Министерство образования и науки Российской Федерации

**Волжский политехнический институт (филиал) федерального государственного бюджетного образовательного учреждения высшего образования «Волгоградский государственный технический университет»**
**(ВПИ (филиал) ВолгГТУ)**

Факультет«*Вечерний факультет*»

Кафедра«*Информатика и технология программирования*»

# КОНТРОЛЬНАЯ РАБОТА

по дисциплине«*Программирование и основы алгоритмизации*»

на тему**РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА С ИСПОЛЬЗОВАНИЕМ АЛГОРИТМОВ МОДУЛЬНОГО ПРОГРАММИРОВАНИЯ**

*(вариант №9)*

Студент*Елизавета Анатольевна Перминова*
(имя, отчество, фамилия)

Группа*ВХАЗ-250а*

Оценка _____
(в баллах)

Проверил _____ *доц. О. Ф. Абрамова*
(подпись и дата подписания) (долж., инициалы и фамилия)

Волжский, 2020г.

**Постановка задачи:**

1. В одномерном массиве, состоящем из n вещественных элементов, вычислить:
    1) сумму элементов массива, расположенных между первым и вторым отрицательными элементами;
    2) преобразовать массив таким образом, чтобы в первой его половине располагались элементы, модуль которых не превышает 1, потом все остальные.

2. Дана целочисленная матрица размерностью NxM. Определить:

1) номер первой из строк, содержащих хотя бы один положительный элемент;

2) максимальный элемент матрицы и поменять местами строку, в которой он находится и столбец, в котором он находится.

3. С помощью генератора случайных чисел занести в массив C(20) числа из диапазона от -15 до 20. Написать функцию, определяющую номера минимальных элементов в первой (с 1 по 10) и во второй половине (с 11 по 20) массива и их значения. Ввод элементов массива и вызов созданной функции осуществлять в основной программе.

4. Написать функцию, подсчитывающую количество нулей в каждом столбце матрицы X(5,6). Вывести номера столбцов, в которых нет нулей. Ввод элементов матрицы (с помощью генератора случайных чисел) и вызов созданной функции осуществлять в основной программе.

5. Из заданной строки удалить все цифры, подсчитать количество и сумму удаленных цифр, заменить все заглавные латинские буквы на маленькие.

**Дополнительные требования:**

Все задачи варианта объединяются одним общим интерфейсом. Выполнение каждой задачи осуществляется как последовательный вызов всех необходимых подпрограмм, согласно разработанному алгоритму. Предусмотреть возможность повтора выполнения каждой задачи. Результаты

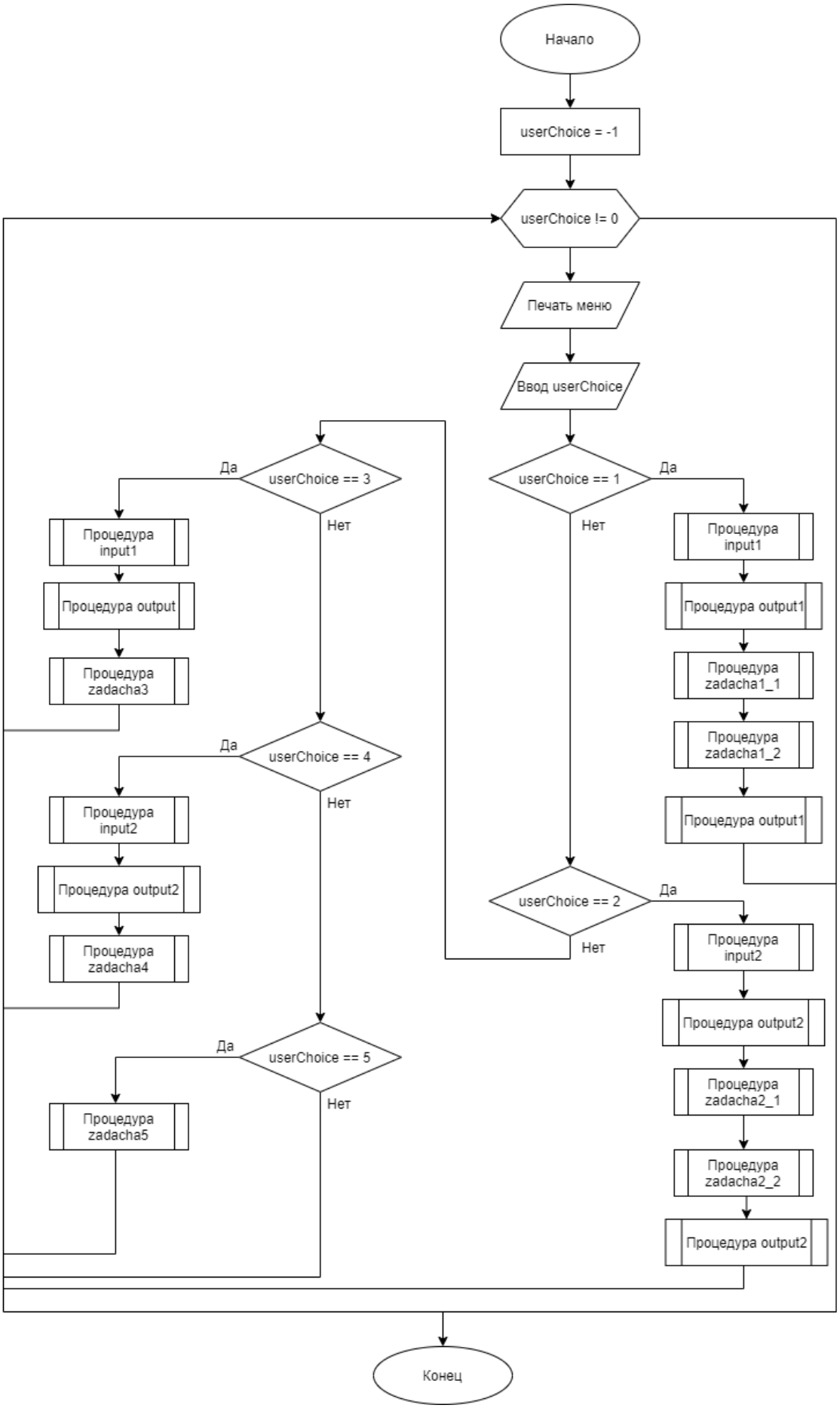
должны печататься с максимально возможными комментариями. Размерность массивов должна задаваться пользователем программного средства.

## 1. Описание глобальных переменных

Было разработано программное средство согласно заявленным в методических указаниях требованиям. Для успешной работы данного программного средства в соответствии с предъявляемыми требованиями разработчиком было решено глобальные переменные не использовать. Это повышает мобильность программы и и соответствует основным принципам структурного программирования.

Выполнение каждой подзадачи производится в отдельной функции, вызов которых производится из главного меню, расположенного в головной функции main(). Организация работы меню выполнена с использованием одной переменной – userChoice. В эту переменную считывается значение, вводимое пользователем программного средства и соответствующее тому номеру пункта меню, которое пользователь выбирает для выполнения.

Блок схема головной функции main().



## 2. Описание основных подпрограмм

В данном программном средстве разработчиком было составлено 5 основных функций задача1…задача5 (userChoice == 1/2/3/4/5) соответственно, реализующие одну из поставленных в условии задач. При этом каждая из этих функций содержит в себе обращение еще к ряду подпрограмм, отвечающих за реализацию отдельных типовых алгоритмов.

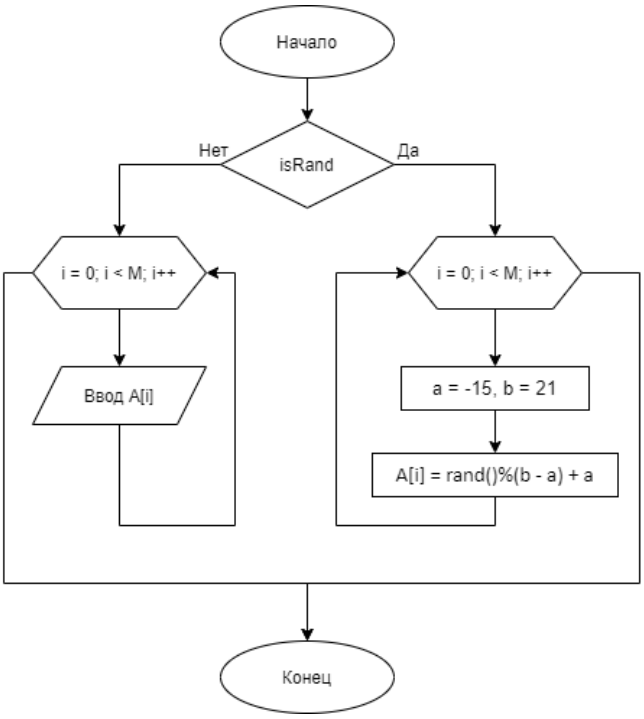
Спецификации используемых в программном средстве подпрограмм (полное описание подпрограмм, включая блок-схемы) и всего средства в целом:

1) void clear_screen(); - функция, вызывающая системную очистку экрана
2) void input1(int M, double A[maxLen], bool isRand); - функция для заполнения 1 мерного массива
    - int M – размерность массива
    - double A[maxLen] – одномерный массив статического размера maxLen
    - bool isRand – флаг для обозначения как будут вводиться данные (true – случайно, false - вручную)
3) void output1(int M, double A[maxLen]); - функция для вывода на экран 1 мерного массива
    - int M – размерность массива
    - double A[maxLen] – одномерный массив статического размера maxLen
4) void input2(int N, int M, int A[maxLen][maxLen], bool isRand); - функция для заполнения 2 мерного массива вручную
    - int M, N – размерность массива
    - double A[maxLen] [maxLen] – двумерный массив статического размера maxLen
    - bool isRand – флаг для обозначения как будут вводиться данные (true – случайно, false - вручную)
5) void output2(int N, int M, int A[maxLen][maxLen]); - функция для вывода на экран 2 мерного массива
    - int M, N – размерность массива
    - int A[maxLen] [maxLen] – двумерный массив статического размера maxLen

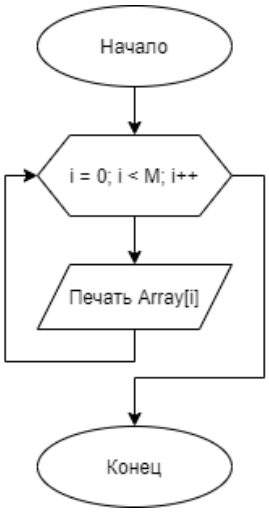
6) double zadacha1_1(int N, double A[maxLen]); функция, считающая сумму элементов между 1 отрицательным и 2 отрицательным элементом
    - int N – размерность массива
    - double A[maxLen] – одномерный массив статического размера maxLen
7) void zadacha1_2(int N, double A[maxLen]);- функция для преобразования массива таким образом, чтобы в первой его половине располагались элементы, модуль которых не превышает 1, потом все остальные.
    - int N – размерность массива
    - double A[maxLen] – одномерный массив статического размера maxLen
8) int zadacha2_1(int N, int A[maxLen][maxLen]); - функция возвращающая номер первой из строк, содержащих хотя бы один положительный элемент. Если таких строк нет вернет –1.
    - int N – размерность массива NxN
    - int A[maxLen] [maxLen] – двумерный массив статического размера maxLen
9) void zadacha2_2(int N, int A[maxLen][maxLen]); - функция, ищущая максимальный элемент матрицы и, меняющая местами строку, в которой он находится и столбец, в котором он находится.
    - int N – размерность массива NxN
    - int A[maxLen] [maxLen] – двумерный массив статического размера maxLen
10) void zadacha3(double A[maxLen]); - функция, определяющая и выводящая номера минимальных элементов в первой (с 1 по 10) и во второй половине (с 11 по 20) массива и их значения
    - double A[maxLen] – одномерный массив статического размера maxLen
11) void zadacha4(int N, int M, int A[maxLen][maxLen]); - функция, подсчитывающая количество нулей в каждом столбце матрицы и выводящая номера столбцов, в которых нет нулей.
    - int M, N – размерность массива
    - int A[maxLen] [maxLen] – двумерный массив статического размера maxLen
12) void zadacha5(); - функция считывающая текст и удаляющая из него все цифры, подсчитывающая количество и сумму удаленных цифр, заменяющая все заглавные латинские буквы на маленькие .

Блок схемы подпрограмм:

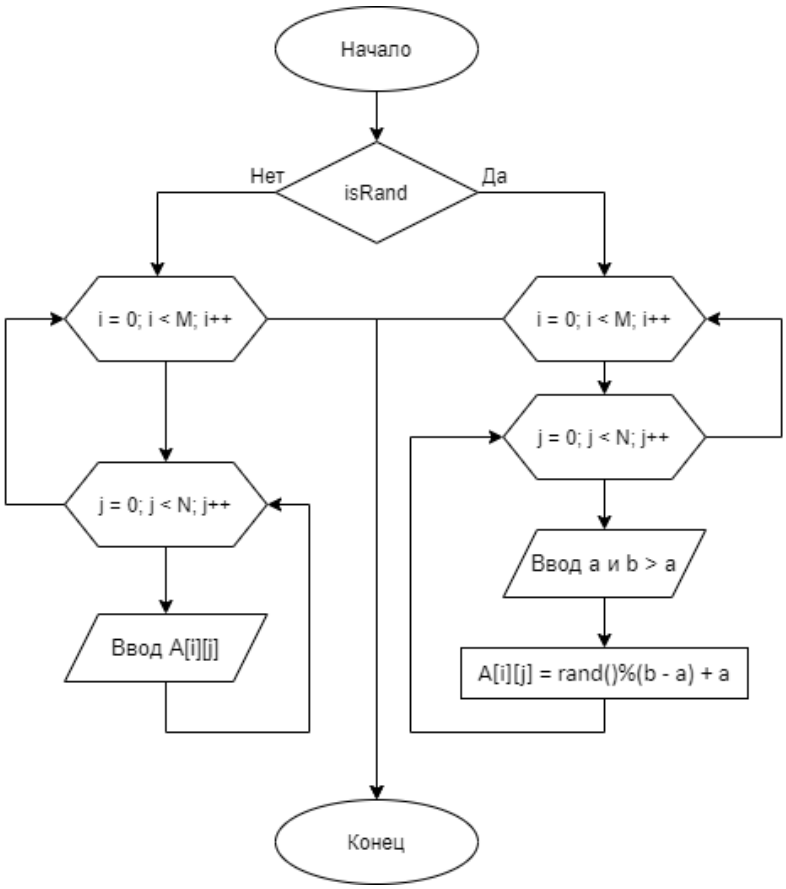
1) void input1(int M, double A[maxLen], bool isRand); - функция для заполнения 1 мерного массива



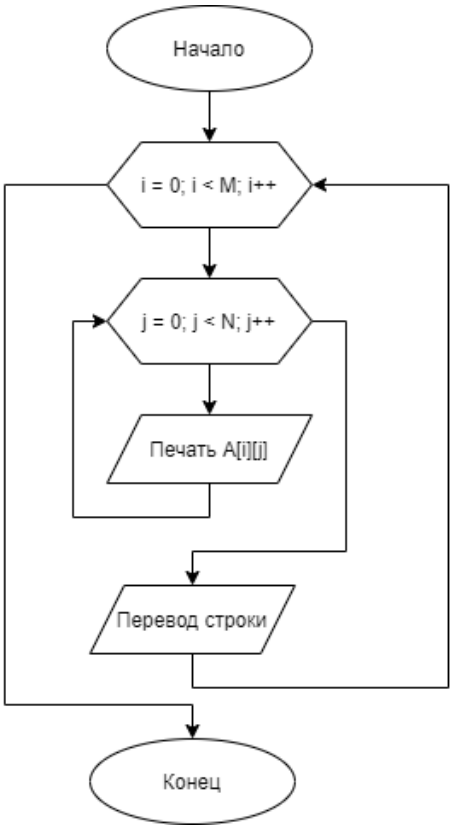

2) void output1(int M, double A[maxLen]); - функция для вывода на экран 1 мерного массива



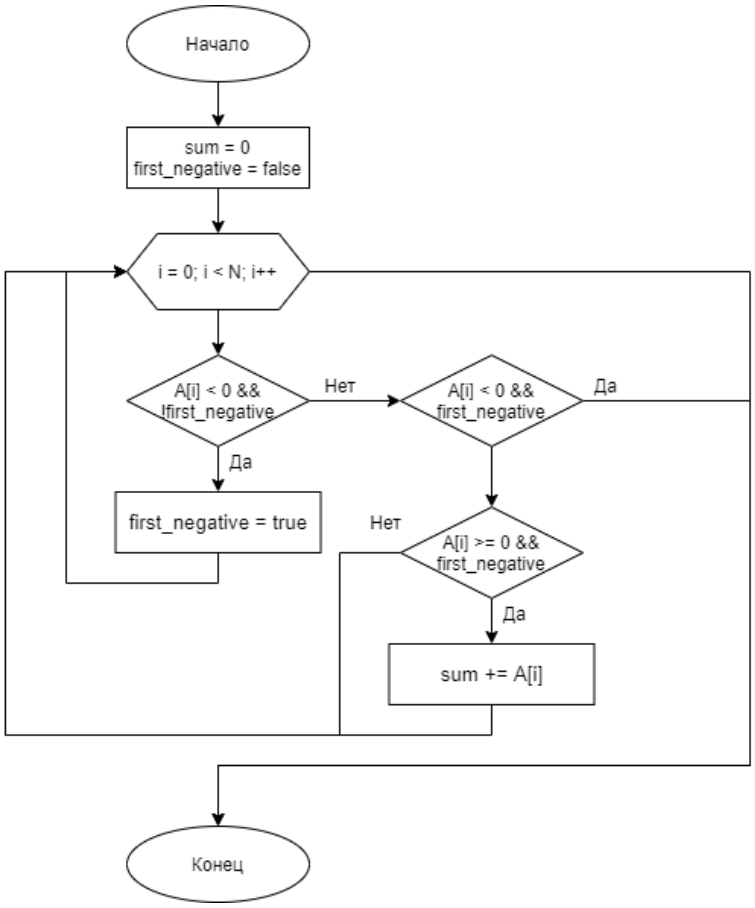

3) void input2(int N, int M, int A[maxLen][maxLen], bool isRand); - функция для заполнения 2 мерного массива вручную



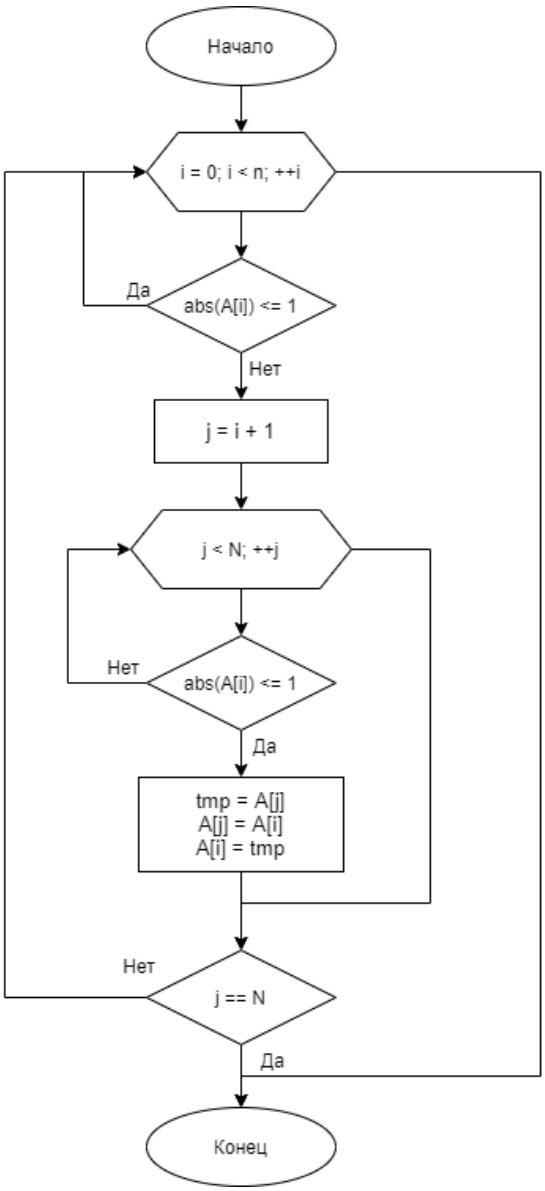

4) void output2(int N, int M, int A[maxLen][maxLen]); - функция для вывода на экран 2 мерного массива



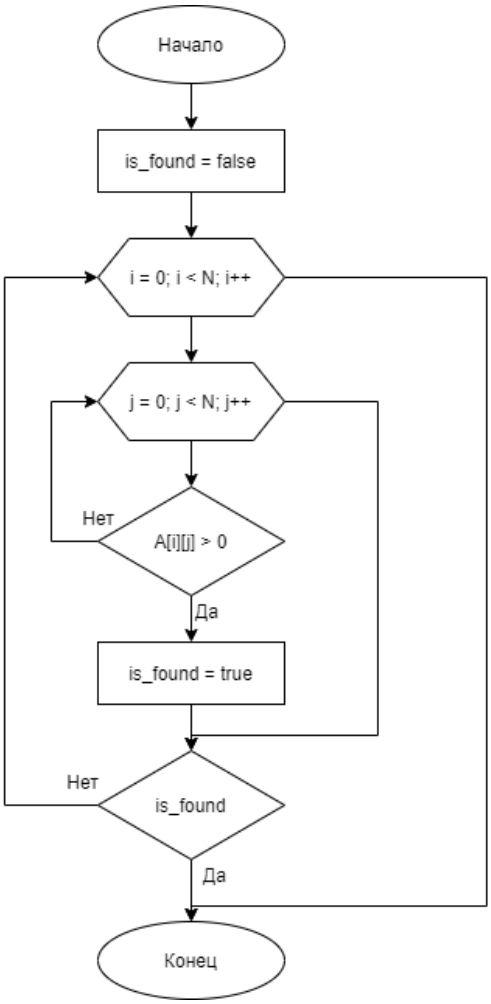
5) double zadacha1_1(int N, double A[maxLen]); функция, считающая сумму элементов между 1 отрицательным и 2 отрицательным элементом



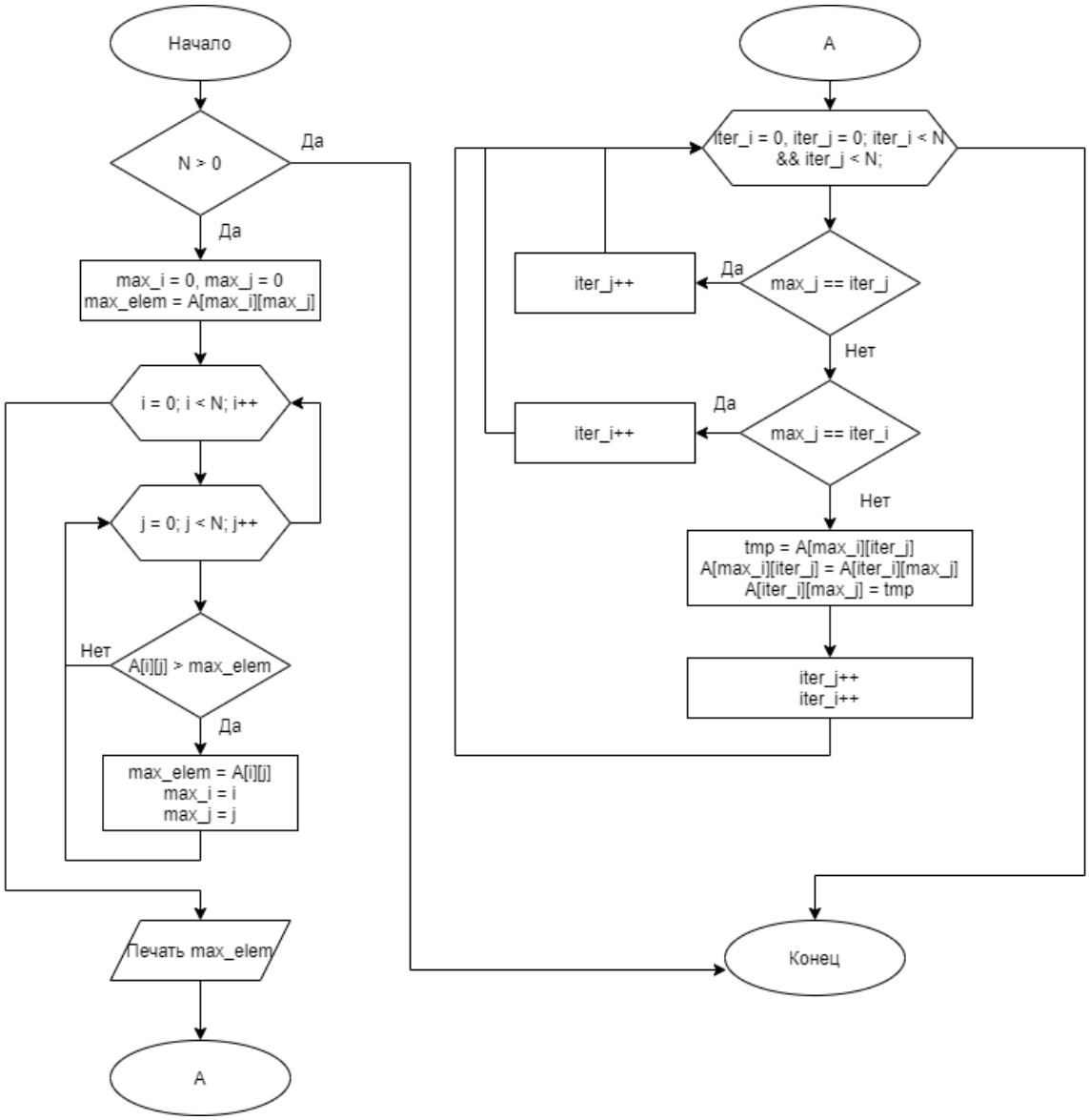

6) void zadacha1_2(int N, double A[maxLen]);- функция для преобразования массива таким образом, чтобы в первой его половине располагались элементы, модуль которых не превышает 1, потом все остальные.


Начало
i = 0; i < n; ++i
Да
abs(A[i]) <= 1
Нет
j = i + 1
j < N; ++j
Нет
abs(A[i]) <= 1
Да
tmp = A[j]
A[j] = A[i]
A[i] = tmp
Нет
j == N
Да
Конец

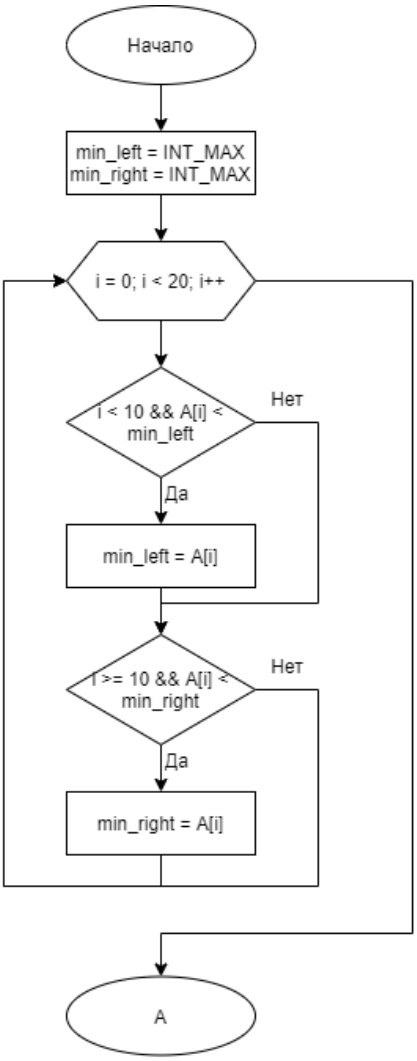
7) int zadacha2_1(int N, int A[maxLen][maxLen]); - функция, возвращающая номер первой из строк, содержащих хотя бы один положительный элемент. Если таких строк нет вернет –1.

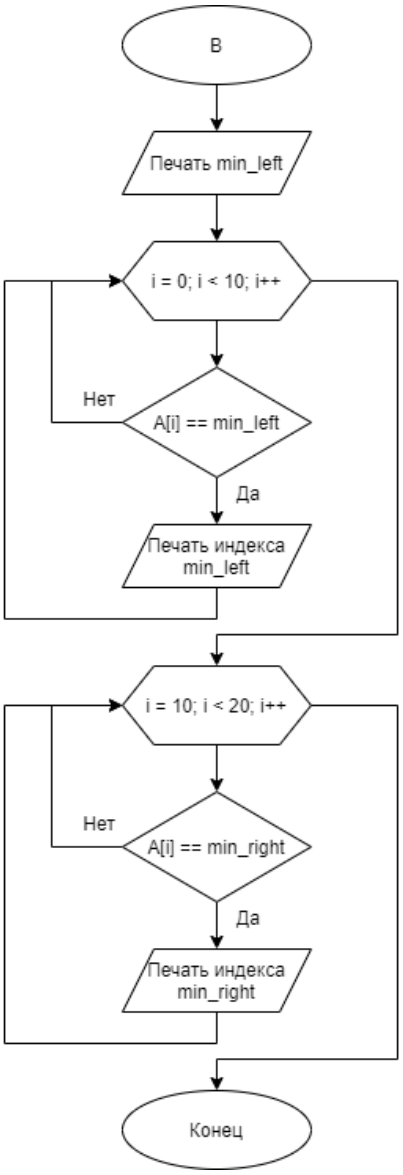


8) void zadacha2_2(int N, int A[maxLen][maxLen]); - функция, ищущая максимальный элемент матрицы и, меняющая местами строку, в которой он находится и столбец, в котором он находится.

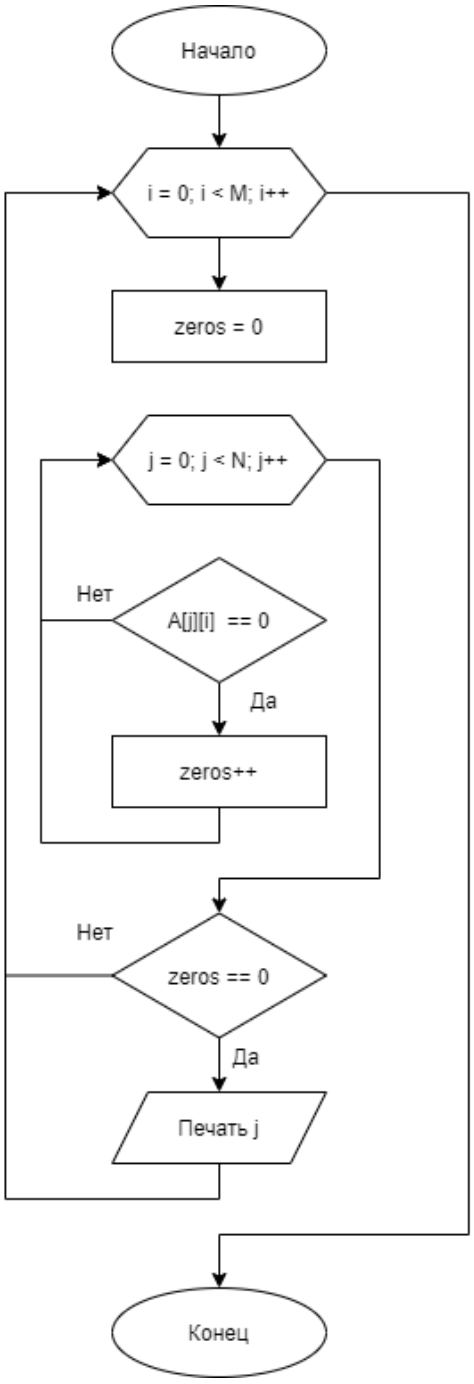


9) void zadacha3(double A[maxLen]); - функция, определяющая и выводящая номера минимальных элементов в первой (с 1 по 10) и во второй половине (с 11 по 20) массива и их значения

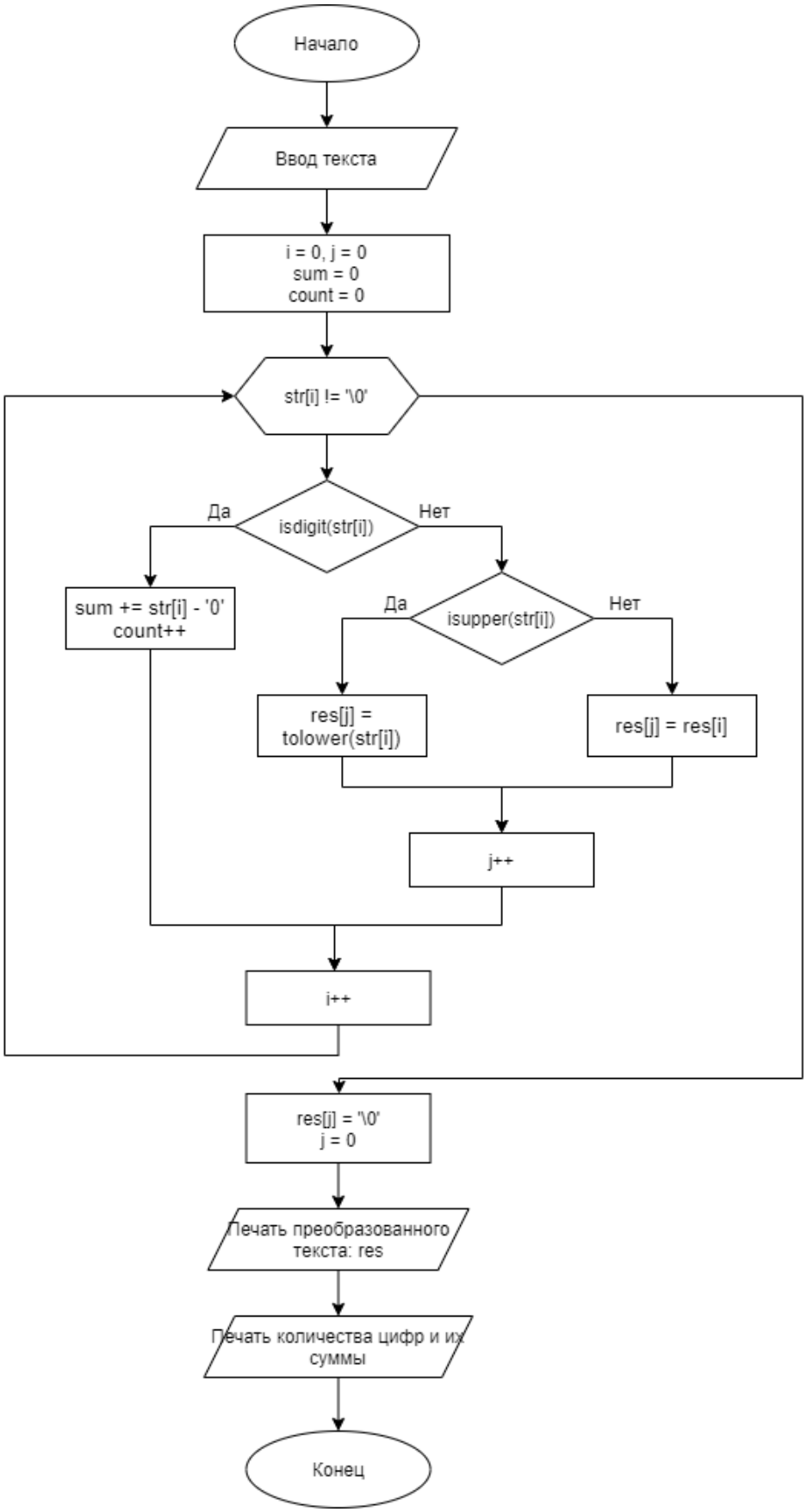


10) void zadacha4(int N, int M, int A[maxLen][maxLen]); - функция, подсчитывающая количество нулей в каждом столбце матрицы и выводящая номера столбцов, в которых нет нулей.



11) void zadacha5(); - функция, считывающая текст и удаляющая из него все цифры, подсчитывающая количество и сумму удаленных цифр, заменяющая все заглавные латинские буквы на маленькие.



## Листинг программного продукта на языке C++

```
#include <iostream>
#include <cstdlib>
#include <cmath>
#include <cstring>
#include <climits>
#define maxLen 30 //максимальная допустимая размерность массива

using namespace std;

void clear_screen();//функция очистки экрана
void input1(int M, double A[maxLen], bool isRand);//заполнение 1 мерного массива
void output1(int M, double A[maxLen]);//вывод 1 мерного массива
void input2(int N, int M, int A[maxLen][maxLen], bool isRand);//заполнение 2 мерного массива
void output2(int N, int M, int A[maxLen][maxLen]);//вывод 2 мерного массива
double zadacha1_1(int N, double A[maxLen]);//сумма элементов между 1 отрицательным и 2
отрицательным элементом
void zadacha1_2(int N, double A[maxLen]);//в первой его половине элементы, модуль которых не
превышает 1, потом все остальные.
int zadacha2_1(int N, int A[maxLen][maxLen]);//вернет номер первой из строк, содержащих хотя бы
один положительный элемент.
void zadacha2_2(int N, int A[maxLen][maxLen]);//меняет местами строку, в которой он находится и
столбец, в котором он находится.
void zadacha3(double A[maxLen]);//функция, определяющая и выводящая номера минимальных
элементов
void zadacha4(int N, int M, int A[maxLen][maxLen]);//выводит номера столбцов, в которых нет нулей.
//функция считывающая текст и удаляющая из него все цифры, подсчитывающая количество и
сумму удаленных цифр,
//заменяющая все заглавные латинские буквы на маленькие .
void zadacha5();

int main()
{
   srand(time(NULL));//для генерации случайных чисел
   setlocale(LC_ALL, "rus");
   int userChoice = -1;//для выбора действия
   while(userChoice != 0){
      clear_screen();
      cout << "###ГЛАВНОЕ МЕНЮ###" << endl;
      cout << "Задача 1. Нажмите 1" << endl;
      cout << "Задача 2. Нажмите 2" << endl;
      cout << "Задача 3. Нажмите 3" << endl;
      cout << "Задача 4. Нажмите 4" << endl;
      cout << "Задача 5. Нажмите 5" << endl;
      cout << "Для выхода нажмите 0" << endl;
      cin >> userChoice;
      switch (userChoice) {
      case 0:
         //выход из программы
         return 0;
      case 1:{
         //1 задание
         double Array[maxLen];
         int n = 0;
         do{
```

```
            cout <<"Введите размер массива n > 0 и n <= 20\n";
            cin >> n;
        }while (n <= 0 || n > maxLen);
        input1(n, Array, false);//считвание массива
        cout << "Введеный массив"<<endl;
        output1(n, Array);
        cout << "Сумма элементов массива, расположенных между 1 и 2 отрицательными
элементами: " << zadacha1_1(n, Array) <<endl;
        zadacha1_2(n, Array);
        output1(n, Array);
        cout << "Для продолжения нажмите Enter."<<endl;
        cin.get(); cin.get();
        break;
      }
      case 2:{
        //2 задание
        int Array[maxLen][maxLen];
        int n = 0;
        do{
            cout <<"Введите размерность матрицы n > 0 и n <= 20\n";
            cin >> n;
        }while (n <= 0 || n > maxLen);
        input2(n, n, Array, false);//считвание массива
        cout << "Введеный массив"<<endl;
        output2(n, n, Array);
        int s_num = zadacha2_1(n, Array);
        if(s_num == -1)
            cout << "Все строки с 0 или отрицательными элементами " <<endl;
        else
            cout << "Номер строки хоть с одним положительным элементом: "<< s_num <<endl;

        zadacha2_2(n, Array);
        output2(n, n, Array);
        cout << "Для продолжения нажмите Enter."<<endl;
        cin.get(); cin.get();
        break;
       }
      case 3:{
        //3 задание
        double C[maxLen];
        int n = 20;//размер массивов
        input1(n, C, true);//заполнение массива
        cout << "Введенный массив "<<endl;
        output1(n, C);
        zadacha3(C);
        cout << "Для продолжения нажмите Enter."<<endl;
        cin.get(); cin.get();
        break;
      }
      case 4:{
        //4 задание
        int Array[maxLen][maxLen];
        int width = 6, height = 5;
        input2(width, height, Array, true);//заполнение массива
        cout << "Полученный массив"<<endl;
        output2(width, height, Array);
        zadacha4(width, height, Array);
        cout << "Для продолжения нажмите Enter."<<endl;
```

```
            cin.get(); cin.get();
            break;
        }
        case 5:{
            zadacha5();
            cout << "Для продолжения нажмите Enter."<<endl;
            cin.get();
            break;
        }
        }

    }
}

void zadacha5(){
    cout << "Вводите текст:"<<endl;
    char str[1000] = "";//строка на 1000 символов
    char res[1000];//массив без цифр
    cin.get();
    cin.getline(str, 1000); // считываем текст
    int i = 0, j = 0;
    int sum = 0;//сумма цифр
    int count = 0;// количество цифр
    while(str[i] != '\0'){
        if(isdigit(str[i])){
            sum += str[i] - '0';//суммируем
            count++;//увеличиваем счетчик
        }
        else{
            if(isupper(str[i]))//если в верхнем регистре меняем на нижний
                res[j] = tolower(str[i]);
            else
                res[j] = str[i];//записываем в строку результата без изменений
            j++;
        }
        i++;
    }
    res[j] = '\0';
    j = 0;
    cout << "\nПреобразованный текст" << endl;
    while(res[j] != '\0'){
        cout << res[j++];
    }
    cout << endl << "Количество численных символов: " << count << endl;
    cout <<  "Сумма элементов: " << sum << endl;
}

void zadacha4(int M, int N, int A[maxLen][maxLen]){
    cout << "Столбцы не содержащие 0" << endl;
    for (int i = 0; i < M; i++){
        int zeros = 0;
            for (int j = 0; j < N; j++){
                if(A[j][i] == 0){//нашли 0
                    zeros++;
                }
            }
        if(zeros == 0)//выводим результат
            cout << i << " ";
```

```
    }
    cout << endl;
}

void zadacha3(double A[maxLen]){
    int min_left = INT_MAX, min_right = INT_MAX;

    for(int i = 0; i < 20; i++){//ищем минимальный элемент в 2 половинах массива
        if(i < 10 && A[i] < min_left)
            min_left = A[i];
        if(i >= 10 && A[i] < min_right)
            min_right = A[i];

    }
    cout << "Минимальный элемент слева: " << min_left <<endl;
    cout << "Индексы: " << endl;
    for(int i = 0; i < 10; i++){
        if(A[i] == min_left)
            cout << i << " ";
    }
    cout << endl;
    cout << "Минимальный элемент справа: " << min_right << endl;
    cout << "Индексы: " << endl;
    for(int i = 10; i < 20; i++){
        if(A[i] == min_right)
            cout << i << " ";
    }
    cout <<endl;
}

void zadacha2_2(int N, int A[maxLen][maxLen]){//поиск максимального элемента и преобразование
матрицы
    //если матрица 0 размерности то ничего не делаем
    if(N > 0){
        int max_i = 0, max_j = 0;
        int max_elem = A[max_i][max_j];

        //поиск максимального элемента
        for(int i = 0; i < N; i++){
            for(int j = 0; j < N; j++){
                if(A[i][j] > max_elem){
                    max_elem = A[i][j];//фиксируем максимальные координаты
                    max_i = i;//фиксируем максимальные координаты
                    max_j = j;
                }
            }
        }
        cout << "Максимальный элемент: " << max_elem << endl;
        cout << "Преобразованный массив"<<endl;
        for(int iter_i = 0, iter_j = 0; iter_i < N && iter_j < N;){
            if(max_j == iter_j){//не изменяем местоположение максимального элемента
                iter_j++;
                continue;
            }
            if(max_i == iter_i){//не изменяем местоположение максимального элемента
                iter_i++;
                continue;
            }
```

```
            //меняем местами элементы
            int tmp = A[max_i][iter_j];
            A[max_i][iter_j] = A[iter_i][max_j];
            A[iter_i][max_j] = tmp;
            iter_j++;
            iter_i++;
        }
    }
}
int zadacha2_1(int N, int A[maxLen][maxLen]){//поиск строки с хотябы 1 положительным элементом
    bool isfound = false;
    for(int i = 0; i < N; i++){
        for(int j = 0; j < N; j++){
            if(A[i][j] > 0){//если нашли элемент то выходим из цикла
                isfound = true;
                break;
            }
        }
        if(isfound)//выходим из функции
            return i;
    }
    if(!isfound)//если обошлм весь цикл и не нашли то возвращаем -1
        return -1;
}

double zadacha1_1(int N, double A[maxLen]){//сумму элементов массива, расположенных между
первым и вторым отрицательными элементами;
    double sum = 0;
    bool first_negative = false;//флаг был ли найден отриц элемент
    for(int i = 0; i < N; i++){
        if(A[i] < 0 && !first_negative)//если это первый отрицательный элемент
            first_negative = true;
        else
        {
            if(A[i] < 0 && first_negative)//нашли 2 отрицательный - алгоритм завершаем
                break;
            else
            {
                if(A[i] >= 0 && first_negative)//суммируем до 2 отрицательного
                    sum += A[i];
            }

        }
    }
    return sum;
}
void zadacha1_2(int N, double A[maxLen]){//преобразование массива алгоритм похож на сортировку
пузырьком
    for (int i = 0; i < N; ++i) {
            if (abs(A[i]) <= 1)//если элемент меньше по модулю то ничего не делаем
                continue;
            int j = i + 1;
            for (; j < N; ++j) {
                if (abs(A[j]) <= 1) {//иначе меняем местами с первым элементом меньше 1
                    double tmp = A[j];
                    A[j] = A[i];
                    A[i] = tmp;
                    break;
```

```
            }
          }
           if (j == N)
             break;
      }
}

void input1(int M, double A[maxLen], bool isRand){
   if(isRand){
      int a = -15, b = 21;
      for (int i = 0; i < M; i++)
         A[i] = rand()%(b - a) + a;//функция генерации случайных чисел
   }
   else{
      cout << "Введите значения элементов массива через Enter: \n";
      for (int i = 0; i < M; i++){
         double a;
         cin >> a;
         A[i] = a;
      }
   }
}
void output1(int M, double A[maxLen]){
   for (int i = 0; i < M; i++)
         cout << A[i] << " ";
   cout << endl;
}
void input2(int N, int M, int A[maxLen][maxLen], bool isRand){
   if(isRand){//если числа генерируются случайно
      int a = 0, b = 0;
      cout <<"Введите левую границу\n";
      cin >> a;

      do{
         cout <<"Введите правую границу b > a\n";
         cin >> b;
      }while (b <= a);
         for (int i = 0; i < M;i++)
            for (int j = 0; j < N;j++)
               A[i][j] = rand()%(b - a) + a;//функция генерации случайных чисел на [a,b)
   }
   else{//если вводятся вручную
      for (int i = 0; i < M; i++){
            cout << "\n Введите значения элементов " << i <<" -й строки массива: \n";
            for (int j = 0; j < N; j++){
               cout <<"x["<<i<<"]["<<j<<"]= ";
               double a;
               cin >> a;
               A[i][j] = a;
            }
      }
   }

}
void output2(int N, int M, int A[maxLen][maxLen]){
   for (int i = 0; i < M; i++){
      for(int j = 0; j < N; j++){
         cout << A[i][j] << " ";
```

```
      }
      cout<<endl;
   }
}
void clear_screen()
{
#ifdef WINDOWS
   system("cls");//очистка экрана для windows
#else
   // Assume POSIX
   system ("clear");
#endif
}
```

## 3. Результаты тестового прогона программы

1) Главное меню

```
Задача 1. Нажмите 1
Задача 2. Нажмите 2
Задача 3. Нажмите 3
Задача 4. Нажмите 4
Задача 5. Нажмите 5
Для выхода нажмите 0
```

2) Задача 1

```
Введеный массив
0.1 -0.1 0 12 9 -0.11 0.22 0.31 5 1.1
Сумма элементов массива, расположенных между 1 и 2 отрицательными элементами: 21
Преобразованный массив
0.1 -0.1 0 -0.11 0.22 0.31 9 12 5 1.1
```

```
Введеный массив
11 0.1 2 3 -1
Сумма элементов массива, расположенных между 1 и 2 отрицательными элементами: 0
Преобразованный массив
0.1 -1 2 3 11
```

```
Введеный массив
-1 0.1 0.01 0.001 -1
Сумма элементов массива, расположенных между 1 и 2 отрицательными элементами: 0.111
Преобразованный массив
-1 0.1 0.01 0.001 -1
```

3) Задача 2

```
Введеный массив
0 -1 -2
-1 -3 -4
-5 -6 -7
Все строки с 0 или отрицательными элементами
Максимальный элемент: 0
Преобразованный массив
0 -1 -5
-1 -3 -4
-2 -6 -7
```

```
Введеный массив
1 2 3 100
4 5 6 7
8 9 10 11
12 13 14 15
Номер строки хоть с одним положительным элементом: 0
Максимальный элемент: 100
Преобразованный массив
7 11 15 100
4 5 6 1
8 9 10 2
12 13 14 3
```

```
Введеный массив
0 0 0
1 2 3
10 2 3
Номер строки хоть с одним положительным элементом: 1
Максимальный элемент: 10
Преобразованный массив
2 0 0
3 2 3
10 0 1
```

4) Задача 3

```
Введенный массив
8 6 -1 18 11 -2 -1 -9 9 -4 17 -7 0 13 18 7 -10 20 13 6
Минимальный элемент слева: -9
Индексы:
7
Минимальный элемент справа: -10
Индексы:
16
```

```
Введенный массив
-5 -4 -6 20 3 -9 -12 0 9 -12 -11 13 9 3 -10 15 -4 5 -15 0
Минимальный элемент слева: -12
Индексы:
6 9
Минимальный элемент справа: -15
Индексы:
18
```

```
Введенный массив
-4 -2 8 12 -10 -15 14 -4 20 -14 -3 10 -3 6 -10 -4 -8 9 -9 17
Минимальный элемент слева: -15
Индексы:
5
Минимальный элемент справа: -10
Индексы:
14
```

5) Задача 4

```
Введите левую границу
1
Введите правую границу b > a
3
Полученный массив
1 2 2 1 1 1
1 1 2 1 2 2
2 1 1 1 2 2
1 1 2 2 1 2
1 1 1 1 2 1
Столбцы не содержащие 0
0 1 2 3 4 5
```

```
Введите левую границу
0
Введите правую границу b > a
3
Полученный массив
2 2 1 0 1 0
2 0 0 1 1 2
1 2 2 1 1 2
1 2 2 0 2 0
1 0 2 1 1 0
Столбцы не содержащие 0
0 4
```

```
Введите левую границу
0
Введите правую границу b > a
5
Полученный массив
3 4 0 2 1 4
4 1 1 4 4 2
2 2 0 1 2 2
4 4 2 3 3 2
3 1 3 2 0 4
Столбцы не содержащие 0
0 1 3 5
```

6) Задача 5

```
Вводите текст:
HELLO 55 hello

Преобразованный текст
hello  hello
Количество численных символов: 2
Сумма элементов: 10
```

```
Вводите текст:
NO NUMBERS HERE.

Преобразованный текст
no numbers here.
Количество численных символов: 0
Сумма элементов: 0
```

```
Вводите текст:
ThErE OnE 2 3 4 55

Преобразованный текст
there one
Количество численных символов: 5
Сумма элементов: 19
```

# ЗАКЛЮЧЕНИЕ

Программа была разработана в рамках контрольной работы по курсу «Программирование и основы алгоритмизации», и отвечает всем требованиям, предъявленным заказчиком (см. пункт «Постановка задачи»).

Разработка программного средства проводилась в полном соответствии с современными тенденциями в технологии программирования, а именно, с использованием принципов модульного и структурного проектирования программных средств. Для решения каждой из подзадач были реализованы отдельные функции, которые могут быть настроены на различные входные типы данных. Данный подход существенно сократил программный код и, соответственно, простоту восприятия кода.

Все задачи варианта объединены одним общим интерфейсом. Выполнение каждой задачи осуществляется как последовательный вызов всех необходимых подпрограмм. Предусмотрена возможность повтора выполнения каждой задачи через пункты главного меню. Результаты печатаются с максимально возможными комментариями. Размерность массивов задается пользователем программного средства. Производится обработка исключительных ситуаций, например когда пользователь вводит некорректные данные.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ


1. Б.И.Березин, С.Б.Березин "Начальный курс С и С++" -М.: ДИАЛОГ-МИФИ, 1996 г.
2. В.А.Скляров "Язык С++ и объектно-ориентированное программирование" - Мн.: Выш. шк., 1997 г.
3. С.Поттс, Т.С.Монк "BORLAND С++ в примерах" - Мн.: ООО "Попурри", 1996 г.