Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа 18

По дисциплине «Основы алгоритмизации и программирования»

На тему «Работа с файлами на языке С»

Выполнил:

Студент 1 курса 6 группы

Кучерук Николай Петрович

Преподаватель: асс. Андронова М.В.

2023, Минск

**Задание**

| **№ варианта** | **Условия задач** |
| --- | --- |
| **9** | 1. Компоненты файла **fileA** –  целые числа, значения которых повторяются.  Получить файл **fileB**, образованный из **fileA** исключением повторных вхождений одного и того же числа.  2. Создать текстовый файл **F1** не менее, чем из 4 строк, и записать в него информацию. Скопировать из файла **F1** в файл **F2** строки, количество символов в которых больше, чем заданное число, которое вводится с клавиатуры. |

|  |
| --- |
| **Код программы** |
| #include <stdio.h>  #include<iostream>  #include <locale>  using namespace std;  int main()  {  srand(time(0));  setlocale(LC\_ALL, "");  int a;  int array[15];  errno\_t errf1,errf2;//Тип errno\_t возвращает код возникающей при работе с файлом ошибки.  FILE\* f,\*f2;//создаём указатели на файлы  errf1 = fopen\_s(&f, "file1.txt", "w");//открываем файл для записи  if (errf1 != 0)//предусматриваем возможность ошибки  {  perror("Невозможно создать файл\n");//Функция perror() выводит эту строку значение глобальной переменной ERRNO с сообщением, указанным в параметре функции.  return EXIT\_FAILURE;  }  for (a = 0; a < 15; a ++)  {  fprintf(f, "%d, ", rand() % 10);//генерируем числа в файл  }  printf("Данные записаны в file1.txt\n");  fclose(f);//закрываем файл  errf1 = fopen\_s(&f, "file1.txt", "r");//открываем тот же файл в режиме чтения  if (errf1 != 0)  {  perror("Невозможно создать файл\n");  return EXIT\_FAILURE;  }  errf2 = fopen\_s(&f2, "file2.txt ", "w");//открываем второй файл в режиме записи  if (errf2 != 0)  {  perror("Невозможно открыть файл\n");  return EXIT\_FAILURE;  }  for (a = 0; a < 15; a++)  {  fscanf\_s(f, "%d, ",&array[a]);//считываем числа из первого файла в массив  }  cout << "Содержимое file1.txt"<<endl;  for (int i = 0; i < 15; i++) {  cout << array[i]<<", ";  }  int arr\_size = 15;  for (int i = 0; i < arr\_size; i++) {//удаляем повторяющиеся элементы из массива  for (int j = i+1; j < arr\_size; j++) {  if (array[i] == array[j]) {  for (int k = j; k < arr\_size; k++) {  array[k] = array[k + 1];    }  j--;  arr\_size--;  }  }  }  cout << endl << "Содержимое file2.txt" << endl;  for (int i = 0; i < arr\_size; i++) {  cout << array[i] << ", ";  }  for (a = 0; a < arr\_size; a++)//записываем значения массива во второй файл  {  fprintf(f2, "%d, ", array[a]);  }  fclose(f);  fclose(f2);//закрываем файлы, с которыми работали  return 0;  } |
| **Результат программы** |
|  |

|  |
| --- |
| **Код программы** |
| #include <stdio.h>  #include<iostream>  #include <locale>  using namespace std;  int main()  {  srand(time(0));  setlocale(LC\_ALL, "");  int a, lenght;  float array[15];  char buffer[250];  char\* estr;  errno\_t errf1, errf2;//Тип errno\_t возвращает код возникающей при работе с файлом ошибки.  FILE\* f, \* f2;//создаём указатели на файлы  errf1 = fopen\_s(&f, "file1.txt", "w");//открываем файл для записи  if (errf1 != 0)//предусматриваем возможность ошибки  {  perror("Невозможно создать файл\n");//Функция perror() выводит эту строку значение глобальной переменной ERRNO с сообщением, указанным в параметре функции.  return EXIT\_FAILURE;  }  printf("Введите текст:\n");  for (a = 0; a < 4; a++)  {  gets\_s(buffer);//передаем введённую строку в буфер  fputs(buffer, f); fprintf(f, "\n");//печатаем буфер в файл  }  printf("Данные записаны в file1.txt\n");  printf("Введите нобходимую длину для переноса:\n");  cin >> lenght;  fclose(f);//закрываем файл  errf2 = fopen\_s(&f2, "file2.txt", "w"); //Открытие для чтения  if (errf2 != NULL)  //Проверка открытия файла  {  printf("Ошибка открытия файла\n"); return EXIT\_FAILURE;  }  errf1 = fopen\_s(&f, "file1.txt", "r"); //Открытие для чтения  if (errf1 != NULL)  //Проверка открытия файла  {  printf("Ошибка открытия файла\n"); return EXIT\_FAILURE;  }  else  printf("\nЗапсианы в file2.txt строки:\n");  for (int i = 0; i < 4; i++) {  fgets(buffer, sizeof(buffer), f); //Чтение строки файла  if(lenght < strlen(buffer)-1) {//если строка проходит критерий длины, то записываем ее во второй файл  fputs(buffer, f2);  puts(buffer);  }  }  fclose(f);  fclose(f2);//закрываем файлы, с которыми работали  } |
| **Результат программы** |
|  |

Дополнительные варианты

| **№ варианта** | **Условия задач** |
| --- | --- |
| **6** | 1. Компоненты файла **fA** – вещественные числа (положительные и отрицательные). Определить и вывести на экран порядковый номер того из них, которое наиболее близко к введенному пользователем целому числу.  2. Создать текстовый файл **F1** не менее, чем из 6 строк, и записать в него информацию. Скопировать в файл **F2** только четные строки из **F1**. |

|  |
| --- |
| **Код программы** |
| #include <stdio.h>  #include<iostream>  #include <locale>  using namespace std;  float randfloat() {//функция для генерации вещественных чисел со значением после запятой  return((float)(rand()) / ((float)(rand()) / 10))-10;  }  int main()  {  srand(time(0));  setlocale(LC\_ALL, "");  int a,number,ind=0;  float array[15];  errno\_t errf1,errf2;//Тип errno\_t возвращает код возникающей при работе с файлом ошибки.  FILE\* f,\*f2;//создаём указатели на файлы  errf1 = fopen\_s(&f, "file1.txt", "w");//открываем файл для записи  if (errf1 != 0)//предусматриваем возможность ошибки  {  perror("Невозможно создать файл\n");//Функция perror() выводит эту строку значение глобальной переменной ERRNO с сообщением, указанным в параметре функции.  return EXIT\_FAILURE;  }  for (a = 0; a < 15; a ++)  {  fprintf(f, "%f, ", randfloat());//генерируем числа в файл  }  printf("Данные записаны в file1.txt\n\n");  fclose(f);//закрываем файл  cout << "Введите целое число:";  cin >> number;  errf1 = fopen\_s(&f, "file1.txt", "r");//открываем тот же файл в режиме чтения  if (errf1 != 0)  {  perror("Невозможно создать файл\n");  return EXIT\_FAILURE;  }  errf2 = fopen\_s(&f2, "file2.txt ", "w");//открываем второй файл в режиме записи  if (errf2 != 0)  {  perror("Невозможно открыть файл\n");  return EXIT\_FAILURE;  }  for (a = 0; a < 15; a++)  {  fscanf\_s(f, "%f, ",&array[a]);//считываем числа из первого файла в массив  }  cout << endl << "Содержимое file1.txt"<<endl;  for (int i = 0; i < 15; i++) {  cout << array[i]<<", ";  if (abs(number - array[i]) < abs(number - array[ind])) { ind = i; }//ищем ближайшее по значению к введённому числу из сгенерированных  }  int arr\_size = 15;  cout << endl << "Содержимое file2.txt" << endl;  cout << array[ind] << ", ";  fprintf(f2, "%f, ", array[ind]);//печатаем нужное число в файл  fclose(f);  fclose(f2);//закрываем файлы, с которыми работали  return 0;  } |
| **Результат программы** |
|  |

|  |
| --- |
| **Код программы** |
| #include <stdio.h>  #include<iostream>  #include <locale>  using namespace std;  int main()  {  srand(time(0));  setlocale(LC\_ALL, "");  int a, lenght;  float array[15];  char buffer[250];  char\* estr;  errno\_t errf1, errf2;//Тип errno\_t возвращает код возникающей при работе с файлом ошибки.  FILE\* f, \* f2;//создаём указатели на файлы  errf1 = fopen\_s(&f, "file1.txt", "w");//открываем файл для записи  if (errf1 != 0)//предусматриваем возможность ошибки  {  perror("Невозможно создать файл\n");//Функция perror() выводит эту строку значение глобальной переменной ERRNO с сообщением, указанным в параметре функции.  return EXIT\_FAILURE;  }  printf("Введите текст:\n");  for (a = 0; a < 6; a++)  {  gets\_s(buffer);//передаем введённую строку в буфер  fputs(buffer, f); fprintf(f, "\n");//печатаем буфер в файл  }  printf("Данные записаны в file1.txt\n");  fclose(f);//закрываем файл  errf2 = fopen\_s(&f2, "file2.txt", "w"); //Открытие для чтения  if (errf2 != NULL)  //Проверка открытия файла  {  printf("Ошибка открытия файла\n"); return EXIT\_FAILURE;  }  errf1 = fopen\_s(&f, "file1.txt", "r"); //Открытие для чтения  if (errf1 != NULL)  //Проверка открытия файла  {  printf("Ошибка открытия файла\n"); return EXIT\_FAILURE;  }  else  printf("\nЗапсианы в file2.txt строки:\n");  for (int i = 0; i < 6; i++) {  fgets(buffer, sizeof(buffer), f); //Чтение строки файла  if(i%2!=0) {//если строка является четной, то записываем ее во второй файл  fputs(buffer, f2);  puts(buffer);  }  }  fclose(f);  fclose(f2);//закрываем файлы, с которыми работали  } |
| **Результат программы** |
|  |

| **№ варианта** | **Условия задач** |
| --- | --- |
| **13** | 1. Компоненты файла **fileA** –  целые числа, значения которых повторяются.  Получить файл **fileB**, образованный из **fileA** числами, которые встречаются в **fileA** более двух раз.  2. Создать текстовый файл **F1** не менее, чем из 5 строк, и записать в него информацию. Скопировать в файл **F2** только те строки из **F1**, которые начинаются с цифры. |

|  |
| --- |
| **Код программы** |
| #include <stdio.h>  #include<iostream>  #include <locale>  using namespace std;  int main()  {  srand(time(0));  setlocale(LC\_ALL, "");  int a,count=0;  int array[15];  errno\_t errf1, errf2;//Тип errno\_t возвращает код возникающей при работе с файлом ошибки.  FILE\* f, \* f2;//создаём указатели на файлы  errf1 = fopen\_s(&f, "file1.txt", "w");//открываем файл для записи  if (errf1 != 0)//предусматриваем возможность ошибки  {  perror("Невозможно создать файл\n");//Функция perror() выводит эту строку значение глобальной переменной ERRNO с сообщением, указанным в параметре функции.  return EXIT\_FAILURE;  }  for (a = 0; a < 15; a++)  {  fprintf(f, "%d, ", rand() % 10);//генерируем числа в файл  }  printf("Данные записаны в file1.txt\n");  fclose(f);//закрываем файл  errf1 = fopen\_s(&f, "file1.txt", "r");//открываем тот же файл в режиме чтения  if (errf1 != 0)  {  perror("Невозможно создать файл\n");  return EXIT\_FAILURE;  }  errf2 = fopen\_s(&f2, "file2.txt ", "w");//открываем второй файл в режиме записи  if (errf2 != 0)  {  perror("Невозможно открыть файл\n");  return EXIT\_FAILURE;  }  for (a = 0; a < 15; a++)  {  fscanf\_s(f, "%d, ", &array[a]);//считываем числа из первого файла в массив  }  cout << "Содержимое file1.txt" << endl;  for (int i = 0; i < 15; i++) {  cout << array[i] << ", ";  }  int arr\_size = 15;  for (int i = 0; i < arr\_size; i++) {//удаляем уникальные элементы из массива  for (int j = 0; j < arr\_size; j++) {  if (array[i] == array[j]) {  count++;  }  }  if(count<2){  for (int k = i; k < arr\_size; k++) {  array[k] = array[k + 1];  }  arr\_size--;  i--;  }  count = 0;  }  cout << endl << "Содержимое file2.txt" << endl;  for (int i = 0; i < arr\_size; i++) {  cout << array[i] << ", ";  }  for (a = 0; a < arr\_size; a++)//записываем значения массива во второй файл  {  fprintf(f2, "%d, ", array[a]);  }  fclose(f);  fclose(f2);//закрываем файлы, с которыми работали  return 0;  } |
| **Результат программы** |
|  |

|  |
| --- |
| **Код программы** |
| #include <stdio.h>  #include<iostream>  #include <locale>  using namespace std;  int main()  {  srand(time(0));  setlocale(LC\_ALL, "");  int a, lenght;  float array[15];  char buffer[250];  char\* estr;  errno\_t errf1, errf2;//Тип errno\_t возвращает код возникающей при работе с файлом ошибки.  FILE\* f, \* f2;//создаём указатели на файлы  errf1 = fopen\_s(&f, "file1.txt", "w");//открываем файл для записи  if (errf1 != 0)//предусматриваем возможность ошибки  {  perror("Невозможно создать файл\n");//Функция perror() выводит эту строку значение глобальной переменной ERRNO с сообщением, указанным в параметре функции.  return EXIT\_FAILURE;  }  printf("Введите текст:\n");  for (a = 0; a < 5; a++)  {  gets\_s(buffer);//передаем введённую строку в буфер  fputs(buffer, f); fprintf(f, "\n");//печатаем буфер в файл  }  printf("Данные записаны в file1.txt\n");  fclose(f);//закрываем файл  errf2 = fopen\_s(&f2, "file2.txt", "w"); //Открытие для чтения  if (errf2 != NULL)  //Проверка открытия файла  {  printf("Ошибка открытия файла\n"); return EXIT\_FAILURE;  }  errf1 = fopen\_s(&f, "file1.txt", "r"); //Открытие для чтения  if (errf1 != NULL)  //Проверка открытия файла  {  printf("Ошибка открытия файла\n"); return EXIT\_FAILURE;  }  else  printf("\nЗапсианы в file2.txt строки:\n");  for (int i = 0; i < 5; i++) {  fgets(buffer, sizeof(buffer), f); //Чтение строки файла  if(buffer[0]<='9' and buffer[0] >= '0') {//если первый символ строки находиться в диапазоне 0-9 ,то записываем ее во второй файл  fputs(buffer, f2);  puts(buffer);  }  }  fclose(f);  fclose(f2);//закрываем файлы, с которыми работали  } |
| **Результат программы** |
|  |

| **№ варианта** | **Условия задач** |
| --- | --- |
| **11** | 1. Компоненты файла **fileA** –  целые числа, значения которых повторяются.  Получить файл **fileB**, образованный из **fileA** числами, которые не повторяются.  2. Создать текстовый файл **F1** не менее, чем из 6 строк, и записать в него информацию. Скопировать в файл **F2** только те строки из **F1**, которые заканчиваются символом «**а**». |

|  |
| --- |
| **Код программы** |
| #include <stdio.h>  #include<iostream>  #include <locale>  using namespace std;  int main()  {  srand(time(0));  setlocale(LC\_ALL, "");  int a;  int array[15];  errno\_t errf1,errf2;//Тип errno\_t возвращает код возникающей при работе с файлом ошибки.  FILE\* f,\*f2;//создаём указатели на файлы  errf1 = fopen\_s(&f, "file1.txt", "w");//открываем файл для записи  if (errf1 != 0)//предусматриваем возможность ошибки  {  perror("Невозможно создать файл\n");//Функция perror() выводит эту строку значение глобальной переменной ERRNO с сообщением, указанным в параметре функции.  return EXIT\_FAILURE;  }  for (a = 0; a < 15; a ++)  {  fprintf(f, "%d, ", rand() % 10);//генерируем числа в файл  }  printf("Данные записаны в file1.txt\n");  fclose(f);//закрываем файл  errf1 = fopen\_s(&f, "file1.txt", "r");//открываем тот же файл в режиме чтения  if (errf1 != 0)  {  perror("Невозможно создать файл\n");  return EXIT\_FAILURE;  }  errf2 = fopen\_s(&f2, "file2.txt ", "w");//открываем второй файл в режиме записи  if (errf2 != 0)  {  perror("Невозможно открыть файл\n");  return EXIT\_FAILURE;  }  for (a = 0; a < 15; a++)  {  fscanf\_s(f, "%d, ",&array[a]);//считываем числа из первого файла в массив  }  cout << "Содержимое file1.txt"<<endl;  for (int i = 0; i < 15; i++) {  cout << array[i]<<", ";  }  int arr\_size = 15;  for (int i = 0; i < arr\_size; i++) {//удаляем повторяющиеся элементы из массива  for (int j = i+1; j < arr\_size; j++) {  if (array[i] == array[j]) {  for (int k = j; k < arr\_size; k++) {  array[k] = array[k + 1];    }  j--;  arr\_size--;  }  }  }  cout << endl << "Содержимое file2.txt" << endl;  for (int i = 0; i < arr\_size; i++) {  cout << array[i] << ", ";  }  for (a = 0; a < arr\_size; a++)//записываем значения массива во второй файл  {  fprintf(f2, "%d, ", array[a]);  }  fclose(f);  fclose(f2);//закрываем файлы, с которыми работали  return 0;  } |
| **Результат программы** |
|  |

|  |
| --- |
| **Код программы** |
| #include <stdio.h>  #include<iostream>  #include <locale>  using namespace std;  int main()  {  srand(time(0));  setlocale(LC\_ALL, "");  int a, lenght;  float array[15];  char buffer[250];  char\* estr;  errno\_t errf1, errf2;//Тип errno\_t возвращает код возникающей при работе с файлом ошибки.  FILE\* f, \* f2;//создаём указатели на файлы  errf1 = fopen\_s(&f, "file1.txt", "w");//открываем файл для записи  if (errf1 != 0)//предусматриваем возможность ошибки  {  perror("Невозможно создать файл\n");//Функция perror() выводит эту строку значение глобальной переменной ERRNO с сообщением, указанным в параметре функции.  return EXIT\_FAILURE;  }  printf("Введите текст:\n");  for (a = 0; a < 6; a++)  {  gets\_s(buffer);//передаем введённую строку в буфер  fputs(buffer, f); fprintf(f, "\n");//печатаем буфер в файл  }  printf("Данные записаны в file1.txt\n");  fclose(f);//закрываем файл  errf2 = fopen\_s(&f2, "file2.txt", "w"); //Открытие для чтения  if (errf2 != NULL)  //Проверка открытия файла  {  printf("Ошибка открытия файла\n"); return EXIT\_FAILURE;  }  errf1 = fopen\_s(&f, "file1.txt", "r"); //Открытие для чтения  if (errf1 != NULL)  //Проверка открытия файла  {  printf("Ошибка открытия файла\n"); return EXIT\_FAILURE;  }  else  printf("\nЗапсианы в file2.txt строки:\n");  for (int i = 0; i < 6; i++) {  fgets(buffer, sizeof(buffer), f); //Чтение строки файла  if (buffer[strlen(buffer)-2]=='a') {//если строка кончается символом "a", то записываем ее во второй файл  fputs(buffer, f2);  puts(buffer);  }  }  fclose(f);  fclose(f2);//закрываем файлы, с которыми работали  } |
| **Результат программы** |
|  |