МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Забайкальский государственный университет» (ФГБОУ ВО «ЗабГУ»)

Факультет энергетический

Кафедра информатики, вычислительной техники и прикладной математики

## ОТЧЕТ

по учебной практике (эксплуатационная)

обучающегося Борисова Екатерина Олеговна

(фамилия, имя, отчество) Курс 1 Группа ИВТ-20

Направление подготовки 09.03.01 Информатика и вычислительная техника

(шифр, наименование)

Направленность ОП «Вычислительные машины, комплексы, системы и сети»

(шифр, наименование)

Руководитель практики от вуза доцент кафедры Семигузов Дмитрий Александрович

(Ученая степень, должность, Ф.И.О.)

г. Чита 2021

## Структура отчёта о прохождении практики

|  |
| --- |
| СОДЕРЖАНИЕ  ВЫПОЛНЕННЫЕ ЗАДАНИЯ  (Условия и исходные коды решенных индивидуальных заданий)  Задание 11а:    Код:  #include <stdio.h>  #include <math.h>//подключаем библиотеку математики  int main()  {      int x, y, z;//целые x, y и z      float a, b;//действительные a и b        printf("Enter x: ");      scanf("%d", &x);    //ввод х      printf("Enter y: ");      scanf("%d", &y);    //ввод у      printf("Enter z: ");      scanf("%d", &z);    //ввод z        a=(sqrt(fabs(x-1))-pow(fabs(y), 1/3))/(1+(pow(x, 2)/2)+(pow(y, 2)/4));//подсчет а      b=x\*(atan(z)+exp(-(x+3)));//подсчет b          printf("a: %.3f\n", a);//вывод а      printf("b: %.3f\n", b);//вывод b      return 0;//конец программы  }  Задание 58а:  Дано действительное число a. Для функции f(x), графики которых представланы на рис.1 а - г, вычислить f(a).    #include <stdio.h>  int main()  {  float a, f;//действительное а и функция f  printf("Enter a: ");  scanf("%f", &a);//ввод а    if (a<=0)   //если а<0      f=fabs(a);    // то f=|a|  else if (a>=0)//если a>0      f=-pow(a,2);//то f=a^2  printf("f=%.0f\n", f);//вывод f      return 0;//конец программы  }  Задание 72б  Дано действительное число *а.*Вычислить *f*(*a*), где *f*—периодическая функция с периодом 2, совпадающая на отрезке [—1, 1]:  б) с функцией, график которой изображен на рис. 5    Код:  #include <iostream>  using namespace std;  float calc(float x)//функция подсчета f(a)  {  while (x < -1)  {  x += 2;  }  while (x > 1)  {  x -= 2;  }  if (x <= 0)  {  return x + 1;  }  else  {  return sqrt(1 - x \* x);  }  }  int main()  {  float a;  cout << "enter a: ";  cin >> a;  float f = calc(a);//считаем функцию f(a)  cout <<"F(a) = " << f;    }  Задание 88г  Дано натуральное число *n.*  г) Приписать по единице в начало и в конец записи числа *n*.  Код:  #include <cstdlib>  #include <iostream>  using namespace std;  void calc(int n);//процедура приписывает по единице в начало и в конец записи числа n  void chek(int n);//процедура проверяет правильность введеных данных  int main()  {  int n;  cin >> n;  chek(n);  }  void calc(int n)//процедура приписывает по единице в начало и в конец записи числа n  {  int l = 1, rezult;//l - приписываемая 1, rezult - результат  rezult = n \* 10 + l;//в результат записываем n и единицу в конце  while (n > 0)  {  l \*= 10;//увеличиваем l в 10  n /= 10;//уменьшаем n в 10  }  rezult += l \* 10;//к результату приписываем единицу в начало  cout << rezult;  }  void chek(int n)  {  if (n > 0)//проверяем является ли n натуральным числом  calc(n);//если да, вызывается процедура calc  else  cout << "wrong number, it should be > 0";//если нет, сообщается о ошибке  }  Задание 204  В некоторых видах спортивных состязаний выступление каждого спортсмена независимо оценивается несколькими судьями, затем из всей совокупности оценок удаляются наибдолее высокая и наиболее низкая, а для оставшихся оценок вычисляется среднее арифметическое, которое и идет в зачет спортсмену. Если наиболее высокую оценку выставило несколько судей, то из совокупности оценок удаляется только одна такая оценка; аналогично поступают с наиболее низкими оценками. Даны натуральное число n, действительные положительные числа a1,...,an (n ≥ 3). Считая, что числа a1,...,an - это оценки, выставленные судьями одному из участников соревнований, определить оценку, которая пойдет в зачет этому спортсмену.  Код:  #include <iostream>  #include <time.h>  using namespace std;  void filling\_array(int\* array, int n);  void array\_out(int\* array, int n);  int max\_el(int\* array, int n);  int min\_el(int\* array, int n);  float grade\_calc(int\* array, int n);  int main()  {  srand(time(NULL));  const int n = 7;  int grade[n];  filling\_array(grade, n);  array\_out(grade, n);  cout << "final grade: " << grade\_calc(grade, n);  }  void filling\_array(int\* array, int n)//заполнение матрицы  {  for (int i = 0; i < n; i++)  {  array[i] = rand() % 10 + 1 ;  }  }  void array\_out(int\* array, int n)//вывод матрицы  {  cout << "given grades: ";  for (int i = 0; i < n; i++)  {  cout << array[i] << " ";  }  cout << "\n";  }  int max\_el(int\* array, int n)//нахождение максимума  {  int max = 0;  for (int i = 0; i < n; i++)  {  if (array[i] > max)  max = array[i];  }  return max;  }  int min\_el(int\* array, int n)//нахождение минимума  {  int min = 0;  for (int i = 0; i < n; i++)  {  if (array[i] < min)  min = array[i];  }  return min;  }  float grade\_calc(int\* array, int n)//подсчет финальной оценки  {  int all\_grades;  float final\_grade;  int max = max\_el(array, n);  int min = min\_el(array, n);  all\_grades = 0;  for (int i = 0; i < n; i++)  {  all\_grades += array[i];  }  final\_grade = (all\_grades - (min + max)) / (n - 2);  return final\_grade;  }  Задание 339е  Даны целые числаa a1,..., an (в этой последовательности могут быть повторяющиеся члены).  е) Выяснить, имеется ли в последовательности хотя бы одна пара совпадающих чисел.  Код:  #include <iostream>  #include <time.h>  using namespace std;  void filling\_array(int\* array, int n);  void out\_array(int\* array, int n);  void out\_array(int\* array, int n);  void calc(int\* array, int n);  int main()  {  srand(time(NULL));  int n;  cout << "enter array's length: ";  cin >> n;  cout << "\n";  int\* array = new int[n];  filling\_array(array, n);  out\_array(array, n);  calc(array, n);  delete[] array;  }  void filling\_array(int\* array, int n)//заполнение массива  {  for (int i = 0; i < n; i++)  {  array[i] = rand() % 10;  }  }  void out\_array(int\* array, int n)//вывод массива  {  cout << "array: ";  for (int i = 0; i < n; i++)  {  cout << array[i] << " ";  }  cout << "\n";  cout << "\n";  }  void calc(int\* array, int n)//поиск пар  {  int found = 0;  for (int i = 1; i < n; i++)  {  if (array[i] == array[i - 1])  found++;  }  if (found >= 1)  cout << "the sequence contains at least one pair of coinciding numbers";  else  cout << "there are no matching numbers in the sequence";  cout << "\n";  }  Задание 269а  Даны натуральное число *n*, символы *s1*,...,sn. Группы символов, разделенные пробелами (одним или несколькими) и не содержащие пробелов внутри себя будем называть *словами*  а) Подсчитать количество слов в данной последовательности.  Код:  #include <iostream>  #include <cstdlib>  #include <string>  using namespace std;  int main()  {  string str;  getline(cin, str);  cout << str << endl;  int space = 0, word = 0;  int flag = 0; //первые элементы пробелы  for (int i = 0; i < str.size() - 1; i++)  {  if ((flag == 0) && (str[i] == ' '))//если в начале стоят пробелы  {  continue;  }  if (str[i] != ' ')  {  flag++;  space = 0;  continue;  }  else  {  space++;  if (space == 1)  {  word++;  }  }  }  if (str[str.size() - 1] != ' ')//проверяем последний элемент  {  word++;  }    cout <<"\n"<< word;  return 0;  }  Задание 374а  Дано натуральное число n. Выяснить, сколько положительных элементов содержит матрица [aij] i, j = 1, ..., n, если    Код:  #include <iostream>  #include <cmath>  using namespace std;  int main()  {  cout << "matrix:\n";  cout << "\n";  const int n = 5;  float arr[n][n];//задаем размер матрицы  for (int i = 0; i < n; i++)//проходим по строкам  {  for (int j = 0; j < 5; j++)//проходим по столбикам  {  arr[i][j] = sin(i+1 + (j+1) / 2);//заполняем матрицу  cout << arr[i][j];//вывод матрицы  cout << " ";  }  cout << "\n";  }  int k = 0;//счетчик положительных элементов  for (int i = 0; i < n; i++)//проходим по строкам  {  for (int j = 0; j < 5; j++)//проходим по столбикам  {  if (arr[i][j] > 0)//если элемент положительный  {  k++;//увеличиваем счетчик  }    }  }  cout << "\n";  cout << "number of positive elements: ";//вывод счетчика  cout << k;  cout << "\n";  }  Задание 699  Даны квадратные матрицы А и В порядка n. Получить матрицу АВ-ВА  Код:  #include <iostream>  #include <time.h>  using namespace std;  int\*\* create\_memory\_matrix(int n)//выделение памяти под матрицу  {  int\*\* matrix = new int\* [n];//выделяем память под массив указателей  for (int i = 0; i < n; i++)  {  matrix[i] = new int[n];  }  return matrix;  }  void filling\_matrix(int\*\* matrix, int n)//заполнение матрицы  {  //srand(time(NULL));  for (int i = 0; i < n; i++)  {  for (int j = 0; j < n; j++)  {  matrix[i][j] = rand() % 10-1;  }  }  }  void matrix\_out (int\*\* matrix, int n)//вывод матрицы  {  for (int i = 0; i < n; i++)  {  for (int j = 0; j < n; j++)  {  cout << matrix[i][j] << " ";  }  cout << "\n";  }  }  void calc\_matrix(int\*\* matrix\_mul, int\*\* matrix\_a, int\*\* matrix\_b, int n)//подсчет АВ-ВА  {  for (int i = 0; i < n; i++)  {  for (int j = 0; j < n; j++)  {  matrix\_mul[i][j] = 0;  for (int k = 0; k < n; k++)  {  matrix\_mul[i][j] += matrix\_a[i][k] \* matrix\_b[k][j] - matrix\_b[i][k] \* matrix\_a[k][j];  }  }  }  }  int main()  {  srand(time(NULL));  int n;  cout << "Enter size of matrix: ";  cin >> n;  int\*\* matrixA = create\_memory\_matrix(n);//выделение памяти под матрицу А  filling\_matrix(matrixA, n);//заполнение матрицы А  cout << "Matrix A:\n";  cout << "\n";  matrix\_out(matrixA, n);//Вывод матрицы А  cout << "\n";  int\*\* matrixB = create\_memory\_matrix(n);//выделение памяти под матрицу В  filling\_matrix(matrixB, n);//заполнение матрицы В  cout << "Matrix B:\n";  cout << "\n";  matrix\_out(matrixB, n);//Вывод матрицы В  cout << "\n";  int\*\* matrix\_mul = create\_memory\_matrix(n);//выделение памяти под матрицу АВ-ВА  calc\_matrix(matrix\_mul, matrixA, matrixB, n);//Подсчет АВ-ВА  cout << "Matrix AB-BA:\n";  cout << "\n";  matrix\_out(matrix\_mul, n);//Вывод матрицы АВ-ВА  cout << "\n";  }  Задание 458  Даны неотрицательные целые числа n, m;вычислить А(n, m), где    (это-так называемая функция Аккермана). Использовать программу, включающую рекурсивную процедуру.  Код:  #include <iostream>  using namespace std;  int input();//ввод данных  void output(long a);//вывд данных  long akker(int n, int m);//подсчет функции аккермана  int main()  {  int n, m;  long a;  cout << "enter n: ";  n = input();  cout << "enter m: ";  m = input();  cout << "result: " << akker(n, m);  }  int input()  {  int n;  cin >> n;  return n;  }  void output(long a)  {  cout << a;  }  long akker(int n, int m)  {  long a = 0;  if (n == 0)//если выполняется первое условие  {  a = m + 1;//находится а  }  if (n != 0 && m == 0)//если выполняется второе условие  {  a = akker(n - 1, 1);//вызывается рекурсия  }  if (n > 0 && m > 0)//если выполняется третье условие  {  a = akker(n - 1, akker(n, m - 1));//вызывается рекурсия  }  return a;  }  Задание 692а  Дана действительная квадратная матрица порядка n. Найти наибольшее из значений элементов, расположенных в заштрихованной части матрицы (рис. 39).    Код:  #include <iostream>  using namespace std;  int\*\* create\_memory\_matrix(int n);  void filling\_matrix(int\*\* arr, int n);  void matrix\_out(int\*\* arr, int n);  void find\_max(int\*\* arr, int n);  int main()  {  int n;  cout << "enter size: ";  cin >> n;  int\*\* MatrixA = create\_memory\_matrix(n);  filling\_matrix(MatrixA, n);  find\_max(MatrixA, n);  }  int\*\* create\_memory\_matrix(int n)//выделение памяти под матрицу  {  int\*\* matrix = new int\* [n];//выделяем память под массив указателей  for (int i = 0; i < n; i++)  {  matrix[i] = new int[n];  }  return matrix;  }  void filling\_matrix(int\*\* arr, int n)//заполнение матрицы  {  srand(time(NULL));  for (int i = 0; i < n; i++)  {  for (int j = 0; j < n; j++)  {  arr[i][j] = rand() % 20 - 1;  printf\_s("%5d", arr[i][j]);  }  cout << "\n";  }  // return matrix;  }  void matrix\_out(int\*\* arr, int n)  {  for (int i = 0; i < n; i++)  {  for (int j = 0; j < n; j++)  {  printf\_s("%5d", arr[i][j]);  }  cout << "\n";  }  }  void find\_max(int\*\* arr, int n)  {  int num\_i, num\_j, max = 0;  for (int i = 0; i < n; i++)  for (int j = 0; j < n; j++)  {  if ((j >= i) && (arr[i][j] > max))  {  max = arr[i][j];  num\_i = i;  num\_j = j;  }  }  cout << "\nmax element = " << max << " coordinates: i = " << num\_i << " j = " << num\_j << endl;  }  Задание 482  Дан символьный файл f. Получить файл g, образованный из файла f заменой всех его прописных(больших) букв одноименными строчными (малыми).  Код:  #include <iostream>  #include <string>  #include <fstream>  using namespace std;  int main()  {  string path = "f.txt";//название файла f  ofstream file\_f;//создаем фаил для записи  file\_f.open(path);//открываем файл для записи, связываем с именем  if (!file\_f.is\_open())//если не удалось открыть  {  cout << "file open error1" << endl;//сообщаем об этом  }  else//если открылся  {  cout << "file opened successfully" << endl;  string something;//создаем строку  cout << "enter something: ";  getline(cin, something);  file\_f << something;//записываем строку в файл  }  file\_f.close();//закрываем файл  ifstream file\_fe;//создаем файл для чтения  file\_fe.open(path);//открываем файл для чтения, связываем с именем  string str;//создаем строку для изменения  if (!file\_fe.is\_open())//если не удалось открыть  {  cout << "file open error2" << endl;//сообщаем об этом  }  else//если открылся  {  while (!file\_fe.eof())//пока не закончился файл  {  getline(file\_fe, str);//записываем данные из файла в строку  for (int i = 0; i < str.length(); i++)  {  if (isupper(str[i]))//если буква заглавная  {  str[i] = (char)tolower(str[i]);//меняем на прописную  }  }  }  }  file\_fe.close();//закрываем файл  string path2 = "g.txt";//название файла g  ofstream file\_g;//создаем файл для записи  file\_g.open(path2);//открываем файл для записи, связываем с именем  if (!file\_g.is\_open())  {  cout << "file open error3" << endl;  }  else  {  file\_g << str;  cout << str << endl;  }  file\_g.close();  }  Задание 497  Дан символьный файл f. Группы символов, разделённые пробелами (одним или нескольким ) и не содержащие пробелов внутри себя, будем, как и прежде (см. задачу [269](https://ivtipm.github.io/Programming/Glava08/index08.htm#z269)), называть словами. Удалить из файла все однобуквенные слова и лишние пробелы. Результат записать в файл g.  Код:  #include <iostream>  #include <fstream>  #include <string>  using namespace std;  int main()  {  ifstream file\_in;//чтение  ofstream file\_out;//запись  string f = "f.txt";//чтение  string g = "g.txt";//запись  file\_in.open(f);  file\_out.open(g);  string str;  while (!file\_in.eof())  {  file\_in >> str;  if (str.size() != 1)  {  file\_out << str << " ";  }  str.clear();  }  file\_in.close();  file\_out.close();  cout << "done" << endl;  }  Задание 844а  Построить графики функций:  а) y = 3x2;  #include <windows.h>  #include <iostream>  #include <cmath>  int main()  {    HWND hwnd = GetConsoleWindow();  HDC hdc = GetDC(hwnd);    int x = 0;    for (float i = -20; i < 20; i += 0.05)  {  SetPixel(hdc, x, 250 - 10 \* (3\*i\*i), RGB(255, 255, 255));  x += 1;  }  ReleaseDC(hwnd, hdc);  std::cin.ignore();  return 0;  }  Задание 136а  Даны натуральное число n, действительные числа a1,..., an. Вычислить:  а) a1 + ... + an;  Код:  #include <iostream>  #include <time.h>  using namespace std;  typedef struct MyStruct  {  int fe;  struct MyStruct\* adress;  }my\_list;  void new\_list(my\_list\* head);  void last\_add(my\_list\* head, my\_list\* novoi);  void list\_show(my\_list\* head);//вывод содержимого списка на экран  int summa(my\_list\* head);//сумма  void del(my\_list\* head);//удаление списка  int main()  {  srand(time(NULL));  my\_list\* head = new my\_list;  head->fe = rand() % 20;  head->adress = nullptr;  int n;  cout << "enter n: ";  cin >> n;  for (int i = 1; i < n; i++)  {  new\_list(head);  }  list\_show(head);  cout << "\nsumm = " << summa(head) << endl;  del(head);  }  void new\_list(my\_list\* head)//создает новый узел списка  {  my\_list\* novoi = new my\_list;  novoi->fe = rand() % 20;  novoi->adress = nullptr;  last\_add(head, novoi);  }  void last\_add(my\_list\* head, my\_list\* novoi)//добавляет к последнему  {  my\_list\* tek = head;  while (tek->adress != nullptr)  {  tek = tek->adress;  }  tek->adress = novoi;  }  void list\_show(my\_list\* head)//вывод содержимого списка на экран  {  my\_list\* tek = head;  while (tek != nullptr)  {  cout << tek->fe << "\t";  tek = tek->adress;  }    }  int summa(my\_list\* head)//сумма  {  int sum = 0;  my\_list\* tek = head;  while (tek != nullptr)  {  sum += tek->fe;  tek = tek->adress;  }  return sum;  }  void del(my\_list\* head)//удаление списка  {  my\_list\* tek = head;  while (head != nullptr)  {  tek = head;  head = head->adress;  tek->adress = nullptr;  free(tek);  }  }  Задание 551в  Даны натуральное число n, символы s1, ..., sn. Будем рассматривать слова, образованные входящими в последовательность s1, ..., snсимволами (см.задачу [269](https://ivtipm.github.io/Programming/Glava08/index08.htm#z269)). Ниже описываются преобразования, каждое из которых следует произвести при выполнении указаного условия. Затем последовательность вне зависимости от того, подвергалась она или нет преобразованию, должна быть отредактирована следующим образом. Должны быть удалены группы пробелов, которыми начинается и заканчивается последовательность, а каждая внутренняя группа пробелов должна быть заменена одним пробелом. Преобразования:  в) если первое и последнее слова совпадают и общее число слов больше единицы, то удалить первое и последнее слова, а оставшиеся символы переставить в обратном порядке.  Код:  #include <iostream>  #include <time.h>  #include <string>  #include <sstream>  using namespace std;  typedef struct MyStruct  {  string fe;  struct MyStruct\* adress;  }my\_list;  void new\_list(my\_list\* head, string str);//создает новый узел списка  void last\_add(my\_list\* head, my\_list\* novoi);//добавляет к последнему  void list\_show(my\_list\* head);//вывод содержимого списка на экран  void del(my\_list\* head);//удаление списка  bool chek(my\_list\* head);//проверяет ровняется ли первое слово последнему  my\_list\* del\_1el(my\_list\* head);//удаление первого элемента  void del\_last\_el(my\_list\* head);//удаление первого элемента  void rev\_str(my\_list\* head);//разворачивает все строки в списке  int main()  {  stringstream bufer;//буфер для нарезки строки  string str, slovo;//str для введения строки, slovo для отрезаного слова  cout << "enter string: ";  getline(cin, str);//ввод строки    bufer << str;//режется строка  my\_list\* head = new my\_list;//создается память под голову  bufer >> slovo;//заносим отрезаное слово в slovo  head->fe = slovo;//заносим slovo в слово  head->adress = nullptr;//адресу головы присваевается ноль  while (bufer >> slovo) //пока слово режется  {  new\_list(head, slovo);//создается список  }  cout << "\ngot: ";  list\_show(head);//выводится список  cout << "\n";  if (chek(head))//проверка на одинаковость первого и последнего эл  { //если они одинаковые  head = del\_1el(head);//удаляется первый эл  del\_last\_el(head);//удаляется последний эл  rev\_str(head);//разворачиваем строку  cout << "\nafter: " << endl;  list\_show(head);//выводим развернутый список  cout << "\n";  }  else//если они разные  {  cout << "they're different" << endl;  }  }  void new\_list(my\_list\* head, string str)//создает новый узел списка  {  my\_list\* novoi = new my\_list;//создаем память под новый элемент  novoi->fe = str;//зпносим строку  novoi->adress = nullptr;//адресу присваеваем ноль  last\_add(head, novoi);//добавляем к концу  }  void last\_add(my\_list\* head, my\_list\* novoi)//добавляет к последнему  {  my\_list\* tek = head;//tek - текущий, записываем в него голову(первый)  while (tek->adress != nullptr)//пока в строке адресса не будет ноль  {  tek = tek->adress;//переходит на следующий  }  tek->adress = novoi;//в конец ставится новый  }  void list\_show(my\_list\* head)//вывод содержимого списка на экран  {  my\_list\* tek = head;//tek - текущий, записываем в него голову(первый)  while (tek != nullptr)//пока не дойдет до последнего элемента  {  cout << tek->fe << "\t";//выводится содержимое на экране  tek = tek->adress;//переход на следующий элемент  }  }  bool chek(my\_list\* head)//проверяет ровняется ли первое слово последнему  {  my\_list\* tek = head;//tek - текущий, записываем в него голову(первый)  while (tek->adress != nullptr)//пока в строке адресса не будет ноль  {  tek = tek->adress;//ищем последний  }  if (head->fe == tek->fe)//если они одинаковые  {  return true;//возращает true  }  else//если разные  {  return false;//возвращает false  }  }  void rev\_str(my\_list\* head)//разворачивает все строки в списке  {  my\_list\* tek = head;//tek - текущий, записываем в него голову(первый)  while (tek != nullptr)//пока не дойдет до последнего элемента  {  reverse(tek->fe.begin(), tek->fe.end());//разворачивает строку в другую сторону  tek = tek->adress;//переходит на следующий элемент  }  }  my\_list\* del\_1el (my\_list\* head)//удаление первого элемента  {  my\_list\* tek = head;//tek - текущий, записываем в него голову(первый)  tek = head;//в тек записывают голову  head = head->adress;//в адресс головы записывается следующий элемент  tek->adress = nullptr;//адрессу тека присваевается ноль  free(tek);//освобождает память из под тека  return head;//возвращает новую голову  }  void del\_last\_el(my\_list\* head)//удаление первого элемента  {  my\_list\* tek = head;//tek - текущий, записываем в него голову(первый)  while (tek->adress != nullptr)//ищем последний эл  {  tek = tek->adress;  }  while (head->adress != tek)//ищем предпоследний эл  {  head = head->adress;  }  head->adress = nullptr;//предпоследнему эл присваеваем ноль  free(tek);//освободить память из под последнего элемента  }  void del(my\_list\* head)//удаление списка  {  my\_list\* tek = head;//tek - текущий, записываем в него голову(первый)  while (head != nullptr)//пока не закончится  {  tek = head;//присваеваем теку голову(чтобы не потерять ее)  head = head->adress;//адресс головы переходит на следущий эл  tek->adress = nullptr;//адресу тека присвается ноль  free(tek);//освобождается память из под тека  }  }  Задание 72а (c#)  Дано действительное число *а.*Вычислить *f*(*a*), где *f*—периодическая функция с периодом 2, совпадающая на отрезке [—1, 1]:  а) с функцией — *x2 +*1;  Код:  using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace WindowsFormsApp2  {  public partial class Form1 : Form  {  public Form1()  {  InitializeComponent();  }  private void calc\_Click(object sender, EventArgs e)  {  string str = enter\_a.Text;  float a = Convert.ToSingle(str);  while (a < -1)  {  a += 2;  }  while (a > 1)  {  a -= 2;  }  listBox1.Items.Add(-(a \* a) + 1);  }  }  }  Задание 339д (c#)  Даны целые числаa a1,..., an (в этой последовательности могут быть повторяющиеся члены).  д) Выяснить, сколько чисел входит в последовательность более чем по одному разу.  Код:  using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace ex339d  {  public partial class Form1 : Form  {  public Form1()  {  InitializeComponent();  }  private void button1\_Click(object sender, EventArgs e)  {  /\*Даны целые числаa a1,..., an (в этой последовательности могут быть повторяющиеся члены).  д) Выяснить, сколько чисел входит в последовательность более чем по одному разу.\*/  string str = enter\_n.Text;  int n = Convert.ToInt32(str);  int[] ar = new int[n];  int found = 0;  Random rand = new Random();  for (int i = 0; i < n; i++)  {  ar[i] = rand.Next(1, 11);  }  for (int i = 0; i < n; i++)  {  for (int j = 1; j < n-1; j++)  {  if (ar[i] == ar[j] && (i != j))  {  found++;  }  }  }  //found--;  listBox1.Items.Add("Последовательность: ");  for (int i = 0; i < ar.Length; i++)  {  listBox1.Items.Add(ar[i]);  }  listBox1.Items.Add("Повторений:");  listBox1.Items.Add(found);  }  }  }  Задание 269б (c#)  Даны натуральное число *n*, символы *s1*,...,sn. Группы символов, разделенные пробелами (одним или несколькими) и не содержащие пробелов внутри себя будем называть *словами*  б) Подсчитать количество букв *а* в последнем слове данной последовательности.  Код:  using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace ex269b  {  public partial class Form1 : Form  {  public Form1()  {  InitializeComponent();  }  private void button1\_Click(object sender, EventArgs e)  {  string str = enter\_str.Text;  int num\_a = 0, i = str.Length-1;  while (str[i] != ' ')  if ((str[i--] == 'a') || (str[i--] == 'A'))  num\_a++;    listBox1.Items.Add("Строка:");  listBox1.Items.Add(str);  listBox1.Items.Add("Количество а:");  listBox1.Items.Add(num\_a);  }  private void label3\_Click(object sender, EventArgs e)  {  }  }  }  Задание 701а (c#)  Даны квадратная матрица А порядка n и вектор b c n элементами. Получить вектор:  а) Ab; |
| Код:  using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace ex701a  {  public partial class Form1 : Form  {  public Form1()  {  InitializeComponent();  }  /\*Даны квадратная матрица А порядка n и  вектор b c n элементами. Получить вектор:  а) Ab;\*/  private void button1\_Click(object sender, EventArgs e)  {  //размер массива и матрицы  string str = enter\_n.Text;  int n = Convert.ToInt32(str);  //объявление массива, матрицы и массива под умножение  int[] array = new int[n];  int[,] matrix = new int[n, n];  int[] multi = new int[n];  //рандом  Random rand = new Random();  //заполнение массива  for (int i = 0; i < n; i++)  {  array[i] = rand.Next(1, 11);  }  //заполнение матрицы  for (int i = 0; i < n; i++)  {  for (int j = 0; j < n; j++)  {  matrix[i, j] = rand.Next(1, 11);  }  }  //умножение  int summ = 0;  for(int i = 0; i < n; i++)  {  summ = 0;  for(int j = 0; j < n; j++)  {  summ = summ + matrix[i, j] \* array[j];  }  multi[i] = summ;  }  //вывод  //вывод массива  dataGridView\_array.RowCount = 1;  dataGridView\_array.ColumnCount = n;  for (int i = 0; i<n; i++)  {  dataGridView\_array.Rows[0].Cells[i].Value = Convert.ToString(array[i]);  }  //вывод матрицы  dataGridView\_matrix.RowCount = n;  dataGridView\_matrix.ColumnCount = n;  for (int i = 0; i < n; i++)  {  for (int j = 0; j < n; j++)  {  dataGridView\_matrix.Rows[i].Cells[j].Value = Convert.ToString(matrix[i, j]);  }  }  //вывод вектора массива  dataGridView\_multi.RowCount = 1;  dataGridView\_multi.ColumnCount = n;  for (int i = 0; i < n; i++)  {  dataGridView\_multi.Rows[0].Cells[i].Value = Convert.ToString(multi[i]);  }  }  }  }  СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ |