**МИНОБРНАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ТУЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Институт прикладной математики и компьютерных наук

Кафедра информационной безопасности

**ЯЗЫКИ ПРОГРАММИРОВАНИЯ**

Отчет по выполнению лабораторной работы № 2

Выполнила \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ст. гр.230711 Павлова Виктория Сергеевна

Проверила \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

доц. каф. ИБ Басалова Галина Валерьевна

Тула 2022

# Лабораторная работа № 2. Работа с массивами, строками и файлами в Паскале

## Цель работы:

Повторение принципов использования структурированных типов данных языка Паскаль.

## Задания на работу:

1. Изучить теоретические положения и конспект лекций по данной теме.

2. Проверить работу программ из примеров 1-4.

3. Разработать программу, читающую данные из произвольного текстового файла (имя файла вводится пользователем, содержимое файла – одна или несколько строк); разработать алгоритмы и функции для выполнения пунктов задания по варианту. Проверить работу программы на различных тестах.

4. Показать результаты работы программы преподавателю.

5. Составить отчет по работе: для каждой задачи по своему варианту: описание разработанного алгоритма (постановка задачи, исходные данные, требуемые результаты, схема алгоритма); текст созданной программы; результаты работы программы для разных исходных данных, список использованных стандартных процедур и функций для работы со строками и файлами.

## Ход работы:

1. Проверю работу программы из *примера 1.*

Входные данные: массив X, состоящий из 4 строк и 5 столбцов.   
Выходные данные: номер строки с максимальным элементом массива.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Тест программы №1. Входные данные: массив с элементами   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 5 | 4 | 3 | 2 | 5 | | 6 | 7 | 7 | **8** | 5 | | 5 | 4 | 3 | 2 | 1 | | 5 | 6 | 7 | 4 | 3 |   Ожидаемый результат: 2 Результат работы программы: | Тест программы №2. Входные данные: массив с элементами     |  |  |  |  |  | | --- | --- | --- | --- | --- | | **435** | 32 | 65 | 23 | 434 | | 435 | 232 | 22 | 1 | 32 | | 44 | 5 | 33 | 22 | 53 | | 10 | 2 | 8 | 3 | 4 |   Ожидаемый результат: 1 Результат работы программы: | Тест программы №3. Входные данные: массив с элементами   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 70 | 17 | 75 | 55 | 25 | | 43 | 4 | 0 | 0 | 3 | | 44 | 5 | 6 | 66 | 71 | | 17 | 20 | **600** | 34 | 44 |   Ожидаемый результат: 4 Результат работы программы: |

Программа из *примера 1* работает корректно.

Проверю работу программы из *примера 2.*

Входные данные: N целых чисел, записываемых в пустой файл Primer1.txt  
Выходные данные: файл Primer1.txt, содержащий введённые числа.

|  |  |  |
| --- | --- | --- |
| Тест №1. Входные данные: N=5 и числа 2,4,6,8,10.  Ожидаемый результат: 2 4 6 8 10  Результат работы программы: | Тест №2. Входные данные: N=10, все числа нулевые  Ожидаемый результат: 0 0 0 0 0 0 0 0 0 0  Результат работы программы: | Тест №3. Входные данные: N=3, числа 1000, 2020, 2021  Ожидаемый результат: 1000 2020 2021  Результат работы программы: |

Программа из *примера 2* работает корректно.

Проверю работу программы из *примера 3.*

Входные данные: N целых чисел в файле Primer1.txt  
Выходные данные: S – сумма чисел, записанных в файле.

|  |  |  |
| --- | --- | --- |
| Тест №1. Входные данные: N=5 и числа 2,4,6,8,10.  Ожидаемый результат: S=30  Результат работы программы: | Тест №2. Входные данные: N=10, все числа нулевые  Ожидаемый результат: S=0  Результат работы программы: | Тест №3. Входные данные: N=3, числа 1000, 2020, 2021  Ожидаемый результат: S=5041.  Результат работы программы: |

Программа из *примера 3* работает корректно.

Проверю работу программы из *примера 4.* Входные данные: файл Text.txt, содержащий некоторый текст Выходные данные: количество строк, записанных в тексте.

|  |  |  |
| --- | --- | --- |
| Тест №1. Входные данные: текст из 6 строк Ожидаемый результат: 6  Результат работы программы: | Тест №2. Входные данные: текст из 4 строк Ожидаемый результат: 4  Результат работы программы: | Тест №3. Входные данные: пустой файл  Ожидаемый результат: 0  Результат работы программы: |

Программа из *примера 4* работает корректно

1. Задания по варианту:

**№36.** На вход программе подаются сведения о пассажирах, желающих сдать свой багаж в камеру хранения на заранее известное время до полуночи. В первой строке сообщается число пассажиров N, которое не меньше 3, но не превосходит 1000; во второй строке – количество ячеек в камере хранения K, которое не меньше 10, но не превосходит 1000. Каждая из следующих N строк имеет следующий формат: <фамилия>, <время сдачи багажа>, <время освобождения ячейки>, где <фамилия>, – строка, состоящая не более чем из 20 непробельных символов; <время сдачи багажа> – через двоеточие два целых числа, соответствующие часам (от 00 до 23 – ровно 2 символа) и минутам (от 00 до 59 – ровно 2 символа); <время освобождения ячейки> имеет тот же формат. Время освобождения больше времени сдачи. Сведения отсортированы в порядке времени сдачи багажа.

Каждому из пассажиров в камере хранения выделяется свободная ячейка с минимальным номером. Если в момент сдачи багажа свободных ячеек нет, то пассажир уходит, не дожидаясь освобождения одной из них. Требуется написать программу, которая будет выводить на экран для каждого пассажира номер ему предоставленной ячейки (можно сразу после ввода данных очередного пассажира). Если ячейка пассажиру не предоставлена, то его фамилия не печатается. Пример входных данных:

3

10

Иванов 09:45 12:00

Петров 10:00 11:00

Сидоров 12:00 13:12

|  |  |  |
| --- | --- | --- |
| **Тест программы №1:**  Входные данные: n=3, k=10  Иванов 09:45 12:00  Петров 10:00 11:00  Сидоров 12:00 13:12  Ожидаемый результат:  Иванов: ячейка №1  Петров: ячейка №2  Ячейка №2 освободилась  Сидоров: ячейка №2  Результат работы программы: | **Тест программы №2:**  Входные данные: n=5, k=3  Иванов 09:45 12:00  Петров 10:00 11:00  Сидоров 12:00 13:12  Лавочкин 12:01 19:00  Скамейкин 12:30 17:00  Ожидаемый результат:  Иванов: ячейка №1  Петров: ячейка №2  Ячейка №2 освободилась  Сидоров: ячейка №2  Лавочкин: ячейка №3  (последнему не хватило)  Результат работы программы: | **Тест программы №3:**  Входные данные: n=4, k=4  Булочкин 09:45 12:00  Плюшкин 10:00 11:00  Пирожков 12:00 12:12  Пончик 12:13 19:00  Ожидаемый результат:  Булочкин: ячейка №1  Плюшкин: ячейка №2  Ячейка №2 освободилась  Пирожков: ячейка №2  Ячейка №2 освободилась  Пончик: ячейка №2  Результат работы программы: |

|  |  |
| --- | --- |
| Схема программы: |  |

**var** n,k,i,cout,t: integer;

p:**array of** (string,string,string,integer);

str,s1,s2,s3:string;

cort: (string,string,string,integer);

f: text;

**begin**

assign(f,'input36.txt');

reset(f);

readln(f,n); //пассажиров

readln(f,k); //ячеек

setlength(p,n);

cout:=1;

**for** i:=0 **to** n-1 **do**

**begin**

readln(f,str);

cout:=1;

s1:=readwordfromstring(str,cout);

s2:=readwordfromstring(str,cout);

s3:=readwordfromstring(str,cout);

cort:=(s1,s2,s3,t);

p[i]:=cort;

**end**;

close(f);

assign(f,'output36.txt');

rewrite(f);

//присваивание первого номера

cout:=1;

t:=cout;

cort:=(p[0].Item1,p[0].Item2,p[0].Item3,t);

p[0]:=cort;

writeln(p[0].Item1,' получил номер ячейки ',t);

//цикл сравнивания

**for** i:=1 **to** n-1 **do begin**

**if** i<=k **then begin**

**if** p[i-1].Item3 > p[i].Item2 **then**

**begin**

inc(cout);

t:=cout;

cort:=(p[i].Item1,p[i].Item2,p[i].Item3,t);

p[i]:=cort;

writeln(p[i].Item1,' получил номер ячейки ',t);

writeln(f,p[i].Item1,' получил номер ячейки ',t);

**end**

**else**

**begin**

writeln('--> ячейка номер ',cout, ' освободилась');

writeln(f,'ячейка номер ',cout, ' освободилась');

t:=cout;

cort:=(p[i].Item1,p[i].Item2,p[i].Item3,t);

p[i]:=cort;

writeln(p[i].Item1,' получил номер ячейки ',t);

writeln(f,p[i].Item1,' получил номер ячейки ',t);

**end**;

**end**

**else**

**begin**

t:=0;

cort:=(p[i].Item1,p[i].Item2,p[i].Item3,t);

p[i]:=cort;

**end end**;

close(f);

**end**.

**Задача №39.** Даны два файла, в каждом из которых записаны неубывающие последовательности целых чисел, первый из N элементов, второй из M элементов. Образовать из них новую последовательность чисел так, чтобы она тоже была неубывающей. Вывести ее в новый файл и на экран. Не использовать sort.

|  |  |  |
| --- | --- | --- |
| **Тест программы №1:**  Входные данные: целые N=2, M=3, последовательности соответственно  2 3  3 4 5  Ожидаемый результат:  2 3 3 4 5  Результат работы программы: | **Тест программы №2:**  Входные данные: целые N=4, M=5, последовательности соответственно  0 0 1 2  0 0 0 1 1  Ожидаемый результат:  0 0 0 0 0 1 1 1 2  Результат работы программы: | **Тест программы №3:**  Входные данные: целые N=7, M=3, последовательности соответственно  1 2 2 3 5 7 9  0 1 2  Ожидаемый результат:  0 1 1 2 2 2 3 5 7 9  Результат работы программы: |

**Код программы:**

|  |  |
| --- | --- |
| **var** x,y,i,n,m:integer;  f1,f2,f3:text;  **begin**  assign(f1,'input39.1.txt');  reset(f1);  assign(f2,'input39.2.txt');  reset(f2);  assign(f3,'output39.txt');  rewrite(f3);  read(f1,n); read(f2,m);  read(f1,x); read(f2,y);    **for** i:=1 **to** m+n **do begin**  **if** x>y **then begin**  write(y,' '); write(f3,y,' ');  **if not** eof(f2) **then** read(f2,y)  **else break**; **end**  **else begin if** y>x **then begin**  write(x,' '); write(f3,x,' ');  **if not** eof(f1) **then** read(f1,x)  **else break**; **end**  **else begin**  write(x,' '); write(y,' '); write(f3,x,' '); write(f3,y,' ');  **if** (eof(f1)) **or** (eof(f2)) **then begin**  **if** eof(f1) **then begin**  read(f2,y); **break**; **end**  **else begin** read(f1,x); **break**; **end**; **end** | **else begin** read(f1,x); read(f2,y); **end**;  **end**; **end**; **end**;  **if** eof(f1) **then begin**  **if** x<y **then begin while not** eof(f2) **do begin**  write(y,' '); write(f3,y,' '); read(f2,y); **end**;  write(y,' '); write(f3,y,' '); **end**  **else begin** write(x,' ');  **while not** eof(f2) **do begin**  read(f2,y);  write(y,' '); write(f3,y,' '); **end**; **end**; **end**  **else if** eof(f2) **then begin**  **if** y<x **then begin**  **while not** eof(f1) **do begin**  write(x,' '); write(f3,x,' ');  read(f1,x); **end**;  write(x,' '); write(f3, x,' ');  **end**  **else begin** write(y,' '); write(f3, y,' ');  **while not** eof(f1) **do begin**  read(f1,x);  write(x,' '); write(f3, x,' ');  **end**;  **end**;  **end**;  close(f1);  close(f2); close(f3); **end**. |

**Схема программы:**



