

n->elem = следующий элемент последовательности

n = следующее звено

}

s->prev = t (предыдущий элемент первому – последний)

}

**подзадача счёта:**

**Назначение:** Вычисление выражения

**Входные данные:** список А с последовательностью

**Выходные данные:** y  ***R***

**Прототип:**

Float y = 0;

a – первый элемент списка;

b – последний элемент списка;

while ((следующий за b != a) && (a != b))

{  
 y = y + 2 \* (a->elem) \* (b->elem);

a = следующие звено;

b = предыдущее звено;

}

If (a == b)

{

y = y + (a->elem) \* (b->elem);

}

**5 ТЕКСТ ПРОГРАММЫ**

#include <stdio.h>

#include <stdlib.h>

struct list

{

list\* next, \* prev;

float elem;

};

void clean(list\* s)

{

list\* k = s->next;

for (; k; k = k->next)

{

delete s;

s = k;

}

delete s;

}

/\*

void check(list\* n)

{

while (n->next != NULL)

{

printf("%f", n->elem);

printf(" ");

n = n->next;

};

printf("%f", n->elem);

}\*/

void save(float num)

{

FILE\* file2;

fopen\_s(&file2, "output.txt", "w");

fprintf(file2, "%f", num);

fclose(file2);

}

float calculate(list\* s)

{

float sum = 0;

list\* x, \* y;

x = s;

y = s->prev;

while ((y->next != x) && (x != y))

{

sum = sum + 2 \* (x->elem) \* (y->elem);

x = x->next;

y = y->prev;

};

if (x == y)

{

sum = sum + (x->elem) \* (y->elem);

}

return sum;

}

void input(char \*fname, list\* s)

{

FILE\* file1;

list\* t, \*n;

char c = ' ';

fopen\_s(&file1, fname, "r");

if (!file1)

{

printf("ERROR!!! FILE DOES NOT FOUNDED");

exit(1);

}

n = s;

fscanf\_s(file1, "%f", &n->elem);

while (c != EOF)

{

t = new list;

t->prev = n;

n->next = t;

n = n->next;

fscanf\_s(file1, "%f", &t->elem);

c = fgetc(file1);

};

s->prev = n;

n->next = NULL;

fclose(file1);

}

int main()

{

list\* s;

float sum;

s = new list;

char fname[10] = "input.txt";

input(fname, s);

sum = calculate(s);

save(sum);

clean(s);

return 0;

};

**6 ТЕСТЫ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Входные данные** | **Выходные данные** | | **назначение** |
| **1** | **ожидаемые** | **полученные** |
| **2** | 1 1 | 2 | 2 | Простой текст с минимальным количеством элементов |
| **3** | 1 2 3 4 | 20 | 20 | Чётное количество элементов |
| **4** | 1 2 3 4 5 | 35 | 35 | Нечётное количество элементов |