**Минобрнауки России**

**Санкт-Петербургский Государственный**

**Электротехнический Университет**

**«ЛЭТИ» Им. В.И. Ульянова (ЛЕНИНА)**

**Кафедра Мо Эвм**

**Отчет**

**По Лабораторной Работе 5**

**По Дисциплине «Программирование»**

**Тема: Индуктивные Функции. Обработка**

**Файлов**

|  |  |  |
| --- | --- | --- |
| Студент(ка) гр. 6382 |  | Вайгачев А.О. |
| Преподаватель |  | Шолохова О.М. |

**Цель работы.**

Изучение однопроходных алгоритмов обработки последовательностей на

примере индуктивных функций. Обработка файлов.

**Постановка задачи.**

При формулировке задания используются следующие обозначения: Z –

целые числа, N – натуральные, N0 – натуральные с нулем, R – вещественные, R0 – неотрицательные вещественные, R+ – положительные вещественные, S – символы, B – логический тип, R2 = R×R (декартово произведение).В формулировке каждого задания указана функция f(**ω**), а также типы X и Y, такие, что f: **Ω**(X) →Y

Вариант 7: Число элементов числовой последовательности, больших всех

предыдущих элементов, f: **Ω** (N) → N0 .

**Основные теоретические положения.**

Для того, чтобы реализовать данную неиндуктивную функцию, построим индуктивное расширение для этой функции.

F : **π** (N,max(N)) = if (N > max(N)) then N0 =N0 + 1

Для удобства работы с последовательностью значения берутся из файла “input.txt”.

**Спецификация программы.**

*Назначение программы*. Программа предназначена для вывода числа элементов числовой последовательности, больших всех предыдущих элементов, f: **Ω** (N) → N0 .

*Описание программы*. Программа написана на языке C с использованием компилятора gcc 5.4.0. Входные данные для программы является последовательность чисел, которые берутся из файла “input.txt”.Если этот фаил отсутствует, то программа завершает работу с ошибкой. Проверка значений, вводимых пользователем отсутствует, поведение программы при некорректном вводе не определено. Выходными данными являются: значение N0.

*Пример диалога с пользователем*.

>> ./a.out

>> 2

*Реализация*.

Глобальные переменные:

1. unsigned int max\_from\_all; - используется для запоминания наибольшего

значения последовательности.

Функции:

1. void test(FILE \*f) – используется для проверки. Программа прерыватся если фаил “input.txt” - не найден.

Входные параметры:

Адрес файла “input.txt”.

2. int omega(FILE \*f) – является надстроенной индуктивной функцией исходной ф.

Входные параметры:

Адрес файла “input.txt”.

**Тестирование.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Последовательность | 1,1,1,1,1 | 1,2,1,1,1,1 | 1,2,3,44,5,6 | 666,66,6,1 |
| Значение | 0 | 1 | 3 | 0 |

**Выводы.**

Изучили однопроходные алгоритмы обработки последовательностей на

примере индуктивных функций. Научился открывать и считывать значения из файла в программе.

Приложение.

**#include <stdio.h>**

**#include <stdlib.h>**

**/\***

**Вариант 7: Число элементов числовой последовательности, больших всех**

**предыдущих элементов**

**\*/**

**void test(FILE \*f){**

**if(f == NULL){**

**printf("There is no file here\n");**

**exit(EXIT\_FAILURE);**

**}**

**}**

**unsigned int max\_from\_all = 0;**

**int omega(FILE \*f){**

**unsigned int s = 0;**

**unsigned int i = 0;**

**if (fscanf(f,"%d",&i) != EOF) max\_from\_all = i;**

**while(fscanf(f,"%d",&i) != EOF){**

**if(i > max\_from\_all){**

**max\_from\_all = i;**

**s += 1;**

**}**

**}**

**return s;**

**}**

**int main(){**

**FILE \*f = fopen("input.txt","r");**

**test(f);**

**unsigned int s;**

**s = omega(f);**

**printf("%d",s);**

**return 0;**

**}**