Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

«Алтайский государственный технический университет им. И. И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Отчет защищен с оценкой \_\_\_\_\_

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2024г.

Отчет

По лабораторной работе №6

**«Файлы»**

по дисциплине «Программирование – 2 семестр»

Студент группы ПИ-32Б Самохвалов Даниил Алексеевич

Преподаватель доцент, к.т.н. Егорова Е.В.

Барнаул 2024

N5.

Создать файл f, компоненты которого являются целыми числами. Никакая из компонент файла не равна 0. Числа в файле записаны в следующем порядке: 5 положительных, 5 отрицательных и т.д. Число компонентов файла f делится на 20.

Переписать компоненты файла f в файл g так, чтобы в файле g числа шли в следующем порядке: ++ -- ++ -- … , то есть два положительных числа, два отрицательных числа и т.д. Допустимо использование вспомогательного файла.

**Текст программы:**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <ctime>**

**#define AMOUNT\_NUMBERS 20 // кратно 20**

**void fillOriginFile(FILE \* pf)**

**{**

**srand(time(NULL));**

**int posNumber = rand() % 100 + 1;**

**int negNumber = -1 \* (rand() % 100 + 1);**

**for (int i = 0; i <= (int)(AMOUNT\_NUMBERS / 20) ; ++i)**

**{**

**for (int i\_p = 0; i\_p < 5; ++i\_p,**

**posNumber = rand() % 100 + 1)**

**fprintf(pf, "%d ", posNumber);**

**for (int i\_n = 0; i\_n < 5; ++i\_n,**

**negNumber = -1 \* (rand() % 100 + 1) )**

**fprintf(pf, "%d ", negNumber);**

**}**

**}**

**void outputDataFile(FILE \* pf)**

**{**

**int numberfFile;**

**for (int i = 0; i < AMOUNT\_NUMBERS; ++i)**

**{**

**fscanf(pf, "%d", &numberfFile);**

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

**}**

**}**

**void sortAndWriteNumberfFile(int in [AMOUNT\_NUMBERS], FILE \* from)**

**{**

**int fromfNumber;**

**int id\_pos = 0, id\_neg = AMOUNT\_NUMBERS / 2;**

**for (int i = 0; i < AMOUNT\_NUMBERS; ++i) {**

**fscanf(from, "%d", &fromfNumber);**

**if (fromfNumber > 0)**

**in[id\_pos++] = fromfNumber;**

**else**

**in[id\_neg++] = fromfNumber;**

**}**

**}**

**void fillResultFile(FILE \* in, FILE \* from)**

**{**

**int fromfData [AMOUNT\_NUMBERS] = {0};**

**sortAndWriteNumberfFile(fromfData, from);**

**int i\_p = 0, i\_n = AMOUNT\_NUMBERS/2;**

**for (int i = 2; i <= (int)(AMOUNT\_NUMBERS / 2); i += 2)**

**{**

**for (; i\_p < i; ++i\_p)**

**fprintf(in, "%d ", fromfData[i\_p]);**

**for (; i\_n < AMOUNT\_NUMBERS/2 + i; ++i\_n)**

**fprintf(in, "%d ", fromfData[i\_n]);**

**}**

**}**

**int main()**

**{**

**FILE \* ptrFile = fopen("origin.txt", "w");**

**FILE \* ptrOutFile = fopen("result.txt", "w");**

**printf("\nOrigin file:");**

**fillOriginFile(ptrFile);**

**fclose(ptrFile); // Записывем в файл**

**ptrFile = fopen("origin.txt", "r");**

**outputDataFile(ptrFile);**

**printf("\n");**

**fseek(ptrFile, 0, SEEK\_SET); // сброс указателя**

**printf("\nResult file:");**

**fillResultFile(ptrOutFile, ptrFile);**

**fclose(ptrOutFile); // Записывем в файл**

**ptrOutFile = fopen("result.txt", "r");**

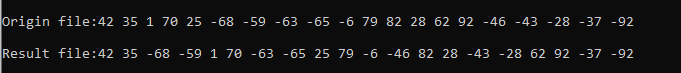
**outputDataFile(ptrOutFile);**

**printf("\n");**

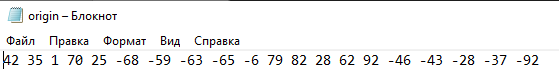
**}**

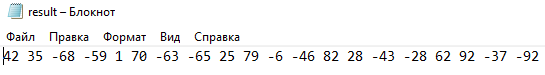
**Тесты программы:**

* 1. **Вывод в консоль**

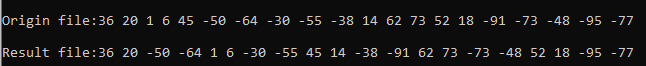
****

* 1. **Вывод в файлы**

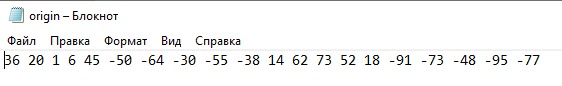
****

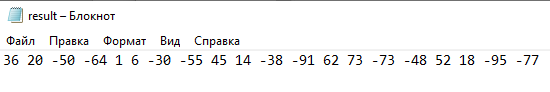
****

**2.1) Вывод в консоль**

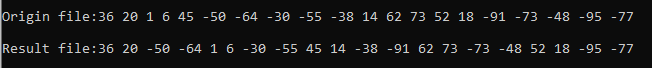
****

**2.2)Вывод в файлы**

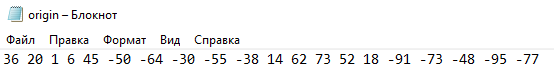
****

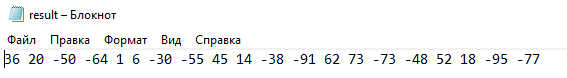
****

**3.1) Вывод в консоль**

****

**3.2) Вывод в файлы**

****

****

N10.

Дан символьный файл (создать его), содержащий слова. Слова в тексте разделяются пробелами и знаками препинания.

Получить 2 наиболее часто встречающихся слова и число их повторений. Если таких слов более 2, выдать соответствующую информацию для каждого из них.

**Текст программы:**

**#include <stdio.h>**

**#include <string.h>**

**#define MAX\_TEXT\_SIZE 255**

**#define TEXT "q a b a v d"**

**#define SEPARATORS " -,/.!?():;\'\""**

**void fillFile(FILE \* f)**

**{**

**fputs(TEXT, f);**

**}**

**void outputDataFile(FILE \* f)**

**{**

**char chff;**

**while ( (chff = fgetc(f)) != EOF )**

**putchar(chff);**

**}**

**char findWord(char words[MAX\_TEXT\_SIZE/2][MAX\_TEXT\_SIZE], int amountWords, char word [] )**

**{**

**for (int i = 0; i < amountWords; ++i)**

**if (stricmp(words[i], word) == 0) return 1;**

**return 0;**

**}**

**void freadCharsTo(char a[], FILE \* f)**

**{**

**char chff;**

**int amountCh = 0;**

**while ( (chff = fgetc(f)) != EOF )**

**a[amountCh++] = chff;**

**}**

**void countWordsAndWrite(char words[MAX\_TEXT\_SIZE/2][MAX\_TEXT\_SIZE],**

**int resultAmount[MAX\_TEXT\_SIZE/2], int amountLex)**

**{**

**for (int i = 0; i < amountLex; ++i)**

**for (int j = 0; j < amountLex; ++j)**

**if (stricmp(words[i], words[j]) == 0)**

**resultAmount[i] += 1;**

**}**

**int findMax(int a[MAX\_TEXT\_SIZE/2], int len)**

**{**

**int max = a[0];**

**for (int i = 0; i < len; i++)**

**if (a[i] > max) max = a[i];**

**return max;**

**}**

**void outResult(FILE \* f)**

**{**

**char wordsf [MAX\_TEXT\_SIZE/2][MAX\_TEXT\_SIZE];**

**char outputedWordsf [MAX\_TEXT\_SIZE/4][MAX\_TEXT\_SIZE];**

**char strfF [MAX\_TEXT\_SIZE];**

**int amountOfWordsInText[MAX\_TEXT\_SIZE/2] = {0};**

**int amountLex = 0;**

**// Считывание**

**freadCharsTo(strfF, f);**

**// Разбиение на слова**

**char \* word = strtok(strfF, SEPARATORS);**

**while (word != NULL)**

**{**

**strcpy(wordsf[amountLex++], word);**

**word = strtok(NULL, SEPARATORS);**

**}**

**// Подсчет каждого слова**

**countWordsAndWrite(wordsf, amountOfWordsInText, amountLex);**

**// Нахождение максимального**

**int max = findMax(amountOfWordsInText, amountLex);**

**// Вывод слов, если они упоминаются в тексте максимальное кол-во раз раз**

**int amountOutLex = 0;**

**printf("\n\nMet most times:");**

**for (int i = 0; i < amountLex; ++i) {**

**if (amountOfWordsInText[i] == max**

**&& !findWord(outputedWordsf, amountOutLex, wordsf[i])) {**

**strcpy(outputedWordsf[amountOutLex++], wordsf[i]);**

**printf("\n%s, amount: %d", wordsf[i], amountOfWordsInText[i]);**

**}**

**}**

**if (amountOutLex == 1) puts("\n--- Only one word");**

**}**

**int main()**

**{**

**FILE \* ptrFileW = fopen("out2.txt", "w");**

**FILE \* ptrFileR = fopen("out2.txt", "r");**

**printf("\nOrigin file: ");**

**fillFile(ptrFileW);**

**fclose(ptrFileW);**

**outputDataFile(ptrFileR);**

**fseek(ptrFileR, 0, SEEK\_SET); // сброс указателя**

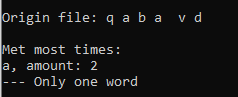
**outResult(ptrFileR);**

**}**

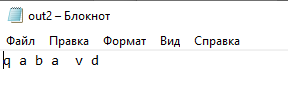
**Тесты программы:**

1. **Текст: “q a b a v d”**

**Вывод в консоль**

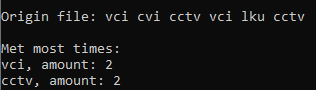
****

**Вывод в файл**

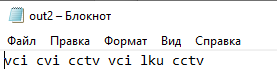
****

1. **Текст: “vci cvi cctv vci lku cctv”**

**Вывод в консоль**

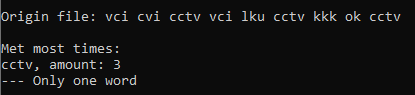
****

**Вывод в файл**

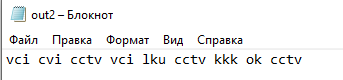
****

1. **Текст “vci cvi cctv vci lku cctv kkk ok cctv”**

**Вывод в консоль**

****

**Вывод в файл**

****