Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт Информационных технологий и анализа данных  
  
Отделение прикладной математики и информатики

ВВЕДЕНИЕ В ЭТАПЫ РАЗРАБОТКИ ПРОГРАММ.

ОТЧЕТ  
по лабораторной работе № 3  
Вариант №10

по дисциплине «программирование»

Выполнил

Студент группы ЭВМб-23-1 \_\_\_\_\_\_\_\_\_\_\_\_Г. М. Распутин  
  
Принял

Доцент \_\_\_\_\_\_\_\_\_\_\_\_\_ А. Б. Столбов

Иркутск 2023

# Массивы и пользовательские структуры. Задание А.

## Условие задачи.

*10в:* Дан целый вектор М(n), n<=12. Сформировать вектор А, элементы которого A[i] равны количеству цифр в десятичной записи элемента М[i].

1. Таблица внешних спецификаций всей программы

Таблица 1.1 – Внешние спецификации всей программы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Имя | Назначение | Тип | Вх/Вых. | Диапазон |
| 1 | M | Исходный массив | Целый массив | Вход | От 1 до 999999999 |
| 2 | N | Количество элементов исходного массива | Целый | Вход | [1..12] |
| 3 | А | Полученный вектор | Целый массив | выход | От 1 до 9 |
| 4 | С1 | неправильный ввод размера | Текст | Выход | “Некорректный ввод. Введите ещё раз” |

## **3. Алгоритмизация**.

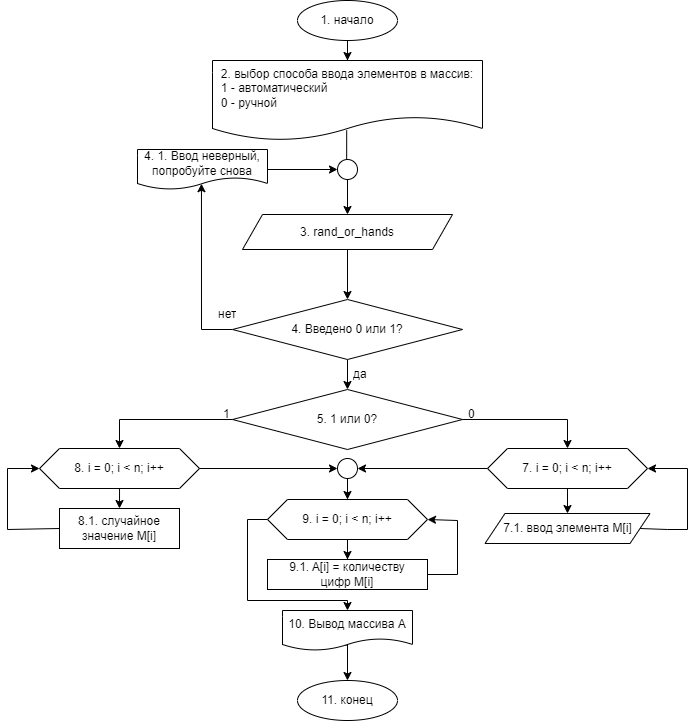
Блок-схема алгоритма представлена на рис. 1.1

Рисунок 1.1 – Блок-схема алгоритма

## 

## 4. Проектирование тестов

Таблица 1.2 – Таблица тестов.

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Назначение теста | Входные данные | Выходные данные |
| 1 | Простой тест с ручным вводом трех чисел | N = 3  M{1, 12, 123} | A{1, 2, 3} |
| 2 | Простой тест с ручным вводом восьми чисел | N = 8  M{1, 12, 123, 1234, 12345, 123456, 1234567, 12345678} | A{1, 2, 3, 4, 5, 6, 7, 8} |
| 3 | Простой тест работы автоматического ввода. | N = 4  M{42, 8469, 6335, 6503} | A{2, 4, 4, 4} |

## 5. Проверка правильности алгоритма с помощью таблицы тестов.

Тест 1: блоки 1. 2. 3. 4. 5. 7. 7.1…7. 9. 9.1…9. 10. 11

Тест 2: блоки 1. 2. 3. 4. 5. 7. 7.1…7. 9. 9.1…9. 10. 11

Тест 3: блоки 1. 2. 3. 4. 5. 8. 8.1…7. 9. 9.1…9. 10. 11

## 6. Кодирование алгоритма или запись алгоритма на языке С++ (ссылка на код: <https://replit.com/@GieorghiiRasput/LabProg3A#main.cpp>)

#include <iostream>

#include <sstream>

using namespace std;

long long int intInputFromConsole(string mess, string err, long long int min,

long long int max) {

long long int curNumber = 0;

string line;

cout << mess;

while (getline(cin, line)) {

stringstream ss(line);

if (ss >> curNumber) {

if (ss.eof()) {

if (curNumber >= min && curNumber <= max) {

return curNumber;

}

}

}

cout << err << endl;

}

}

int main() {

setlocale(LC\_ALL, "Russian");

int M[12];

int A[12];

int rand\_or\_hands;

int counterN = 1;

cout << "Как вводить значения?\n 0 – ручной ввод\n 1 – случайные числа\n";

rand\_or\_hands =

intInputFromConsole("\n", "Ввод невреный, попробуйте снова\n", 0, 1);

cout << "Ок!\n\n";

cout << "Сколько будет элементов вектора?(Максимум - 12)\n";

counterN =

intInputFromConsole("\n", "Ввод невреный, попробуйте снова\n", 1, 12);

cout << "Ок!\n\n";

if (rand\_or\_hands == 1) {

for (int i = 0; i < counterN; i++) {

M[i] = rand() % 9999 + 1;

}

}

else if (rand\_or\_hands == 0) {

cout << "Вводите значения элементов массива\n";

for (int i = 0; i < counterN; i++) {

M[i] = intInputFromConsole("\n", "Некорректный ввод. Введите ещё раз.\n",

1, 999999999);

cout << "ок!\n";

}

}

for (int i = 0; i < counterN; i++) {

int numb = M[i];

int count = 0;

while (numb) {

numb /= 10;

count++;

}

A[i] = count;

}

cout << "\nМножество M:\n";

for (int i = 0; i < counterN; i++) {

cout << M[i] << ' ';

}

cout << "\n";

cout << "Множество А:\n";

for (int i = 0; i < counterN; i++) {

cout << A[i] << ' ';

}

cout << "\n";

return 0;

}

# Массивы и пользовательские структуры. Задание Б.

## Условие задачи.

1. На вход программы подается бинарное отношение, заданное перечислением элементов. Реализовать программу, определяющую является ли бинарное отношение отношением эквивалентности. Вывести матрицу отношения..;

## **Таблица внешних спецификаций**

Таблица 2.1 – Внешние спецификации

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Имя | Назначение | Тип | Вх/Вых. | Диапазон |
| 1 | n | Количество элементов в множестве(для построения квадратной матрицы n x n) | Целый | Вход | от 1 до 10 |
| 2 | relation | Исходная матрица бинарных отношений | Целый массив размерности 2 размера n2 | Вход | От 0 до 1 |
| 3 | С1 | Полученная матрица | Целый массив размерности 2 размера n2 | Выход | От 0 до 1 |
| 4 | С2 | Рефлективность | текст | Выход | 2 вида |
| 5 | С3 | Симметричность | текст | Выход | 2 вида |
| 6 | С4 | Транзитивность | текст | Выход | 2 вида |
| 7 | С5 | Эквивалентность | текст | Выход | 2 вида |

C2:

1. бинарное отношение рефлексивно!!!
2. бинарное отношение не рефлексивно.

C3:

1. бинарное отношение симметрично!!!
2. бинарное отношение не симметрично.

C4:

1. бинарное отношение транзитивно!!!
2. бинарное отношение не транзитивно.

C5:

1. бинарное отношение эквивалентно!!!
2. бинарное отношение не эквивалентно.

## **Проектирование тестов.**

Таблица 2.2 – Таблица тестов.

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Назначение теста | Входные данные | Выходные данные |
| 1 | Ввод рефлексивной, не симметричной, транзитивной, не эквивалентной матрицы | 1 0 0  1 1 0  0 0 1 | бинарное отношение рефлексивно!!!  бинарное отношение не симметрично.  бинарное отношение транзитивно!!!  бинарное отношение не эквивалентно. |
| 2 | Ввод не рефлексивной, не симметричной, транзитивной, не эквивалентной матрицы | 1 1 1  0 0 0  1 1 1 | бинарное отношение не рефлексивно.  бинарное отношение не симметрично.  бинарное отношение транзитивно!!!  бинарное отношение не эквивалентно. |
| 3 | Ввод эквивалентной матрицы бинарных отношений | 1 1 1  1 1 1  1 1 1 | бинарное отношение рефлексивно!!!  бинарное отношение симметрично!!!  бинарное отношение транзитивно!!!  бинарное отношение эквивалентно!!! |

## **Кодирование алгоритма или запись алгоритма на языке С++** (ссылка на код: <https://replit.com/@GieorghiiRasput/Labprog3B#main.cpp>)

#include <iostream>

#include <sstream>

using namespace std;

int intInputFromConsole(string mess, string err, int min, int max)

{

int curNumber = 0;

string line;

cout << mess;

while (getline(cin, line))

{

stringstream ss(line);

if (ss >> curNumber)

{

if (ss.eof())

{

if (curNumber >= min && curNumber <= max)

{

return curNumber;

}

}

}

cout << err << "\n";

}

}

void checkerMatrix(int\*\* relation, int n)

{

for (int i = 0; i < n; i++)

{

relation[i] = new int[n];

for (int j = 0; j < n; j++)

{

int a;

cin >> a;

relation[i][j] = a;

}

}

bool checkchek = true; //Переменная для того, чтобы выйти из вложенного цикла

for (int i = 0; i < n; i++)

{

for (int j = 0; j < n; j++)

{

if (relation[i][j] != 1 && relation[i][j] != 0)

{

cout << "Ошибка ввода матрицы, попробуйте снова\n";

cout << "Неверное число: " << relation[i][j] << '\n';

checkerMatrix(relation, n);

checkchek = false;

break;

}

}

if (checkchek == false) break;

}

}

int main() {

setlocale(LC\_ALL, "ru");

int n;

cout << "Элементов в множесетве будет: ";

n = intInputFromConsole("", "erorr", 1, 10);

int\*\* relation = new int\* [n];

cout << "Введите бинарные отношения для матрицы (" << n << "x" << n << "):" << "\n";

checkerMatrix(relation, n);

//for (int i = 0; i < n; i++)

//{

// relation[i] = new int[n];

// for (int j = 0; j < n; j++)

// {

// relation[i][j] = intInputFromConsole("", "Ошибка, введите ещё раз:\n", 0, 1);

// }

//}

//проверка на рефлексивность

bool reflex = true;

for (int i = 0; i < n; i++)

{

if (relation[i][i] != 1)

{

reflex = false;

break;

}

}

//проверка на симметричность

bool symetr = true;

for (int i = 0; i < n; i++)

{

for (int j = 0; i < n; i++)

{

if (relation[i][j] != relation[j][i])

{

symetr = false;

break;

}

}

if (symetr == false) break;

}

//проверка на транзитивность

bool tran = true;

for (int i = 0; i < n; i++)

{

for (int j = 0; j < n; j++)

{

for (int k = 0; k < n; k++)

{

if (relation[i][j] == 1 && relation[j][k] == 1 && relation[i][k] != 1) {

tran = false;

break;

}

}

if (tran == false) break;

}

if (tran == false) break;

}

cout << "Матрица бинарных отношений: \n";

for (int i = 0; i < n; i++)

{

for (int j = 0; j < n; j++)

{

cout << relation[i][j] << " ";

}

cout << "\n";

}

cout << "\n\n\n";

cout << (reflex ? "бинарное отношение рефлексивно!!!\n" : "бинарное отношение не рефлексивно.\n");

cout << (symetr ? "бинарное отношение симметрично!!!\n" : "бинарное отношение не симметрично.\n");

cout << (tran ? "бинарное отношение транзитивно!!!\n" : "бинарное отношение не транзитивно.\n");

cout << ((tran && symetr && reflex) ? "бинарное отношение эквивалентно!!!\n" : "бинарное отношение не эквивалентно.\n");

for (int i = 0; i < n; i++)

{

delete[] relation[i];

}

delete[] relation;

return 0;

}

# Массивы и пользовательские структуры. Задание В.

## Условие задачи.

1. Создать файл, содержащий информацию об игроках футбольной команды. Структура информации: номер игрока, фамилия, имя, название клубов, за которые играл и в какие сезоны, дата рождения, количество забитых мячей. Определить список игроков команды, рассортированный по дате рождения. Ввести фамилию игрока и определить игроков этой команды, которые играли, хотя бы один сезон, против игрока фамилию которого вы ввели. Посчитать количество игроков, которые играли в разные сезоны в одном клубе.

## **Таблица внешних спецификаций.**

Таблица 3.1 – Внешние спецификации

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Имя | Назначение | Тип | Вх/Вых. | Диапазон |
| 1 | one\_Player | Игрок | Football\_chel | Вход | Текстовые значения полей структуры Football\_chel |
| 2 | Player | Массив игроков | Одномерный массив типа Football\_chel | Вход | Football\_chel |

## **Спецификация созданных пользовательских структур.**

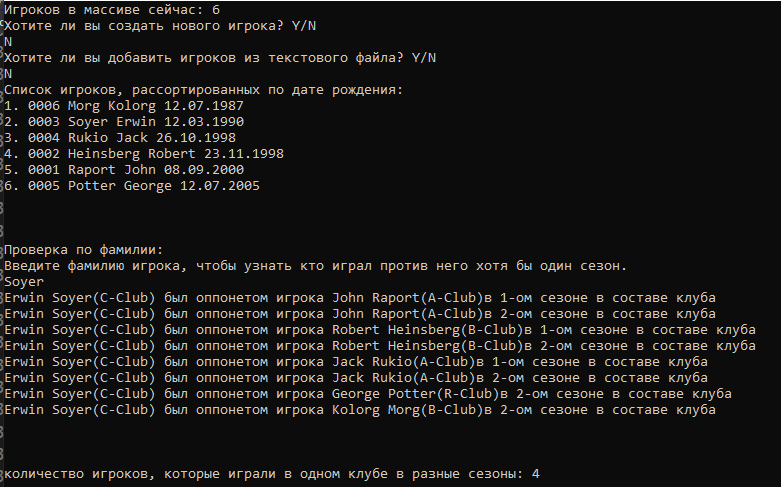
|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Назначение поля | Тип поля | Диапазон |
| number\_of\_player | Номер игрока | Строка | Символы ASCII кодировки |
| last\_name | Фамилия игрока | Строка | Символы ASCII кодировки |
| first\_name | Имя игрока | Строка | Символы ASCII кодировки |
| clubs\_seasons | Список клубов и сезонов, в которых участвовал игрок | Двумерный массив, содержащий данные | Символы ASCII кодировки |
| date\_of\_birth | Дата рождения игрока | Строка | Символы ASCII кодировки в формате “XX.YY.ZZZZ”, где XX – день, YY – месяц, ZZZZ – год рождения игрока |
| killed\_balls | Количество забитых мячей | Целое | От 0 до 999 |

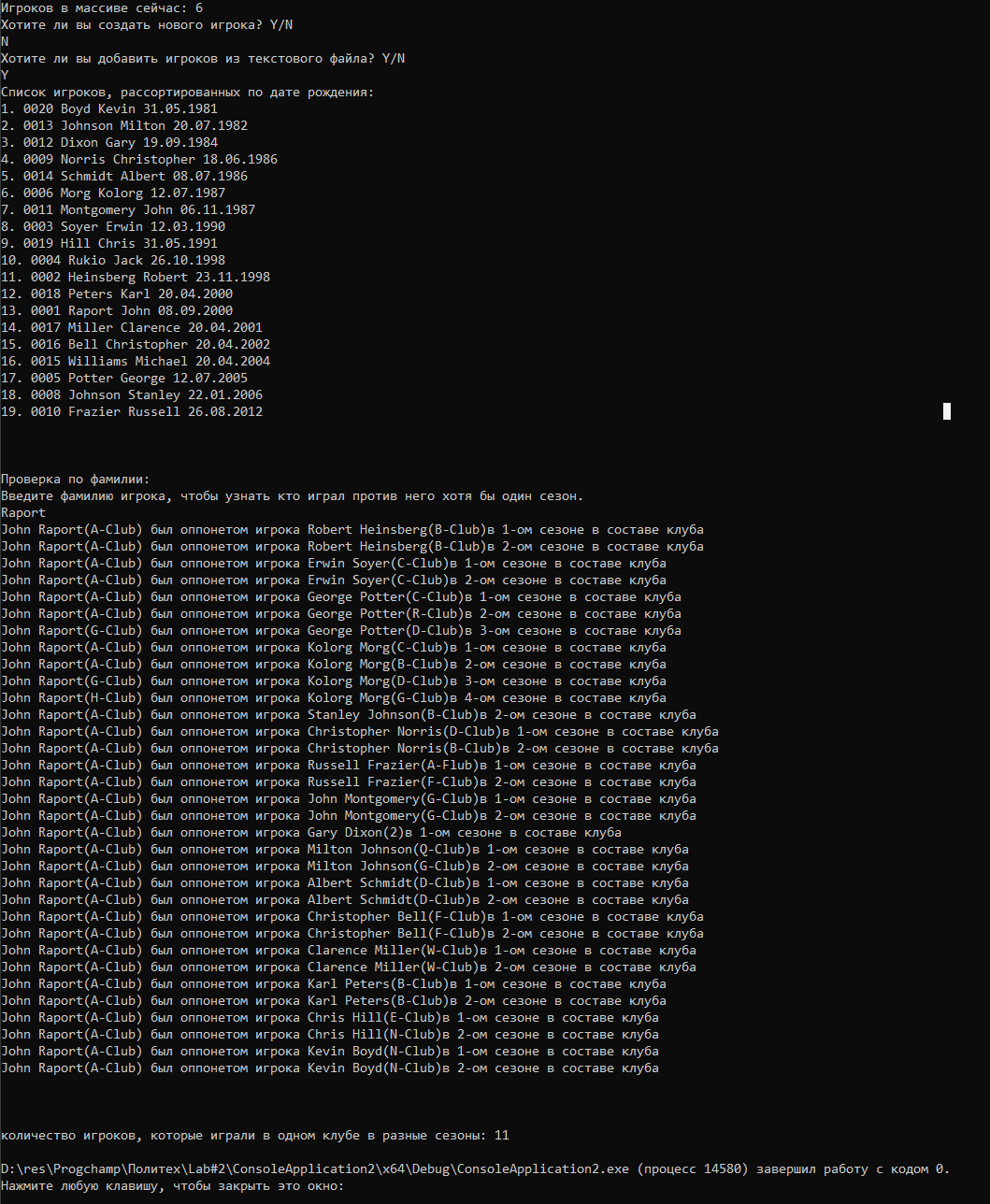
Таблица 3.2 – Спецификация созданных пользовательских структур

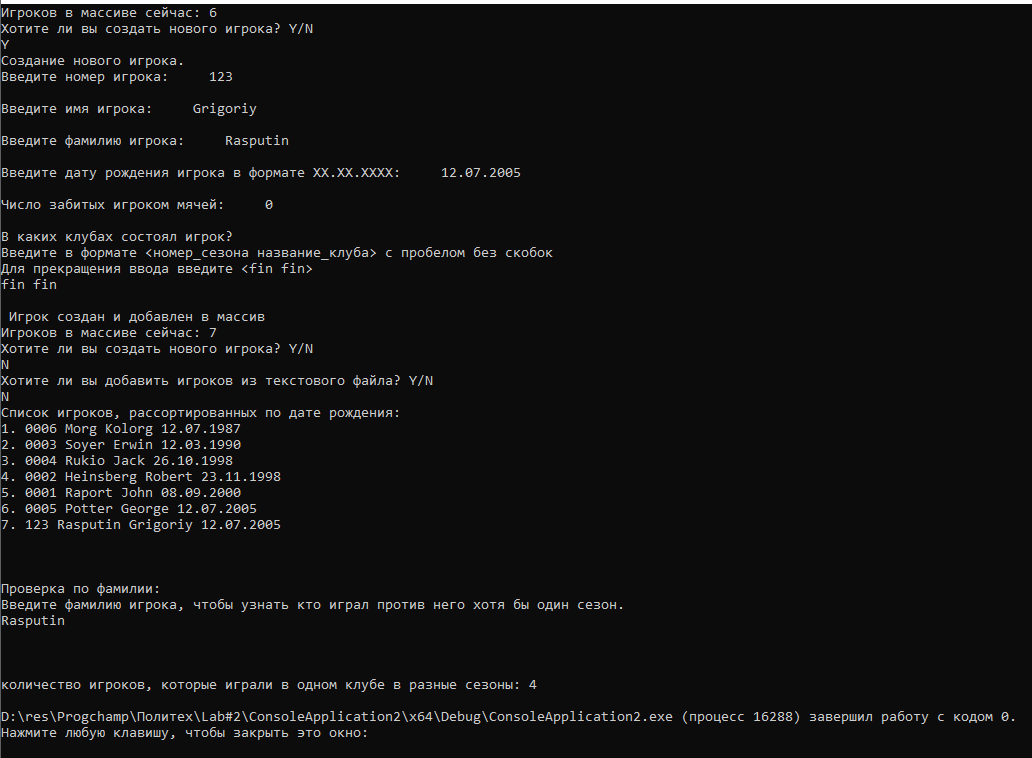
## **Проектирование тестов.**

Таблица 3.3 – Таблица тестов.

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Назначение теста | Входные данные | Выходные данные |
| 1 | Использование всех введённых заранее в код игроков для выполнения данных в задании подзадач | - | С1 |
| 2 | Использование всех введённых заранее в код игроков + игроков, заданных в файле для выполнения данных в задании подзадач | Текстовый файл "players.txt" | C2 |
| 3 | Использование всех введённых заранее в код игроков и создание своего для выполнения данных в задании подзадач | 123 Grigoriy Rasputin 12.07.2005 0 | C3 |

C1:   
  
  
  
  
C2:



C3: 

## **5. Кодирование алгоритма или запись алгоритма на языке С++** (ссылка на код: <https://replit.com/@GieorghiiRasput/Labprog3C#main.cpp>)

//Примечание: В этом коде 401 строка… поэтому я загружу его вместе с отчетом и файлом с игроками.