

Министерство образования Российской Федерации  
Государственное образовательное учреждение  
высшего профессионального образования  
Московский государственный технологический университет  
«СТАНКИН»

Кафедра «Управление и информатика в технических системах»  
Учебный курс «Информатика»

ЛАБОРАТОРНАЯ РАБОТА №6  
по теме «Модульное программирование»  
Вариант №99

Выполнил:

студент гр. М-4-10

15.04.09

(дата)

А.Б. Иванов

(подпись)

Принял:

преподаватель

(дата)

В.Г. Петров

(подпись)

Москва – 2009 год

## СОДЕРЖАНИЕ

<b>1.ТЕКСТ ЗАДАНИЯ.....</b>	<b>3</b>
<b>2.БЛОК-СХЕМЫ АЛГОРИТМОВ.....</b>	<b>4</b>
<b>3.ИДЕНТИФИКАТОРЫ.....</b>	<b>9</b>
3.1.Тип данных NormFunction.....	9
3.2.Переменные.....	9
3.3.Функции.....	10
<b>4.ПРОГРАММА.....</b>	<b>11</b>
4.1.Текст программы.....	11
4.1.1. NormUsage.cpp.....	11
4.1.2. Norm.h.....	13
4.1.3. Norm.cpp.....	13
4.2.Результат выполнения программы.....	14
<b>5.СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....</b>	<b>15</b>

## 1. ТЕКСТ ЗАДАНИЯ

Напишите программу, состоящую из двух модулей:

а) DLL - содержит набор функций для вычисления норм векторов.

б) EXE - динамически загружает модуль DLL и использует его функции для вычисления евклидова расстояния, манхеттенского расстояния и максимального элемента массива.

## 2. БЛОК-СХЕМЫ АЛГОРИТМОВ

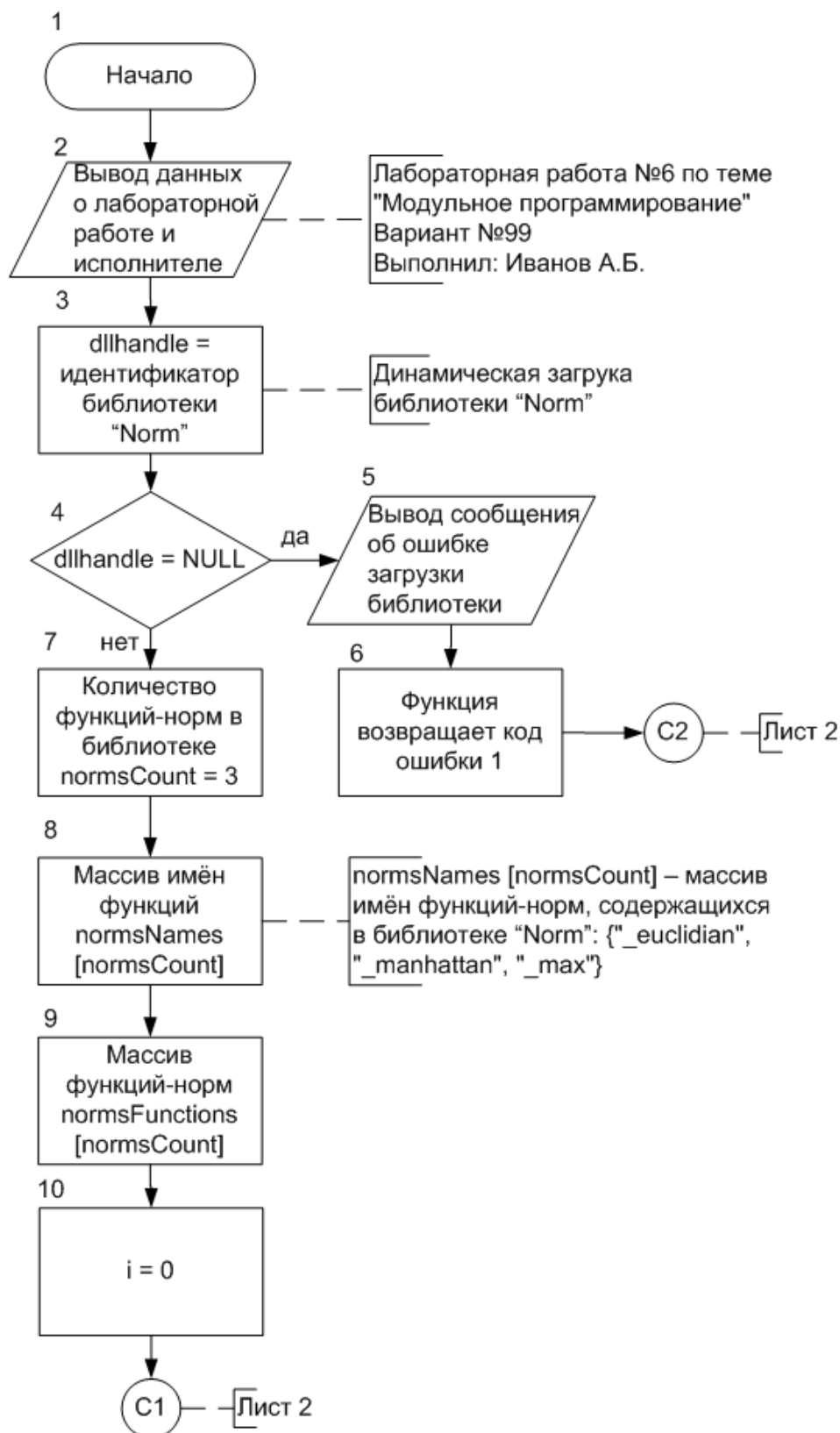


Рис. 1. Лист 1. Основная блок-схема программы. Динамическая загрузка DLL.

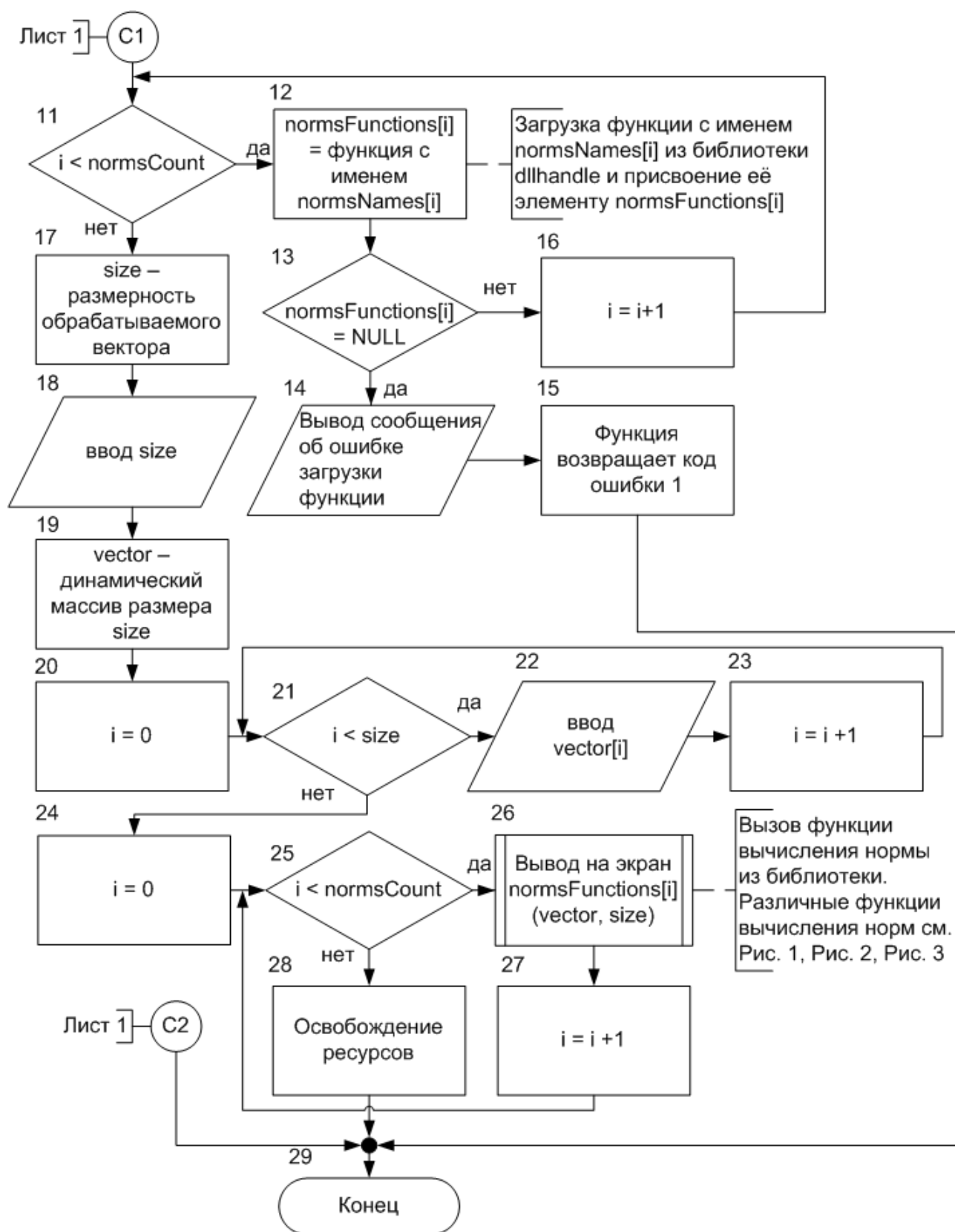


Рис. 1. Лист 2. Основная блок-схема программы. Объявление вектора и вычисление его нормы различными методами.

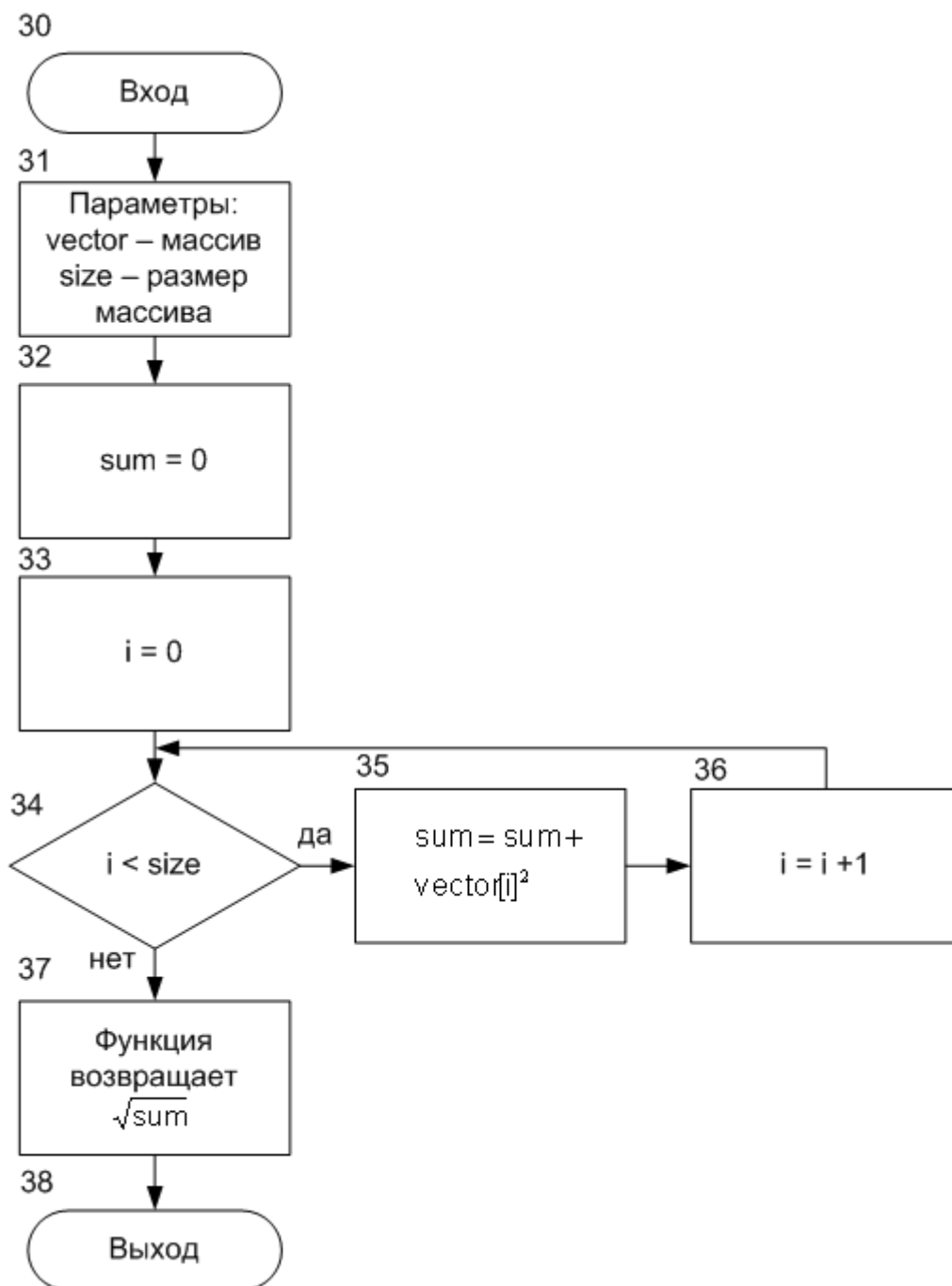


Рис. 2. Функция `double euclidian(double* vector, int size)`. Вычисление нормы при помощи евклидова расстояния.

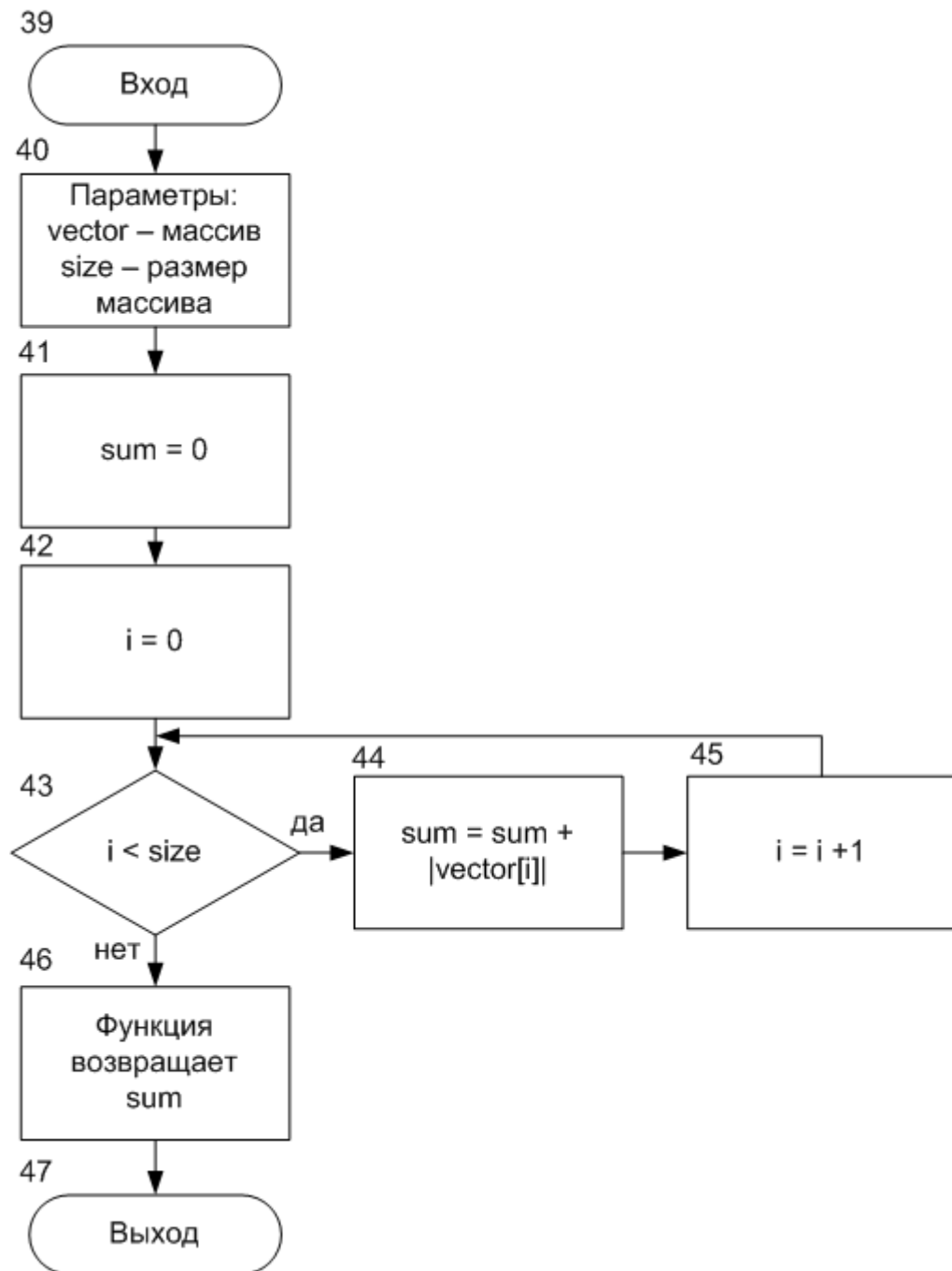


Рис. 3. Функция `double manhattan(double* vector,int size)`. Вычисление нормы при помощи манхеттенского расстояния.

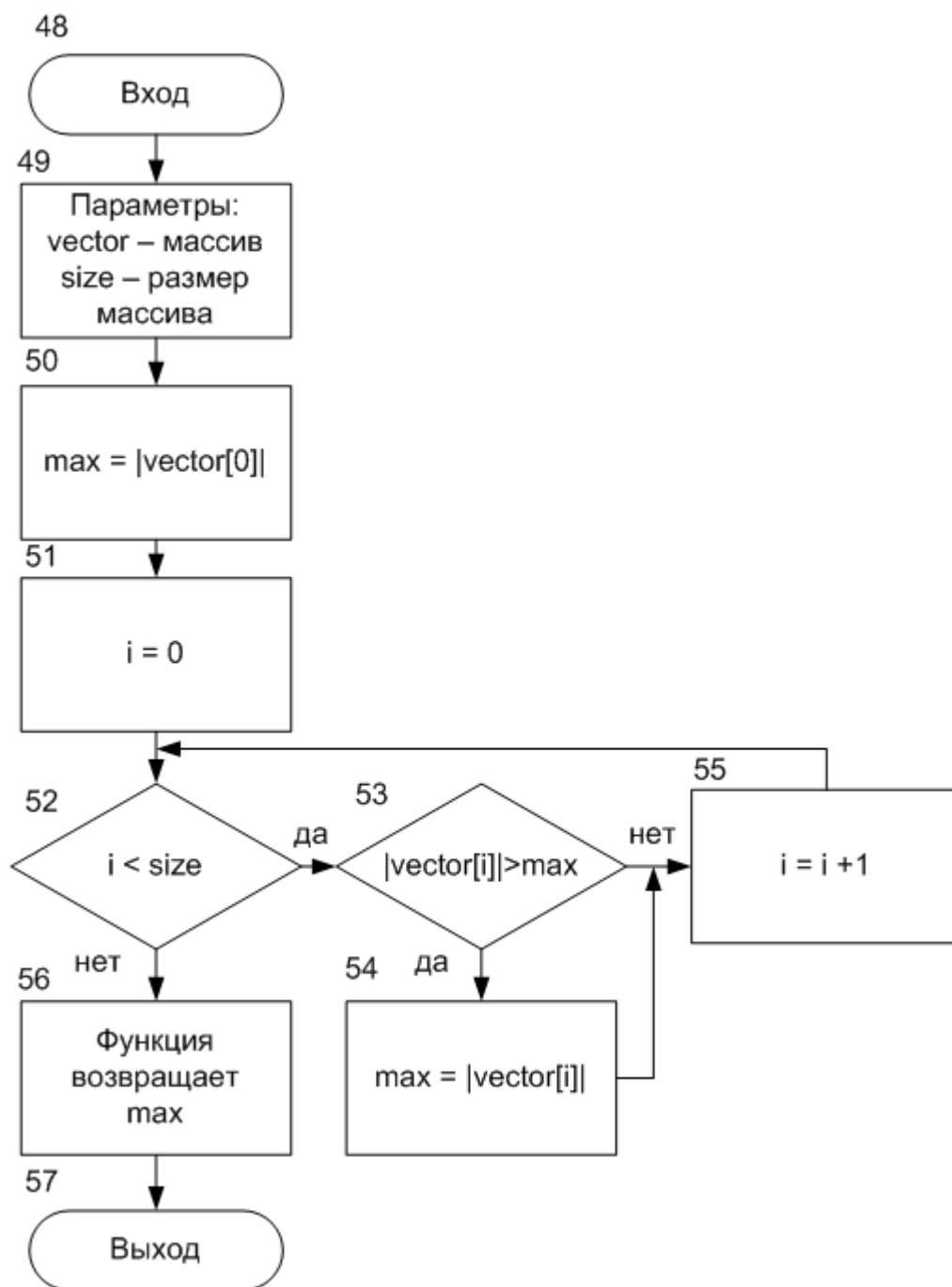


Рис. 4. Функция `double max(double* vector,int size)`. Вычисление нормы при помощи определения максимального значения.



### 3. ИДЕНТИФИКАТОРЫ

#### 3.1. Тип данных NormFunction

Описание: указатель на функцию-норму.

#### 3.2. Переменные

таблица 1.

№	Имя	Область видимости	Тип	Описание и назначение
1.	dllhandle	main(int, char*[])	целочисленное (HINSTANCE)	идентификатор библиотеки
2.	normsCount	main(int, char*[])	целочисленная константа (const int)	количество функций-норм в библиотеке
3.	normsNames	main(int, char*[])	массив строк (const char*[])	массив имён функций-норм в библиотеке
4.	normsFunctions	main(int, char*[])	массив NormFunction (NormFunction[] )	массив функций-норм, загруженных из библиотеки
5.	i	main(int, char*[])	целочисленное (int)	переменная цикла
6.	size	main(int, char*[])	целочисленное (int)	размерность вектора
7.	vector	main(int, char*[])	указатель на действительное число (double*)	массив, хранящий вектор, для которого вычисляется норма
8.	sum	euclidian(double*,int)	действительное (double)	сумма квадратов координат вектора
9.	i	main(int, char*[])	целочисленное (int)	переменная цикла
10.	sum	manhattan(double*,int)	действительное (double)	сумма модулей координат вектора

11.	i	main(int, char*[])	целочисленное (int)	переменная цикла
12.	max	max(double*, int)	действительное (double)	максимальный элемент вектора
13.	i	main(int, char*[])	целочисленное (int)	переменная цикла

### 3.3. Функции

таблица 2.

№	Имя	Тип возвращаемого значения	Описание и назначение функции	Список формальных параметров (тип, название, назначение)
1.	double <b>euclidian</b> (double* <i>vector</i> , int <i>size</i> )	действительное (double)	вычисление нормы при помощи евклидова расстояния	double* <i>vector</i> — указатель на исходный вектор int <i>size</i> — размерность вектора
2.	double <b>manhattan</b> (double* <i>vector</i> , int <i>size</i> )	действительное (double)	вычисление нормы при помощи манхеттенского расстояния	double* <i>vector</i> — указатель на исходный вектор int <i>size</i> — размерность вектора
3.	double <b>max</b> (double* <i>vector</i> , int <i>size</i> ).	действительное (double)	вычисление нормы при помощи определения максимального значения	double* <i>vector</i> — указатель на исходный вектор int <i>size</i> — размерность вектора

## 4. ПРОГРАММА

### 4.1. Текст программы

#### 4.1.1. NormUsage.cpp

```
//-----

#pragma hdrstop

//-----

#pragma argsused

#include <windows.h>
#include <iostream>
#include <stdlib.h>

using namespace std;

// Объявление типа функции-нормы
typedef double (__import * NormFunction)(double*, int);

int main(int argc, char* argv[])
{
    // Вывод на экран информации о лабораторной работе
    // и исполнителе
    cout << "Лабораторная работа №6 ";
    cout << "по теме \"Модульное программирование\" \" << endl;
    cout << "Вариант №99\" << endl;
    cout << "Выполнил: Иванов А.Б.\" << endl;

    // загрузка библиотеки и получение её идентификатора
    HINSTANCE dllhandle = LoadLibrary("norm");

    // проверка, была ли загружена библиотека
    if (dllhandle == NULL)
    {
        // если библиотека не была загружена - вывод сообщения
        // об ошибке и завершение работы программы
        cout << "error while loading library\" << endl;
        system("pause");
        return 1;
    }

    // объявление количества функций нормы в библиотеке
    const int normsCount = 3;
    // задание имён функции нормы в библиотеке
```

```

const char* normsNames[normsCount] =
{"_euclidian", "_manhattan", "_max"};

// объявление массива функций-норм
NormFunction normsFunctions[normsCount];

// цикл заполнения массива функций-норм функциями
// из библиотеки
for (int i = 0; i < normsCount; i++)
{
    // загрузка функции-нормы по её имени
    normsFunctions[i] =
        (NormFunction)
        GetProcAddress(dllhandle, normsNames[i]);

    // проверка была ли загружена функция-норма
    if (normsFunctions[i] == NULL)
    {
        // если функция не была загружена - вывод сообщения
        // об ошибке и завершение работы программы
        cout << "error while getting function "
            << normsNames[i] << endl;
        system("pause");
        return 1;
    }
}

// ввод вектора, норму которого предстоит найти
int size;
cout << "Введите размерность вектора: ";
cin >> size;

double* vector = new double[size];

// Ввод элементов вектора
cout << "Введите элементы вектора: " << endl;
for (int i = 0; i < size; i++)
{
    cin >> vector[i];
}

// Вычисление нормы вектора при помощи функций из библиотеки
cout << "Вычисленные нормы: " << endl;
for (int i = 0; i < normsCount; i++)
{
    cout << normsNames[i] << " : "
        << normsFunctions[i](vector, size) << endl;
}

delete[] vector;
FreeLibrary(dllhandle);

```

```

        system("pause");

    return 0;
}

```

#### 4.1.2. Norm.h

```

//-----

#ifndef normH
#define normH

// функция вычисления евклидова расстояния
extern "C" __declspec(dllexport)
double euclidian(double* vector, int size);

// функция вычисления манхеттенского расстояния
extern "C" __declspec(dllexport)
double manhattan(double* vector, int size);

// функция вычисления максимума
extern "C" __declspec(dllexport)
double max(double* vector, int size);

//-----
#endif

```

#### 4.1.3. Norm.cpp

```

//-----

#pragma hdrstop

#include <math.h>
#include "norm.h"

double euclidian(double* vector, int size)
{
    double sum = 0;

    for (int i = 0; i < size; i++)
    {
        sum += pow(vector[i], 2);
    }

    return sqrt(sum);
}

double manhattan(double* vector, int size)
{
    double sum = 0;

```

```

    for (int i = 0; i < size; i++)
    {
        sum += fabs(vector[i]);
    }

    return sum;
}

double max(double* vector, int size)
{
    double max = fabs(vector[0]);

    for (int i = 1; i < size; i++)
    {
        if (fabs(vector[i]) > max)
        {
            max = fabs(vector[i]);
        }
    }

    return max;
}

```

```

//-----
#pragma package(smart_init)

```

## 4.2. Результат выполнения программы

Лабораторная работа №6 по теме "Модульное программирование"

Вариант №99

Выполнил: Иванов А.Б.

Введите размерность вектора: 3

Введите элементы вектора:

-4

0

3

Вычисленные нормы:

\_euclidian : 5

\_manhattan : 7

\_max : 4

Для продолжения нажмите любую клавишу ...

## 5. СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Дейтел Х., Дейтел П. Как программировать на С++. Пер. с англ. – М.: ЗАО «Издательство БИНОМ», 1998 г. – 1024 с.: ил.
2. Керниган Б. Ритчи Д. Язык программирования Си. Пер. С англ. 3-е изд., испр. - Спб.: «Невский Диалект», 2001 г. - 354 с.: ил.
3. Страуструп Б. Язык программирования С++. Специальное издание — The C++ programming language. Special edition. — М.: Бином-Пресс, 2007. — 1104 с.
4. ГОСТ 19.701-90 (ИСО 5807-85). Единая система программной документации. Схемы алгоритмов, программ данных и систем. Условные обозначения и правила выполнения.