Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Лабораторная работа 2

По дисциплине «Основы алгоритмизации и программирования»

На тему «Работа с файлами на языке С»

Выполнила:

Студентка 1 курса 2 группы

Глухова Д.В.

Преподаватель: асс. Андронова М.В.

2023, Минск

Вариант 2.

5. В соответствии со своим вариантом разработать программы для условий, приведенных в таблице ниже, и изучить способы работы с файлами на языке С.

1. Даны два файла вещественных чисел с именами **fA** и **fB**, содержащие элементы прямоугольных матриц **M1** и **M2** по строкам, причем начальный элемент каждого файла содержит количество столбцов соответствующей матрицы. Создать файл той же структуры с именем **fC**, содержащий сумму **M1+M2**.

|  |
| --- |
| Программный код |
| // 2 вариант 1 задание  #include <stdio.h>  #include <iostream>  int main()  {  setlocale(LC\_ALL, "");  using namespace std;  srand((unsigned)time(NULL));  const int N = 100;  int M1[N][N], M2[N][N], M3[N][N], columns, rows;  errno\_t errA, errB, errC; // errno\_t возвращает в соответствующей переменной код ошибки, которая возникает при работе с файлом  FILE\* fA, \*fB, \*fC; // указатель fA на файл  errA = fopen\_s(&fA, "fA.txt", "w"); // файл открывается для записи, т к "w"  errB = fopen\_s(&fB, "fB.txt", "w");  errC = fopen\_s(&fC, "fС.txt", "w");  if ((errA != 0) || (errB != 0) || (errC != 0)) // блок выполняется в случае возникабщей при работе с файлом ошибки  {  perror("Невозможно создать файл\n"); // интерпретирует errno в строку и выводит вместе с сообщением в скобках  return EXIT\_FAILURE; // макрос для возвращения кода при неудачном выполнении и завершении  }    cout << "количество строк в первой матрице = ";  cin >> rows;  cout << "количество столбцов в первой матрице = ";  cin >> columns;    // заполняем и выводим в консоль первую матрицу  for (int i = 0; i < rows; i++)  {  cout << "\n";  for (int j = 0; j < columns; j++)  {  M1[i][j] = rand() % 10;  cout << M1[i][j] << "\t";  }  }    cout << "\n" << "\n";  // заполняем и выводим в консоль вторую матрицу  for (int i = 0; i < rows; i++)  {  cout << "\n";  for (int j = 0; j < columns; j++)  {  M2[i][j] = rand() % 10;  cout << M2[i][j] << "\t";  }  }  cout << "\n" << "\n";  // третья матрица и вывод в консоль  for (int i = 0; i < rows; i++)  {  cout << "\n";  for (int j = 0; j < columns; j++)  {  M3[i][j] = M1[i][j] + M2[i][j];  cout << M3[i][j] << "\t";  }  }  fprintf(fA, "%d", columns, "\n"); // начальный элемент файла = количеству столбцов  fprintf(fB, "%d", columns, "\n");  fprintf(fC, "%d", columns, "\n");  // запись матриц  for (int i = 0; i < rows; i++)  {  fprintf(fA, "\n");  fprintf(fB, "\n");  fprintf(fC, "\n");  for (int j = 0; j < columns; j++)  {  fprintf(fA, "%d\t ", M1[i][j]);  fprintf(fB, "%d\t ", M2[i][j]);  fprintf(fC, "%d\t ", M3[i][j]);  }  }  // файлы закрываются  fclose(fA);  fclose(fB);  fclose(fC);  return 0;  } |
| Консоль отладки |
|  |
| Исходные файлы fA.txt и fB.txt |
|  |
| Конечный файл fC.txt |
|  |

2. Компоненты файла**f** – целые двухзначные (отличные от нуля) числа, причем 10 положительных чисел, 10 отрицательных, и т. д. Получить файл **g***,* в котором записаны сначала 5 положительных чисел, затем 5 отрицательных и т. д.

|  |
| --- |
| Программный код |
| // 2 вариант 2 задание  #include <stdio.h>  #include <iostream>  using namespace std;  int main()  {  setlocale(LC\_ALL, "rus");  const int len = 3;  char line1[len], line2[len];  FILE\* f1;  fopen\_s(&f1, "f.txt", "r");  FILE\* f2;  fopen\_s(&f2, "g.txt", "w");  for (int j = 0; j < 5; j++)  {  for (int i = 0; i < 2; i++)  {  fgets(line1, len, f1);  fputs(line1, f2);  }  }  for (int j = 0; j < 5; j++)  {  for (int i = 0; i < 2; i++)  {  fgets(line1, len, f1);  }  }  for (int j = 0; j < 5; j++)  {  for (int i = 0; i < 2; i++)  {  fgets(line1, len, f1);  fputs(line1, f2);  }  }  for (int j = 0; j < 5; j++)  {  for (int i = 0; i < 2; i++)  {  fgets(line1, len, f1);  }  }  for (int j = 0; j < 5; j++)  {  for (int i = 0; i < 2; i++)  {  fgets(line1, len, f1);  fputs(line1, f2);  }  }  for (int j = 0; j < 5; j++)  {  for (int i = 0; i < 2; i++)  {  fgets(line1, len, f1);  }  }  for (int j = 0; j < 5; j++)  {  for (int i = 0; i < 2; i++)  {  fgets(line1, len, f1);  fputs(line1, f2);  }  }  cout << "Данные записаны в файл g.txt" << endl;  } |
| Консоль отладки |
|  |
| Исходный файл f.txt |
|  |
| Конечный файл g.txt |
|  |

Дополнительные задания.

Вариант 3.

1. Дан файл вещественных чисел, содержащий элементы квадратной матрицы (по строкам), причем начальный элемент файла содержит значение количества столбцов матрицы. Создать новый файл той же структуры, содержащий транспонированную матрицу.

|  |
| --- |
| Программный код |
| // 3 вариант 1 задание  #include <stdio.h>  #include <iostream>  int main()  {  setlocale(LC\_ALL, "");  int a; errno\_t err, columns = 4;  FILE\* f;  err = fopen\_s(&f, "matrix.txt", "w"); // открываем файл matrix.txt для записи  if (err != 0)  {  perror("Невозможно создать файл\n"); // в случае ошибки открытия  return EXIT\_FAILURE;  }  int Matrix[16] = { 2,7,12,-4,  6,-45,0,13,  96,43,-2,8,  33,-5,63,-11 };  // вывод первой матрицы  fprintf(f, "%d столбца\n", columns); // выводим количество столбцов  for (int j = 0; j < 16; j += 4)  {  for (int i = j; i < j + 4; i++)  {  fprintf(f, "%d\t", Matrix[i]); // записываем в файл matrix.txt первоначальную матрицу  }  fprintf(f, "\n");  }  // вывод второй матрицы  err = fopen\_s(&f, "matrixT.txt", "w"); // открываем файл matrixT.txt  if (err != 0)  {  perror("Невозможно создать файл\n"); // в случае ошибки открытия  return EXIT\_FAILURE;  }  fprintf(f, "%d столбца.\n", columns); // выводим количество столбцов  for (int j = 0; j < 4; j++) // меняем местами столбцы и строки, чтобы транспонировать  {  for (int i = j; i <= j + 12; i += 4)  {  fprintf(f, "%d\t", Matrix[i]); // записываем в файл matrixT.txt транспонированную матрицу  }  fprintf(f, "\n");  }  fprintf(f, "\n");  printf("Данные записаны в файл matrixT.txt\n");  fclose(f); // закрываем файл  return 0;  } |
| Консоль отладки |
|  |
| Исходный файл matrix.txt |
|  |
| Конечный файл matrixT.txt |
|  |

1. 2. Даны три файла целых чисел одинакового размера с именами **NameA**, **NameB** и **NameC**. Создать новый файл с именем **NameD**, в котором чередовались бы элементы исходных файлов с одним и тем же номером: **A0, B0, C0, A1, B1, C1, A2, B2, C2, ...**

|  |
| --- |
| Программный код |
| // 3 вариант 2 задание  #include <stdio.h>  #include <iostream>  using namespace std;  int main()  {  setlocale(LC\_ALL, "");  const int len = 4;  char line1[len], line2[len], line3[len];  FILE\* f1;  fopen\_s(&f1, "NameA.txt", "r"); // открываем файл для чтения  FILE\* f2;  fopen\_s(&f2, "NameB.txt", "r");  FILE\* f3;  fopen\_s(&f3, "NameC.txt", "r");  FILE\* f4;  fopen\_s(&f4, "NameD.txt", "w"); // открываем файл для записи  for (int i = 0; i <= 6; i++)  {  fgets(line1, len, f1); // читаем строку из указанного потока, пока не встретится символ перевода на новую строку  fputs(line1, f4); // записываем строку в указанный поток(f4 - NameD.txt)  fgets(line2, len, f2);  fputs(line2, f4); // записываем строку  fgets(line3, len, f3);  fputs(line3, f4); // записываем строку  }  cout << "Данные записаны в файл NameD.txt" << endl;  fclose(f1);  fclose(f2);  fclose(f3);  fclose(f4);  } |
| Консоль отладки |
|  |
| Исходные файлы |
|  |
| Конечный файл |
|  |

Вариант 14

1. Дан файл вещественных чисел, содержащий элементы квадратной матрицы по строкам, причем начальный элемент файла содержит значение количества столбцов матрицы. Создать новый файл той же структуры, содержащий **k**-ый столбец исходной матрицы.

|  |
| --- |
| Программный код |
| // 14 вариант 1 задание  #include <iostream>  #include <stdio.h>  using namespace std;  #define N 100  int main()  {  setlocale(LC\_ALL, "rus");  FILE\* file1, \* file2;  int arr[N][N], size;  errno\_t err1, err2, err3;  srand((unsigned)time(NULL));  err1 = fopen\_s(&file1, "file1.txt", "w"); // открываем файл  if (err1 != 0)  {  perror("ошибка открытия file1.txt"); // при ошибке  return EXIT\_FAILURE;  }  cout << "количество строк/столбцов в матрице: " << endl;  cin >> size;  fprintf(file1, "%d \n", size);  for (int i = 0; i < size; i++)  {  for (int k = 0; k < size; k++)  {  arr[i][k] = rand() % 10; // создание массива, его вывод в консоль и запись в файл  cout << arr[i][k] << '\t';  fprintf(file1, "%d \t", arr[i][k]);  }  cout << '\n';  fprintf(file1, "\n");  }  fclose(file1);//закроем файл для открытия в другом режиме  err3 = fopen\_s(&file1, "file1.txt", "r"); // открытие файла для считывания данных  if (err3 != 0)  {  perror("ошибка открытия file1.txt");  return EXIT\_FAILURE;  }  err2 = fopen\_s(&file2, "file2.txt", "w");  if (err2 != 0)  {  perror("ошибка открытия file2.txt");  return EXIT\_FAILURE;  }  int x;  fscanf\_s(file1, "%d", &x); //узнаем размер массива, считывая файл  int\*\* arr1 = new int\* [x];  // инициализация динамического массива  for (int i = 0; i < x; i++)  {  arr1[i] = new int[x];  }  for (int i = 0; i < x; i++)  {  for (int j = 0; j < x; j++)  {  fscanf\_s(file1, "%d", &(arr1[i][j])); // читаем матрицу  }  }  cout << "столбец для вывода" << endl;  int k;  cin >> k;  cout << endl << "столбец: " << endl;  for (int i = 0; i < x; i++)  {  fprintf\_s(file2, "%d\n", arr1[i][k]); // записываем и выводим выбранный столбец  printf("%d\n", arr1[i][k]);  }  fclose(file1); //закрытие всех файлов  fclose(file2);  delete[] arr1; // освобождение памяти  return 0;  } |
| Консоль отладки |
|  |
| Исходные файлы |
|  |
| Конечный файл |
|  |

1. Создать текстовый файл **F1** не менее, чем из 5 строк, и записать в него информацию. Скопировать в файл **F2** только строки из **F1**, которые начинаются с цифры.

|  |
| --- |
| Программный код |
| // 14 вариант 2 задание  #include <stdio.h>  #include <iostream>  int main()  {  using namespace std;  setlocale(LC\_ALL, "Russian");  FILE\* F1;  FILE\* F2;  char str[255];  errno\_t err;  err = fopen\_s(&F1, "f1\_14.txt", "w");  for (int i = 0; i < 6; i++)  {  printf("Строка %d: \n", i + 1);  gets\_s(str);  fputs(str, F1);  fputs("\n", F1);  }  fclose(F1);  err = fopen\_s(&F1, "f1\_14.txt", "r");  err = fopen\_s(&F2, "f2\_14.txt", "w");  while (fgets(str, 255, F1) != 0)  {  if (str[0] >= '0' && str[0] <= '9')  {  fputs(str, F2);  }  }  fclose(F1);  fclose(F2);  } |
| Консоль отладки |
|  |
| Исходные файлы |
|  |
| Конечный файл |
|  |

15 вариант

1. Дан файл вещественных чисел, содержащий элементы квадратной матрицы по строкам, причем начальный элемент файла содержит значение количества столбцов матрицы. Создать новый файл той же структуры, содержащий **k**-ый столбец исходной матрицы.

|  |
| --- |
| Программный код |
| // 15 вариант 1 задание  #include <iostream>  #include <stdio.h>  using namespace std;  #define N 100  int main()  {  setlocale(LC\_ALL, "rus");  FILE\* file1, \* file2;  int arr[N][N], size;  errno\_t err1, err2, err3;  srand((unsigned)time(NULL));  err1 = fopen\_s(&file1, "file1.txt", "w"); // открываем файл  if (err1 != 0)  {  perror("ошибка открытия file1.txt"); // при ошибке  return EXIT\_FAILURE;  }  cout << "количество строк/столбцов в матрице: " << endl;  cin >> size;  fprintf(file1, "%d \n", size);  for (int i = 0; i < size; i++)  {  for (int k = 0; k < size; k++)  {  arr[i][k] = rand() % 10; // создание массива, его вывод в консоль и запись в файл  cout << arr[i][k] << '\t';  fprintf(file1, "%d \t", arr[i][k]);  }  cout << '\n';  fprintf(file1, "\n");  }  fclose(file1);//закроем файл для открытия в другом режиме  err3 = fopen\_s(&file1, "file1.txt", "r"); // открытие файла для считывания данных  if (err3 != 0)  {  perror("ошибка открытия file1.txt");  return EXIT\_FAILURE;  }  err2 = fopen\_s(&file2, "file2.txt", "w");  if (err2 != 0)  {  perror("ошибка открытия file2.txt");  return EXIT\_FAILURE;  }  int x;  fscanf\_s(file1, "%d", &x); //узнаем размер массива, считывая файл  int\*\* arr1 = new int\* [x];  // инициализация динамического массива  for (int i = 0; i < x; i++)  {  arr1[i] = new int[x];  }  for (int i = 0; i < x; i++)  {  for (int j = 0; j < x; j++)  {  fscanf\_s(file1, "%d", &(arr1[i][j])); // читаем матрицу  }  }  cout << "столбец для вывода" << endl;  int k;  cin >> k;  cout << endl << "столбец: " << endl;  for (int i = 0; i < x; i++)  {  fprintf\_s(file2, "%d\n", arr1[i][k]); // записываем и выводим выбранный столбец  printf("%d\n", arr1[i][k]);  }  fclose(file1); //закрытие всех файлов  fclose(file2);  delete[] arr1; // освобождение памяти  return 0;  } |
| Консоль отладки |
|  |
| Исходные файлы |
|  |
| Конечный файл |
|  |

2. Создать текстовый файл **F1** не менее, чем из 8 строк, и записать в него информацию. Скопировать из файла **F1** в файл **F2** строки, начиная с **k** до **k + 3**.

|  |
| --- |
| Программный код |
| #include <stdio.h>  #include <iostream>  #define LEN 100  using namespace std;  int main()  {  setlocale(LC\_ALL, "ru");  int row1, row2;  FILE\* f1;  FILE\* f2;  errno\_t err1, err2;  err1 = fopen\_s(&f1, "t1.txt", "r");  if (err1 != NULL)  {  printf("ошибка открытия файла\n"); return -1;  }  err2 = fopen\_s(&f2, "t2.txt", "w");  if (err2 != NULL)  {  printf("Ошибка\n"); return -1;  }  fclose(f1);  fclose(f2);  cout << "k строки:\n";  cin >> row1;  row2 = row1 + 3;  fopen\_s(&f1, "t1.txt", "r");  fopen\_s(&f2, "t2.txt", "w");  int sum = 0;  char str[LEN];  while (fgets(str, LEN, f1))  {  row1--;  row2--;  if (sum < row1)  {  }  else  {  if (row1 <= row2)  {  fputs(str, f2);  }  row1++;  }  }  } |
| Консоль отладки |
|  |
| Исходные файлы |
|  |
| Конечный файл |
|  |