Лабораторные работы по курсу «Программирование»

Данный вариант документа не окончателен и может быть дополнен в течение семестра

Преподаватели:

Иван Григорьевич Корниенко

Алексей Константинович Федин

Санкт-Петербург
2019

1 Назначение курса

1 Упражнения и лаборатории курса «Программирование» предназначены для практического закрепления навыков, полученных при прослушивании лекционного материала по соответствующему курсу.

2 Общие требования

2.1 Порядок выполнения практической работы

1 Выполнение лабораторной работы начинается с получения задания.
2 Студент должен ознакомиться с заданием на лабораторную работу. В случае если задание непонятно, он может проконсультироваться у преподавателя.
3 Практическая работа выполняется индивидуально или в группах (не более трех участников). Номер варианта определяется порядковым номером студента в списке, выложенном в папке курса. Первый в списке получает первый вариант. Если предположить, что вариантов шесть, то седьмой студент по списку получает снова первый вариант. В случае выполнения работы в группе, выбирается наименьший из вариантов участников. Студент не может выбрать другой вариант.
4 После получения и согласования с преподавателем задания на лабораторную работу студент приступает к выполнению теоретической части работы. В ходе теоретической части работы студент разрабатывает:
   - требования к программному продукту;
   - математические методы и алгоритмы;
   - структуру программы;
   - форматы представления данных.
5 После разработки теоретической части студент представляет преподавателю разработанный материал, и, получив разрешение преподавателя, приступает к непосредственному кодированию разработанной программы на компьютере.
6 Написав программу, студент должен её тщательно протестировать. После этого он должен сдать программу преподавателю, продемонстрировав её работу и исходный код.
7 По завершении работы студент оформляет отчёт, к которому прилагается исходный код программы.

2.2 Требования к программному коду

1 Все идентификаторы (названия переменных, функций, классов и модулей) должны отражать назначение именуемых объектов. Все называния должны быть даны на английском языке. Не допускается использование транслитерации и других нестандартных приёмов именования.
2 Все константные литералы в коде должны быть объявлены как константы (const).
3 Стиль оформления кода (отступы, именование переменных, функций, классов и т.д.) должен последовательно соблюдаться в работе и соответствовать одному из общепринятых стандартов (например, Google C++ Style Guide).

2.3 Требования к программе

1 Программа при запуске должна выводить подробную информацию о назначении программы, авторе, решаемой задаче и предоставляемых результатах. В случае если программа выполнена с использованием графического интерфейса пользователя (приложение Windows) – программа должна выводить соответствующую информацию при запуске и в ответ на выбор соответствующего пункта меню. В этом случае должна быть возможность отключить вывод приветствия при запуске программы.

2 Выполнение консольной программы должно быть закольцовано. Завершение работы программы должно производиться только при выборе соответствующего пункта меню.

2.4 Общие требования к работам

1 Все ошибки, которые могут возникнуть в программе должны быть обработаны с выдачей соответствующих диагностических сообщений. Метод обработки ошибок должен быть единым для всей программы.
2 Интерфейс пользователя должен быть полностью отделён от вычислительных процедур программы.
3 Все интерфейсы должны быть документированы.

2.4.1 Работа с файлами

1 В работах, требующих кроме консольного ввода-вывода (экран, клавиатура) файловый ввод-вывод, необходимо предоставить возможность:
   - вводить исходные данные из файла;
   - сохранять исходные данные в файле;
   - сохранять результат работы программе в файле.
2 Полный путь к файлам в каждом случае задает пользователь.
3 В случае обнаружения файла с заданным именем по указанному пути, пользователю предлагается перезаписать существующий файл или указать новый. Данная проверка продолжается до тех пор, пока пользователь не укажет уникальное имя или не выберет перезапись.

2.4.2 Операции ввода-вывода

1 Для реализации ввода-вывода (экранного и файлового) необходимо использовать потоковые библиотеки C++.
2 Консольный ввод-вывод должен осуществляться с помощью объектов `cin` и `cout`.
3 Файловый ввод-вывод должен осуществляться с использованием объектов `ifstream`, `ofstream`.

2.4.3 Разделение на модули

1 Программа должна быть разделена (минимум) на три независимых cpp-файла.
   - основной модуль, осуществляющий запуск программы;
   - модуль, осуществляющий пользовательский интерфейс;
   - модуль с используемыми алгоритмами.
2 Для каждого cpp-файла должен быть предоставлен соответствующий h-файл, содержащий интерфейс данного модуля.

2.5 Содержание отчёта

1 В отчёт входит описание:
   - постановки задачи;
   - исходных данных;
   - особых ситуаций;
   - математических методов и алгоритмов решения задачи;
   - форматов представления данных в памяти и на внешних носителях;
   - структуры программы;
   - модулей, функций и переменных программы;
   - блок-схем алгоритмов программы;
   - хода выполнения работы;
   - полученных результатов.

2 Отчёт оформляется на листах формата А4 с обязательным титульным листом, на котором указываются название работы; ФИО исполнителя; ФИО преподавателей и т.д.

2.5.1 Постановка задачи

1 Постановка задачи указывает, какая цель должна быть достигнута при разработке программы. Какую задачу должна решать программа, и в каких условиях будет функционировать.

2.5.2 Исходные данные

1 Исходными данными являются любые данные, которые программа получает для обработки.
2 Описание исходных данных должно содержать:
- семантику (назначение) данных;
- единицы изменения;
- представление в программе.

2.5.3 Особые ситуации

1 Под особыми ситуациями понимаются ситуации, в которых поведение программы может не соответствовать поведению, ожидаемому пользователем.
2 Все особые ситуации должны быть описаны и соответствующим образом обработаны в программе.
3 Примерами особых ситуаций являются:
- некорректный ввод;
- отсутствие ожидаемых программой файлов;
- возможное деление на ноль в ходе вычислений;
- нехватка оперативной памяти.

2.5.4 Математические методы и алгоритмы решения задач

1 Все используемые программой алгоритмы и математические методы решения задач должны быть описаны в специальном разделе в форме и полноте, достаточной для восприятия другими разработчиками.

2.5.5 Форматы представления данных

1 Для всех пользовательских типов данных (не являющихся частью языка) должны быть документированы назначение и мотивация выбора конкретного типа данных.
2 Должны быть документированы форматы всех внешних ресурсов. Структура данных, сохраняемых в файлах и т.д.

2.5.6 Структура программы

1 Разработанная структура программы (разделение на модули, интерфейсы, шаблоны проектирования) должна быть документирована.
2 Должна быть описана основная последовательность работы программы (вызова функций, методов и т.д.).
3 Все модули, функции, методы и пользовательские типы данных должны быть соответствующим образом документированы в отчёте.

2.5.7 Блок-схемы алгоритмов программы

1 Построение блок-схем алгоритмов регламентируется ГОСТ 19.701-90 (ИСО 5807-85) «Единая система программной документации. Схемы алгоритмов программ, данных и систем. Условные обозначения и правила выполнения».

2.5.8 Описание хода выполнения лабораторной работы

1 В отчёте должно содержаться подробное описание хода выполнения лабораторной работы.
2 Основное внимание должно быть уделено выполнению работы за компьютером.
3 Должны быть описаны все новые методы и приёмы, использованные в ходе выполнения лабораторной работы. Таковыми могут быть:
   - создание проекта и запуск программы;
   - работа с отладчиком (точки останова и протоколирования, просмотр значений переменных);
   - поиск ошибок в программе;
   - работа с устройствами ввода-вывода.
4 Из отчёта должно быть ясно:
   - как выполнялась лабораторная работа;
   - какие были проблемы и как они были решены;
   - какие проблемы остались нерешёнными;
   - какие моменты были или остались непонятными.

2.5.9 Результаты работы программы

1 Необходимо указать, какие результаты производит программа.
2 Необходимо указать в каком формате пользователь получает результат.
3 Должно быть приведено не менее трех вариантов результатов выполнения программы с различными исходными данными.

2.5.10 Исходный текст программы

1 Исходный текст программы распечатывается и прилагается к отчёту.

2.5.11 Документирование и комментирование исходного текста

1 Все пользовательские типы данных должны быть прокомментированы.
2 Все функции, классы и модули должны быть прокомментированы.
3 Каждый модуль (h или cpp) должен начинаться с комментария, указывающего его назначение, автора, используемые алгоритмы.
4 Каждая нетривиальная функция должна предваряться комментарием, описывающим:
   - назначение;
   - входные данные;
   - результаты.
5 В функциях, где соответствующее описание будет полезным, также следует описать:
   - предусловия;
   - постусловия;
   - инварианты.

2.6 Защита и сдача лабораторной работы

1 В весеннем семестре выполняются четыре лабораторные работы из данного документа. График сдачи лабораторных работ:

Лабораторная работа №	**Период защиты**
1	18.02 – 01.03
2	18.03 – 29.03
3	15.04 – 26.04
4	13.05 – 24.05

2 Лабораторная работа защищается преподавателям, ведущим лабораторные и практические работы.
3 Для защиты необходимо иметь отчёт о проделанной работе и продемонстрировать работоспособную программу.
4 Подпись за лабораторную работу выставляет преподаватель, которому работа была защищена.
5 Окончательная сдача лабораторных работ является допуском к экзамену (или зачёту) по предмету.

2.6.1 Окончательная сдача лабораторных работ

1 Для сдачи лабораторных работ, необходимо представить:
   - комплект отчётов по лабораторным работам;
   - папку с исходными кодами программ и выполняемыми модулями.
2 В папке должен содержаться файл readme.txt, в котором должно быть указано:
   - Ф.И.О. выполнившего работы;
   - год, название предмета, названия работ.

2.7 Пример выполнения задания (часть пунктов опущена)

При выполнении лабораторной работы будет использоваться компилятор Microsoft Visual Studio C++ 2017.

2.7.1 Постановка задачи

Методом Монте-Карло вычислить число «пи».

2.7.2 Исходные данные

В качестве исходных данных программа использует вводимое пользователем число испытаний при проведении эксперимента.

2.7.3 Особые ситуации

Необходимо рассмотреть следующие особые ситуации.

- Если пользователь ввёл число испытаний меньше одного, то эксперимент провести невозможно.
- Если пользователь ввёл число испытаний меньше разумного предела для проведения статистического эксперимента, результаты могут быть недостоверными.
- Если пользователь ввёл очень большое число испытаний, то эксперимент может занять значительное время, о чём пользователь должен быть предупреждён.

2.7.4 Математические методы и алгоритмы решения задач

Согласно постановке задачи для составления программы будет использован метод Монте-Карло, который заключается в случайном выборе координат точек внутри квадрата заданного размера.

Для каждой точки будет проверяться попадание во вписанную в квадрат окружность. Таким образом, по окончании эксперимента мы будем располагать двумя числами:

- N – число случайно выбранных точек;
- M – число точек, попавших внутрь вписанной окружности.

Поскольку нам известна площадь квадрата и площадь вписанной окружности, то мы можем вычислить отношение площадей этих фигур. Если принять сторону квадрата за единицу, то его площадь будет равна одной квадратной единице.

Площадь круга, вписанного в этот квадрат, может быть определена по формуле:

$$S = \pi r^2 = \frac{\pi d^2}{4} \quad (1)$$

Таким образом, число π можно выразить через площадь и диаметр вписанного круга следующим образом:

$$\pi = 4\frac{S}{d^2} \quad (2)$$

Площадь вписанного круга можно найти из отношения M к N. Они относятся друг к другу так, как относится площадь вписанной окружности к площади квадрата. А площадь квадрата нам известна – она единична. Исходя из этого, получаем полную формулу для вычисления числа π по известным нам числам M и N.

$$\pi = \frac{4}{d^2} \cdot \frac{M}{N} \quad (3)$$

Все величины в данной формуле нам известны. Однако, как было указано выше, M является статистической величиной, которая будет рассчитана в результате проверки попадания случайных точек в окружность. Для генерации набора точек и проверки их попадания в круг необходимо написать программу.

2.7.5 Форматы представления данных

Программа использует следующие переменные:

Таблица 1 – Переменные, используемы в программе

Имя	Тип	Описание
tries	int	Число попыток в эксперименте
in_circle	int	Число точек, попавших в окружность

Для задания минимального и максимального пределов числа экспериментов используются следующие константы

Таблица 2 – Константы, используемые в программе

Имя	Тип	Значение	Описание
LowerLimit	const int	100	Минимальное число экспериментов, обеспечивающих достоверность
UpperLimit	const int	10000000	Число экспериментов, при превышении которого вычисления могут быть долгими

Для задания координат точки на плоскости используется структура *Point2D*, в которой задаются x и y координаты точки.

2.7.6 Структура программы

В силу своей простоты программа помещена в одном исполняемом модуле. Программа разделена на несколько функций:

Таблица 3 – Функции, составляющие программу

Имя	Описание
GeneratePoint	Создание точки в заданных координатах
IsInsideCircle	Проверка на нахождение точки внутри окружности
ProcessInput	Обработка ввода пользователя

2.7.7 Описание хода выполнения лабораторной работы

1. В ходе лабораторной работы было создано решение (Solution) в интегрированной среде разработки Microsoft Visual Studio C++ 2017. В нём был создан проект.

2. После набора текста программы выяснилось, что вывод текста на экран консольного приложения работает неправильно из-за различия кодировок консольного приложения и среды разработки. Для решения этой проблемы была использована функция `setlocale(LC_ALL, "Russian")`, которая обеспечивает работу приложения с символами кириллицы.
3. Программа после запуска выдавала одни и те же результаты, хотя в коде использовался вызов функции `rand`, возвращающей случайное число. После изучения справочной системы выяснилось, что необходимо использовать функцию `srand` для начальной инициализации генератора случайных чисел. После этого программа стала работать правильно.
4. В начальном варианте программы переменная `InputSuccess` не была инициализирована, о чём свидетельствовало соответствующее предупреждение отладчика. Учитывая, что это автоматическая переменная, создаваемая на стеке, при выполнении программы могла возникать неоднозначность. Ошибка была исправлена инициализацией переменной в значение `false`.
5. В начальном варианте программы было перепутано условие выхода из цикла `while`. Было указано `while (InputSuccess)` в результате чего тело цикла ни разу не выполнялось. Для выявления проблемы было использовано пошаговое выполнение программы, в ходе которого выяснилась причина ошибки, и программа была исправлена.

2.7.8 Результаты работы программы

В результате вычислений программа выводит примерное значение числа «пи». Точность вычислений определяется числом проведённых экспериментов и качеством используемого генератора случайных чисел.

2.7.9 Исходный текст программы

```
[ Начало программы ---]
// Lab1.cpp
// Лабораторная работа №0.
// Использование языка C++ для математических расчётов
// Расчёт числа "пи" методом Монте-Карло
// Студент группы NNN, Фамилия Имя Отчество. 2019 год

#include <math.h>
#include <time.h>
#include <iostream>
#include <windows.h>
#include <string>

using namespace std;

// Структура, описывающая точку на плоскости
struct Point2D {
  double x,y;
};

// Генерация случайной точки с координатами x и y
// в интервале от нуля до единицы
Point2D GeneratePoint() {
  Point2D pt;
  pt.x = rand() / double(RAND_MAX);
  pt.y = rand() / double(RAND_MAX);
```

```
  return pt;
}

// Проверка нахождения точки внутри окружности
bool IsInsideCircle(const Point2D& pt) {
  const double circle_radius = 0.5;
  const Point2D circle = {0.5, 0.5};

  float dist_from_center =
        sqrt(pow(circle.x - pt.x, 2) + pow(circle.y - pt.y, 2));
  return dist_from_center < circle_radius;
}

// Обработка ввода пользователя
int ProcessInput() {
  const int LowerLimit = 100;
  const int UpperLimit = 10000000;
  bool InputSuccess = false;
  int tries;

  while (!InputSuccess) {
    cout << "Введите число экспериментов:";
    cin >> tries;

    if (tries <= 0) {
      cout <<
       "Число экспериментов должно быть положительным." << endl;
      continue;
    }
    InputSuccess = true;
  }
  if (tries < 100) {
    cout << "Результат может быть неточным." << endl;
  }
  if (tries > UpperLimit) {
    cout << "Вычисления могут быть длительными." << endl;
  }
  return tries;
}

// Основной модуль
int main() {
  setlocale(LC_ALL, "Russian");
  int in_circle = 0;
  int tries;
  tries = ProcessInput();

  // Инициализация генератора ПСЧ
  srand((unsigned)time(NULL));
  // Генерация и проверка точек
  for (int i = 1; i <= tries; ++i) {
    Point2D pt = GeneratePoint();
    if (IsInsideCircle(pt))
```

```
      ++in_circle;
  }
  // Вычисление числа "пи"
  cout << "Число пи = ";
  cout << 4 * (in_circle / double(tries)) << endl;

  return 0;
}
[--- Конец программы.]
```

3 Лабораторная работа №1. Использование массивов

При выполнении лабораторной работы должен использоваться массив в стиле языка С, то есть нельзя использовать контейнеры и алгоритмы библиотеки STL или других библиотек.

Необходимо выделить массив требуемого размера, запросив его у пользователя при запуске программы или считав из файла. В программе должны быть предусмотрены три варианта заполнения исходного массива: пользователем с клавиатуры, из файла и случайными числами.

В работе должны быть использованы методы вывода на экран с использованием различных цветов шрифта. Например, исходный и измененный массив (элементы массива) должны отличаться цветом.

3.1 Варианты заданий

1 Задан двумерный массив A из N строк и M столбцов. Сформировать одномерный массив B из отрицательных элементов массива A, выполнить сортировку массива B в порядке неубывания.
2 Определить среднее значение элементов матрицы. Найти далее индекс строки и столбца элемента массива, наиболее близкого к среднему значению.
3 Задан двумерный массив A из N строк и M столбцов. Задан одномерный массив B из трех элементов. Указать номера строк массива A, в которых есть фрагмент элементов, повторяющий массив B.
4 Задан квадратный массив A из N строк и N столбцов, элементами которого являются нули и единицы. Установить в нем наличие всех квадратов из единиц со стороной длины M. Если такой квадраты найдены, то вывести координаты их левых верхних углов.
5 Заданы две матрицы A и B размером N x N. Сформировать из них прямоугольную матрицу X размером N x 2N, включая в последовательно столбец из матрицы A, затем столбец из матрицы B.
6 Матрица A из N строк и N столбцов размещена в одномерном массиве по строкам. Удалить K-ю строку матрицы (K задано) из одномерного массива. Результат напечатать по строкам.
7 Матрица A из N строк и N столбцов размещена в одномерном массиве по строкам. Удалить K-й столбец матрицы (K задано) из одномерного массива. Результат напечатать по строкам.
8 Матрица A из N строк и N столбцов размещена в одномерном массиве по строкам. Поменять местами K-ю и L-ю строки матрицы (K и L заданы). Результат напечатать по строкам.
9 Задан массив A из N строк и M столбцов. Вывести на экран сумму значений всех элементов исходного массива. Заменить 1 элемент разностью между первым и вторым элементом, второй элемент разностью между вторым и третьим и т.д. Последний элемент остается неизменным. Вывести на экран сумму значений всех элементов полученного массива.

10 Матрица A из N строк и N столбцов размещена в одномерном массиве по строкам. Поменять местами K-й и L-й столбцы матрицы (K и L заданы). Результат напечатать по строкам.
11 В заданной матрице заменить K-ю строку и L-й столбец нулями, кроме элемента, расположенного на их пересечении.
12 У пользователя запрашивается размерность M x N матрицы. У матрицы заполняются ячейки первой строки. В дальнейшем четные строки заполняются по следующему правилу: $U_{i,j} = \left(U_{i,j-1} + U_{i+1,j-1}\right)/2$; нечетные: $U_{i,j} = \left|U_{i,j-1} + U_{i+1,j-1}\right|/2$.
13 Задан массив X размером N. Сформировать из него матрицу A, содержащую по L элементов в строке. Недостающие элементы в последней строке (если такие будут) заполнить нулями. Напечатать матрицу по строкам.
14 Задана матрица A размером N x N. Сформировать два одномерных массива. В один переслать по строкам верхний треугольник матрицы, включая элементы главной диагонали, в другой – нижний треугольник. Распечатать верхний и нижний треугольники по строкам.
15 Квадратная матрица задана в виде одномерного массива по столбцам. Напечатать верхний треугольник матрицы (включая элементы главной диагонали) по строкам.
16 Матрица, симметричная относительно главной диагонали, задана верхним треугольником в виде одномерного массива по строкам. Восстановить исходную квадратную матрицу и напечатать по строкам.
17 Задана квадратная матрица. Исключить из нее строку и столбец, на пересечении которых расположен максимальный элемент главной диагонали.
18 Задана матрица размером N x N. Найти максимальный по модулю элемент матрицы. Переставить строки и столбцы матрицы таким образом, чтобы максимальный по модулю элемент был расположен на пересечении K-й строки и K-го столбца.
19 Задан квадратный массив A из N строк и N столбцов, элементами которого являются нули и единицы. Установить в нем количество строк, в которых нули и единицы чередуются.
20 Задан квадратный массив A из N строк и N столбцов. Произвести «поворот по часовой стрелке» массива таким образом, чтобы элементы 1 строки стали элементами N столбца, второй строки элементами N-1 столбца.
21 Задан квадратный массив A из N строк и N столбцов, N – нечетное число. В предложенном массиве можно выделить две диагонали. Эти диагонали отчерчивают четыре треугольника с числами. Произвести замену верхнего и нижнего треугольников, правого и левого треугольников.
22 Задан квадратный массив A из N строк и N столбцов, N – нечетное число. В предложенном массиве можно выделить две диагонали. Эти диагонали отчерчивают четыре треугольника с числами. Произвести замену правого треугольника верхним, нижнего треугольника правым, левого треугольника нижним, верхнего треугольника левым.
23 Задан массив A из N строк и M столбцов, N и M четные. Создать массив из N/2 строк и M/2 столбцов. В полученном массиве вписать средние значения из четырех соседних элементов исходного массива.
24 Задан квадратный массив A из N строк и N столбцов, N – нечетное число. В предложенном массиве можно выделить две диагонали, среднюю строку и средний столбец. Необходимо поменять элементы первой диагонали с элементами центрального столбца, элементы второй диагонали с элементами центральной строки
25 У пользователя запрашивается размер матрицы M x N. Массив заполняется значениями от 0 до 9. Далее выполняется преобразование массива. При преобразовании на каждой итерации сначала находятся одинаковые значения в соседних ячейках по строкам и производится замена левого числа на его удвоенное значение, правого числа на случайное (от 0 до 9), а затем выполняется аналогичное действия для столбцов. В

случае получения значений ячейки после удвоения больше 9, значение уменьшается на 10. Ячейки, подверженные изменению на каждой итерации требуется выделить цветом. Цвета для изменений строк и столбцов различные. Вывод на экран модифицированного массива проводить после каждой итерации. Продолжение вычислений производить после подтверждения пользователя. Требуется продолжать итерационный процесс до случая отсутствия одинаковых значений в соседних ячейках.

26 У пользователя запрашивается ввести размер матрицы M x N. Программа выводит матрицу, в которой ячейки «раскрашены» наподобие шахматной доски. После подтверждения продолжения работы программы, требуется создать 2 матрицы, в которые будут перемещены «белые» и «черные» клетки. Вывести на экран 2 сформированные матрицы.

27 У пользователя запрашивается ввести размер матрицы M x N. Программа выводит матрицу и просит пользователя ввести число. В случае если введенное число присутствует в ячейке матрицы, требуется «поднять» это число «вверх» на первую (первые) строки матрицы.

28 У пользователя запрашивается размер матрицы M x N. Программа выводит матрицу и просит пользователя ввести два числа A и B. Программа производит поиск ячеек с совпадением введенных чисел и заменяет значения ячеек A на значение ячеек B.

29 У пользователя запрашивается ввести размер матрицы M x N. Программа выводит матрицу и просит пользователя ввести число Z. Программа производит замену в ячейках с значением больше Z на значение в ячейке минус Z.

30 У пользователя запрашивается ввести размер матрицы M x N. Массив заполняется значениями от 0 до 9, у пользователя запрашивается число Z. Программа заменяет значения в ячейках с одновременно двумя четными или одновременно двумя нечетными индексами на значение плюс Z, в ячейках с индексами один четный, другой нечетный на значение минус Z. В случае получения значений ячейки после замены больше 9, значение уменьшается на 10.

31 У пользователя запрашивается ввести размер матрицы M x N. Программа выводит матрицу и запрашивает у пользователя число Z. В матрице производится циклический сдвиг значений вправо по строкам и вниз по столбцам на величину Z.

32 У пользователя запрашивается ввести размер матрицы M x N. Программа выводит матрицу и запрашивает у пользователя число Z. Программа производит поиск по матрице Z штук идущих подряд одинаковых значений по строкам и столбцам и производит выделение этих мест цветом.

33 У пользователя запрашивается ввести размер матрицы M x N. Программа выводит матрицу и запрашивает у пользователя число Z. Программа производит поиск по матрице Z штук идущих подряд одинаковых значений по диагоналям матрицы и производит выделение этих мест цветом.

34 В матрице произвести поиск элементов, отвечающих правилу чисел Фибоначчи.

4 Лабораторная работа №2. Методы сортировки

Необходимо составить программу для сортировки массива данных методами: пузырьковой, отбора, вставки, Шелла и быстрой сортировки. Вывести на экран неупорядоченный (один раз) и упорядоченные (для каждого из методов) массивы данных. Составить сравнительную таблицу эффективности методов, в которой необходимо указать число сравнений и перестановок переменных в каждом методе сортировки.

При выполнении лабораторной работы должен использоваться массив в стиле языка C, то есть нельзя использовать контейнеры и алгоритмы библиотеки STL или других библиотек.

Неупорядоченная матрица из N строк и M столбцов задается и заполняется один раз (с клавиатуры, из файла или случайными числами), далее она используется для каждого из методов сортировки.

4.1 Варианты заданий

1 Упорядочить каждую строку матрицы по убыванию
2 Упорядочить каждую четную строку по возрастанию, каждый нечетный столбец по возрастанию абсолютных величин.
3 Упорядочить каждый столбец матрицы по убыванию абсолютных величин
4 Упорядочить каждую нечетную строку по возрастанию абсолютных величин, каждый четный столбец по возрастанию.
5 Упорядочить каждую строку матрицы по возрастанию абсолютных величин
6 Упорядочить каждую строку матрицы по убыванию суммы значений цифр элементов матрицы.
7 Упорядочить каждый столбец матрицы по возрастанию суммы значений цифр элементов матрицы.
8 Упорядочить каждую строку матрицы по убыванию абсолютных величин.
9 Упорядочить диагональные элементы матрицы по возрастанию.
10 Упорядочить каждый столбец матрицы по возрастанию.
11 Упорядочить все нечетные элементы (значения элементов) строк по возрастанию.
12 Упорядочить все четные элементы (значения элементов) столбцов по убыванию.
13 Упорядочить каждый столбец матрицы по возрастанию абсолютных величин.
14 Упорядочить каждую четную строку по возрастанию, каждый четный столбец по возрастанию.
15 Упорядочить каждую строку матрицы по возрастанию.
16 Упорядочить каждую нечетную строку матрицы по возрастанию суммы значений цифр элементов матрицы.
17 Упорядочить каждый столбец матрицы по убыванию.
18 Упорядочить каждый четный столбец матрицы по убыванию суммы значений цифр элементов матрицы.
19 Упорядочить каждую строку матрицы по возрастанию отрицательных величин.
20 Упорядочить каждую строку по возрастанию, каждый столбец по убыванию.
21 Упорядочить каждую строку матрицы по возрастанию четных чисел.
22 Упорядочить каждый четный столбец по убыванию, каждую строку по убыванию.
23 В представленной матрице производить замену четных чисел по возрастанию по строкам, нечетных чисел по возрастанию по столбцам.
24 Представить шахматную доску. Упорядочить белые клетки по возрастанию по строкам, черные фигуры по убыванию по столбцам.
25 Упорядочить в каждом значении чисел матрицы цифры по возрастанию, затем упорядочить данные в столбцах по убыванию.
26 Упорядочить в каждом значении чисел матрицы цифры по убыванию, затем упорядочить данные в строках по возрастанию.
27 Упорядочить главную диагональ матрицы по возрастанию, данные сверху от главной диагонали упорядочить по убыванию, снизу от главной диагонали по возрастанию.

5 Лабораторная работа №5. Методы хэширования

Необходимо составить программу для поиска по хэшам данных. Хэширование проводить в соответствии с индивидуальными заданиями. В модуле поиска, предусмотреть реализацию обработки случая, при котором хэш-коды различных данных совпадают.

При выполнении лабораторной работы допускается использование контейнеров и алгоритмов библиотеки STL.

5.1 Варианты заданий

1 Дан текст. Произвести хэширование по строкам и поиск всех совпадающих строк.
2 Дан текст. Произвести хэширование по словам и поиск всех совпадающих слов.

3 Дан текст. Произвести хэширование по блокам, содержащих в себе 10 элементов (символов).
4 Дан текст. Произвести хэширование по словам, поиск проводить двух слов подряд по двум хэшам подряд.
5 Дан текст. Хэш-код представляет из себя два символа, первый из которых – символ с максимальным числом вхождения в слове, второй – число вхождений. Требуется произвести поиск введенного слова в тексте в соответствии с правилом хеширования.
6 Дана таблица текстовой базы данных с полями фиксированной ширины. Произвести хэширование по двум полям и поиск в этих полях.
7 Дана таблица текстовой базы данных с полями фиксированной ширины. Причем имеются как текстовые, так и числовые поля. Произвести хэширование по двум полям и поиск в этих полях.
8 Дана таблица текстовой базы данных записями: фамилия; имя; отчество. Произвести хэширование отдельно каждого поля и поиск по запросу «Фамилия Имя Отчество».
9 Дана таблица текстовой базы данных записями: фамилия; имя; отчество. Произвести хэширование вместе трех полей и поиск по запросу «Фамилия Имя Отчество»
10 Дана таблица текстовой базы данных записями: фамилия; имя; отчество. Произвести хэширование вместе трех полей и поиск по запросу «Фамилия Имя Отчество» и «Фамилия И.О.»
11 Дана таблица текстовой базы данных записями: фамилия; имя; отчество; телефон. Произвести хэширование и поиск по номеру телефона.
12 Дана таблица текстовой базы данных записями: фамилия; имя; отчество; адрес (улица, дом). Произвести хэширование и поиск по адресу (отдельно по названию улицы и по названию улицы и номеру дома).
13 Требуется создать систему авторизации с помощью хранения хэш-кодов. Создать программу, которая позволит добавлять нового пользователя и пароль, проверять соответствие имени пользователя паролю, проверять совпадение добавляемого имени с существующим в базе.

6 Лабораторная работа №6. Использование библиотек динамической компоновки

Написать программу, в которой для вычисления функции, используется динамически подключенная библиотека dll. Вычисление ряда проводить до условия минимизации значения разности двух соседних членов ряда меньше заданного эпсилон. Вычислить значения невязки значений рядной и стандартной Windows (`math.h`) функций $\delta = \sqrt{\left|MyFunc(x)^2 - Function(x)^2\right|}$, проанализировать динамику изменения значения невязки в зависимости от количества слагаемых в ряде. Для остановки счета рядов необходимо использовать следующее условие: $\left|F(x_n) - F(x_{n-1})\right| < eps$.

В работе должны вводиться следующие переменные: точность ε, границы промежутка вычисления x_{start}, x_{end} и шаг Δx, значение x_{ideal} для точного расчета.

В результате работы на экране пользователя должно быть выведена таблица с вычислением значения функции в интервале $x_{start} - x_{end}$ для заданной точности ε:

x	MyFunction(x)	Function(x)	δ
x_{start}			
…			
x_{end}			

Имя функции в таблице (Function) должно передаваться из библиотеки и должно изменяться при замене файла dll на файл другого варианта.

Далее пользователю должно быть предложено ввести значение $x_{ideal} = \ldots$ и должна быть выведена таблица с вычислением значения функции для заданного x_{ideal} для различных значений точности ε (в диапазоне 10^{-1} –10^{-7}):

ε	MyFunction(x)	Function(x)	δ
0.1			
…			
0.0000001			

Для корректной совместной работы приложения и dll, в проекте библиотеки необходимо использовать следующий формат функций:

```
double myf_группа(double /*1 параметр*/,
                  double /*2 параметр*/,
                  double /*точность*/)
```

`double myf_группа` возвращает значение собственной функции. В теле функции `myf_группа` должен быть вызов внутренней функции `фамилия_func`, в которой будет производиться вычисление «рядной» функции.

Для возврата значения математической функции необходимо использовать формат:

```
double myf_math(double /*1 параметр*/, double /*2 параметр*/)
```

Для возврата строки названия функции необходимо использовать формат:

```
char * FName();
```

6.1 Варианты заданий

1 Вычисление $\ln(x) = \sum_{n=1}^{\infty} (-1)^{n+1} \frac{(x-1)^n}{n}$;

2 Число Пи $\frac{\pi}{4} + x = x + \sum_{n=0}^{\infty} (-1)^n \frac{1}{2n+1}$;

3 Число e, $e + x = x + \sum_{n=0}^{\infty} \frac{1}{n!}$;

4 Вычисление sin(x) $\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!}$;

5 Вычисление $a^x = \sum_{k=0}^{\infty} \frac{(x \ln a)^k}{k!}$;

6 Вычисление cos(x) $\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots$;

7 Вычисление exp(x) $e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}$;

8 Вычисление $\cos^3(x)$ $\cos^3(x) = \frac{1}{4} \sum_{n=0}^{\infty} (-1)^n \frac{(3^{2n} + 3) x^{2n}}{(2n)!}$;

9 Вычисление $\sin^2(x)$ $\sin^2(x) = \sum_{n=1}^{\infty} (-1)^{n+1} \frac{2^{2n-1} x^{2n}}{(2n)!}$;

10 Вычисление $\cos^2(x)$ $\cos^2(x) = 1 - \sum_{n=1}^{\infty} (-1)^{n+1} \frac{2^{2n-1} x^{2n}}{(2n)!}$;

11 Вычисление $\sqrt{1+x} = 1 + \frac{1}{2}x - \frac{1}{2 \cdot 4}x^2 + \frac{1}{2 \cdot 4 \cdot 6}x^3 - \frac{1}{2 \cdot 4 \cdot 6 \cdot 8}x^4$;

12 Вычисление $\sqrt{1-x} = 1 - \frac{1}{2}x - \frac{1}{2 \cdot 4}x^2 - \frac{1}{2 \cdot 4 \cdot 6}x^3 - \frac{1}{2 \cdot 4 \cdot 6 \cdot 8}x^4$;

13 Вычисление $\frac{1}{\sqrt{1+x}} = 1 - \frac{1}{2}x + \frac{1 \cdot 3}{2 \cdot 4}x^2 - \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6}x^3 + \frac{1 \cdot 3 \cdot 5 \cdot 7}{2 \cdot 4 \cdot 6 \cdot 8}x^4$;

14 Вычисление $\frac{1}{\sqrt{1-x}} = 1 + \frac{1}{2}x + \frac{1 \cdot 3}{2 \cdot 4}x^2 + \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6}x^3 + \frac{1 \cdot 3 \cdot 5 \cdot 7}{2 \cdot 4 \cdot 6 \cdot 8}x^4$;

15 Вычисление $\ln \frac{1+x}{1-x} = 2\left(x + \frac{x^3}{3} + \frac{x^5}{5} + \frac{x^7}{7} \right), \ |x| < 1$

16 Вычисление $\frac{x}{(1-x)^2} = \sum_{k=1}^{\infty} k x^k, \ |x^2 < 1|$

17 Вычисление $\frac{x}{1-x} = \sum_{k=1}^{\infty} \frac{x^{2^{k-1}}}{1 - x^{2^k}}, \ |x^2 < 1|$

18 Вычисление $\frac{1}{x-1} = \sum_{k=1}^{\infty} \frac{2^{k-1}}{x^{2^{k-1}} + 1}, \ |x^2 > 1|$

19 Вычисление $e^{-x^2} = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k}}{k!},$

20 Вычисление $e^x (1+x) = \sum_{k=0}^{\infty} \frac{x^k (k+1)}{k!},$

21 Вычисление $\frac{1}{(1+x)^2} = 1 - 2x + 3x^2 - 4x^3 + \ldots = \sum_{k=1}^{\infty} (-1)^{k-1} k x^{k-1}$

22 Вычисление $$tg\frac{\pi x}{2}=\frac{4x}{\pi}-\sum_{k=1}^{\infty}\frac{1}{(2k-1)^2-x^2}$$

23 Вычисление $$ctg\ \pi x=\frac{1}{x\pi}+\frac{2x}{\pi}\sum_{k=1}^{\infty}\frac{1}{x^2-k^2}$$