Министерство образования Республики Беларусь

Учреждение образования

«Белорусский Государственный технологический университет»

Кафедра программной инженерии

Отчет по лабораторной работе № 6

**«Структурное программирование»**

Дисциплина «Основы программной инженерии»

Специальность «Программная инженерия»

**Выполнил(а)**:

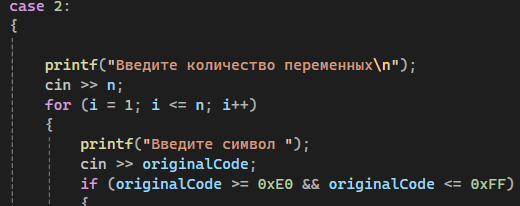
Студентка 1 курса 10 группы 1 подгруппы ФИТ

Загнетова Мария Игоревна

2024

**Задание3:**

**1)**

****

Добавили цикл for

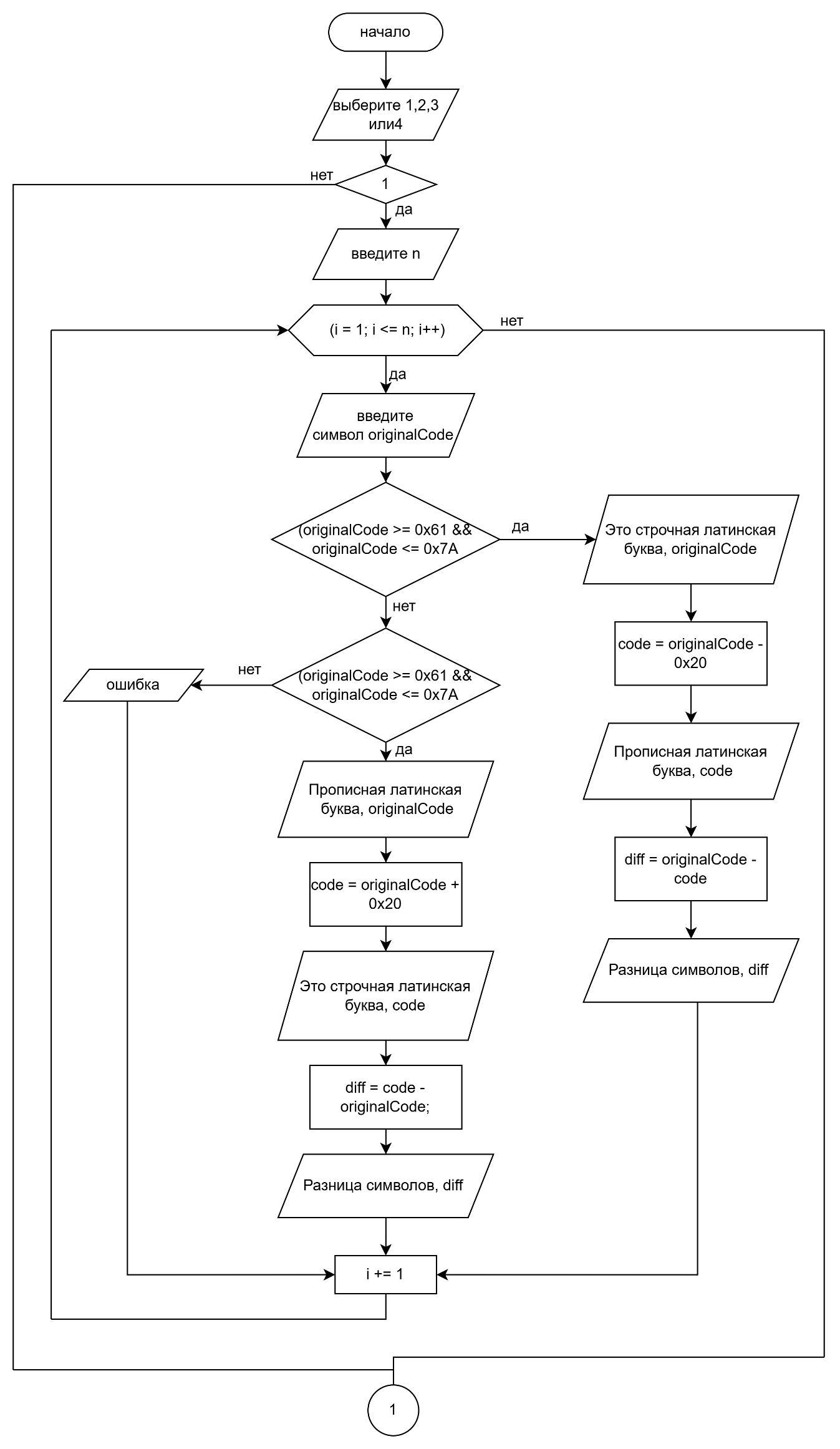
2) Cоздать диалоговую программу, которая в зависимости от выбранного варианта использования выполняет действия пунктов 1, 2, 3 для любого количества введенных с клавиатуры символов.

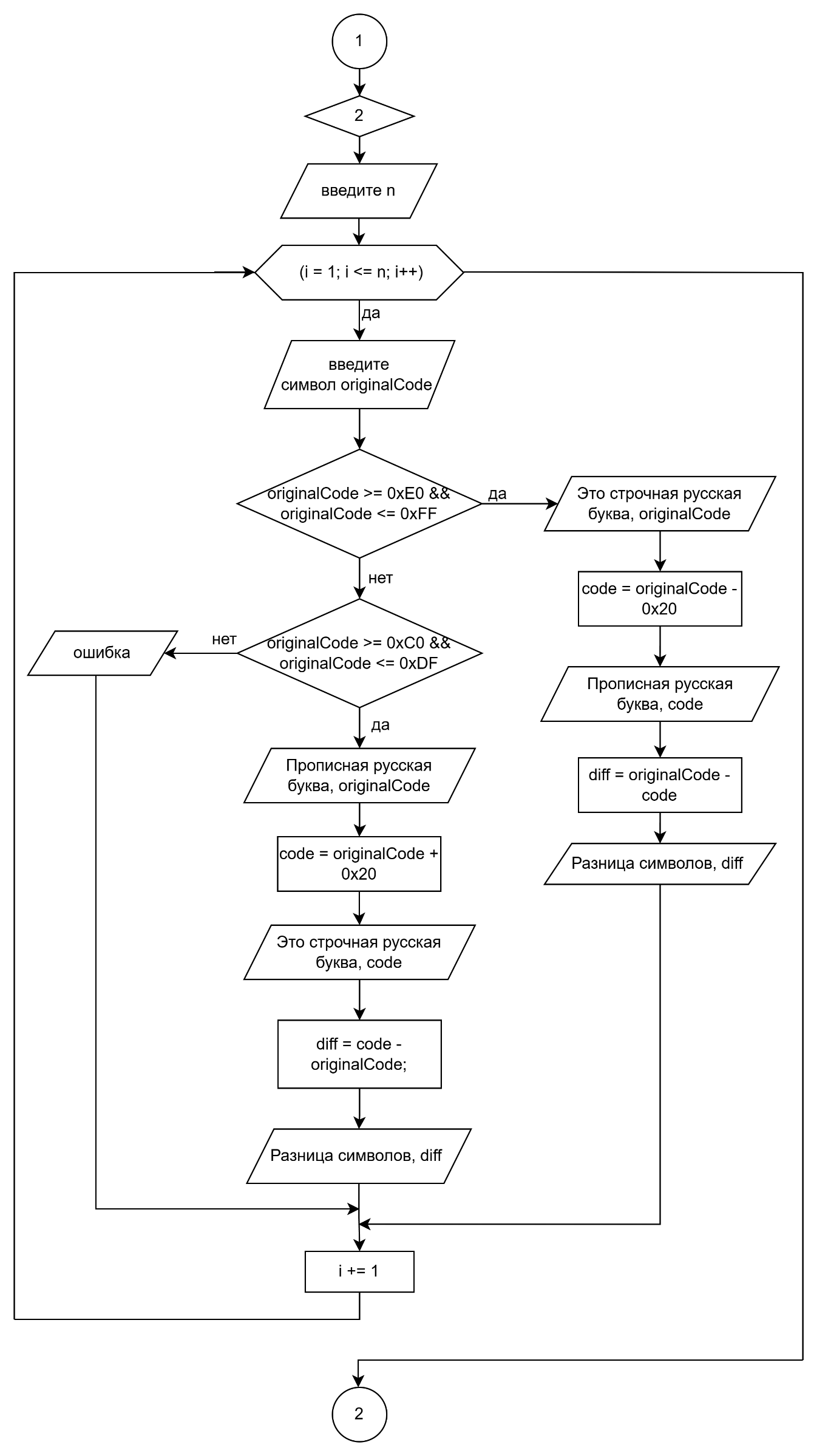
3)

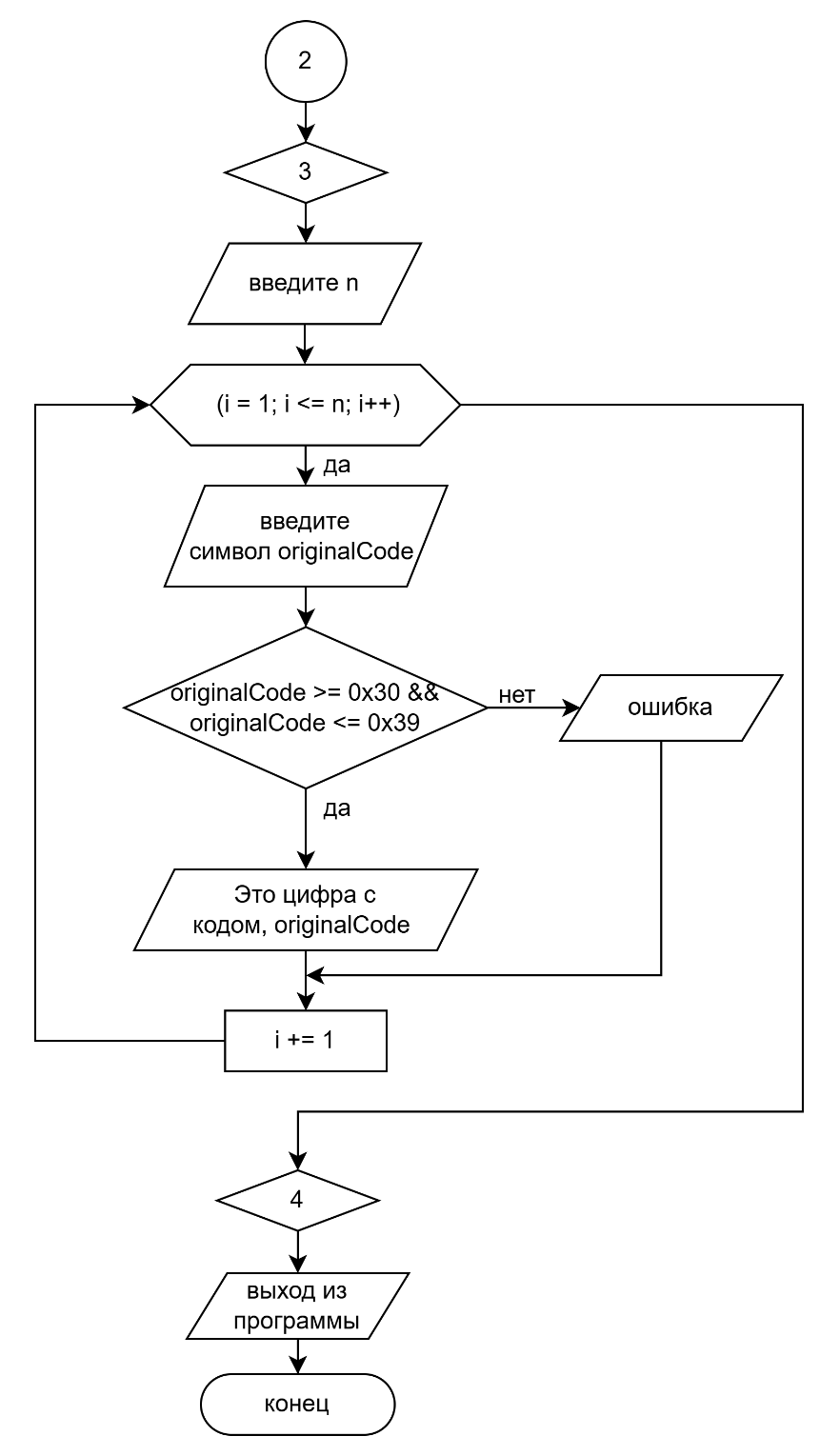
Входные: (номер варианта использования программы, количество символов, символ)

Выходные:(символы букв в строчном и прописном вариантах, код символов в кодировках, варианты использования программы, ошибка)

4)







**5,6)**

**Первый модуль:**

#include <iostream>

#include <Windows.h>

using namespace std;

int latinica()

{

unsigned char code;

unsigned char originalCode;

int diff, n, i;

int k = 0;

printf("Введите количество переменных\n");

cin >> n;

for (i = 1; i <= n; i++)

{

printf("Введите символ ");

cin >> originalCode;

if (originalCode >= 0x61 && originalCode <= 0x7A)

{

printf("Это строчная латинская буква %c , код Windows - 1251 = %X\n", originalCode, originalCode);

code = originalCode - 0x20;

printf("Прописная латинская буква %c , код Windows - 1251 = %X\n", code, code);

diff = originalCode - code;

printf("Разница символов: %X\n", diff);

}

else if (originalCode >= 0x41 && originalCode <= 0x5A)

{

printf("Это прописная латинская буква %c , код Windows - 1251 = %X\n", originalCode, originalCode);

code = originalCode + 0x20;

printf("Строчная латинская буква %c , код Windows - 1251 = %X\n", code, code);

diff = code - originalCode;

printf("Разница символов: %X\n", diff);

}

else

{

printf("Ошибка\n");

}

}

return 0;

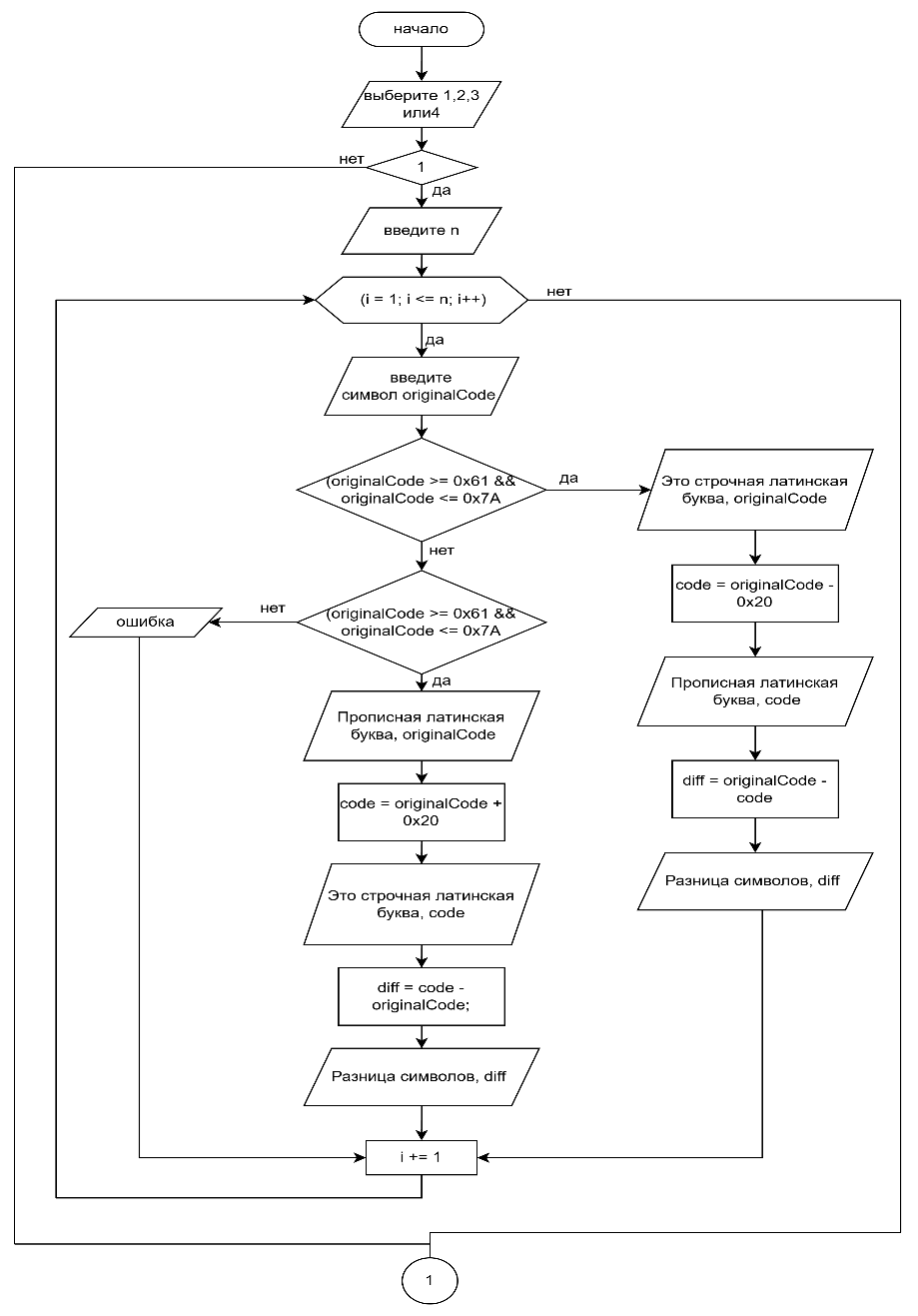
}

**Назначение модуля** – модуль служит для определения разницы кодов введенного символа латинского алфавита строчных и прописных букв в кодировке Windows-1251

**Входные данные** – символ

**Выходные данные** – разница значений кодов или ошибка

**Блок-схема:**



**Второй модуль:**

#include <iostream>

#include <Windows.h>

using namespace std;

int russian()

{

unsigned char code;

unsigned char originalCode;

int diff, n, i;

int k = 0;

printf("Введите количество переменных\n");

cin >> n;

for (i = 1; i <= n; i++)

{

printf("Введите символ ");

cin >> originalCode;

if (originalCode >= 0xE0 && originalCode <= 0xFF)

{

printf("Это строчная русская буква %c , код Windows - 1251 = %X\n", originalCode, originalCode);

code = originalCode - 0x20;

printf("Прописная русская буква %c , код Windows - 1251 = %X\n", code, code);

diff = originalCode - code;

printf("Разница символов: %X\n", diff);

}

else if (originalCode >= 0xC0 && originalCode <= 0xDF)

{

printf("Это прописная русская буква %c , код Windows - 1251 = %X\n", originalCode, originalCode);

code = originalCode + 0x20;

printf("Строчная русская буква %c , код Windows - 1251 = %X\n", code, code);

diff = code - originalCode;

printf("Разница символов: %X\n", diff);

}

else

{

printf("Ошибка\n");

}

}

return 0;

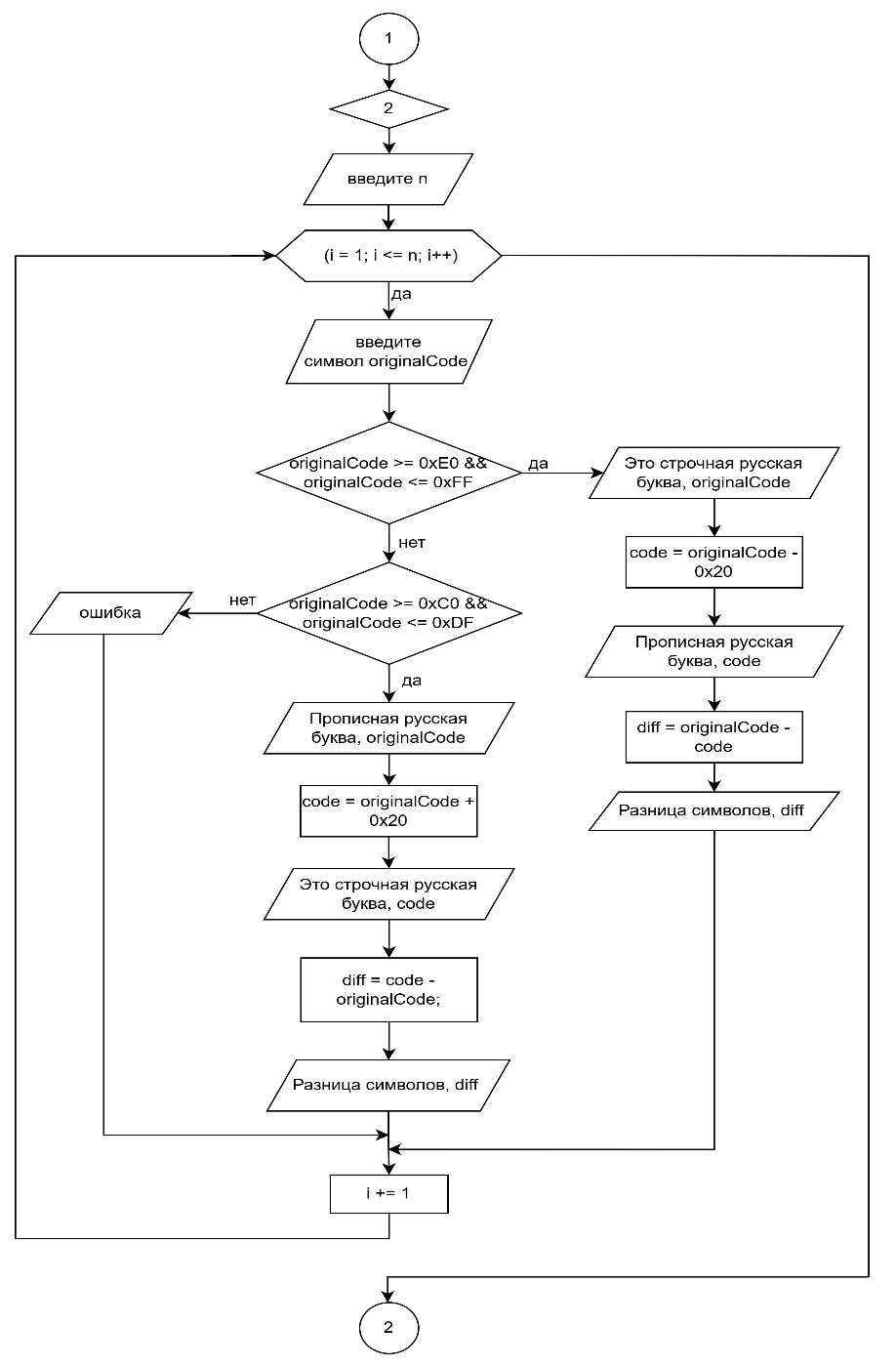
}

**Назначение модуля** – модуль служит для определения разницы кодов введенного символа кириллицы строчных и прописных букв в кодировке Windows-1251

**Входные данные** – символ

**Выходные данные** – разница значений кодов или ошибка

**Блок-схема**:



**Третий модуль:**

#include <iostream>

#include <Windows.h>

using namespace std;

int digits()

{

unsigned char originalCode;

int n, i;

int k = 0;

printf("Введите количество переменных\n");

cin >> n;

for (i = 1; i <= n; i++)

{

printf("Введите цифру: ");

cin >> originalCode;

if (originalCode >= 0x30 && originalCode <= 0x39)

{

printf("Это цифра с кодом: %X\n", originalCode);

}

else {

printf("Ошибка\n");

}

}

return 0;

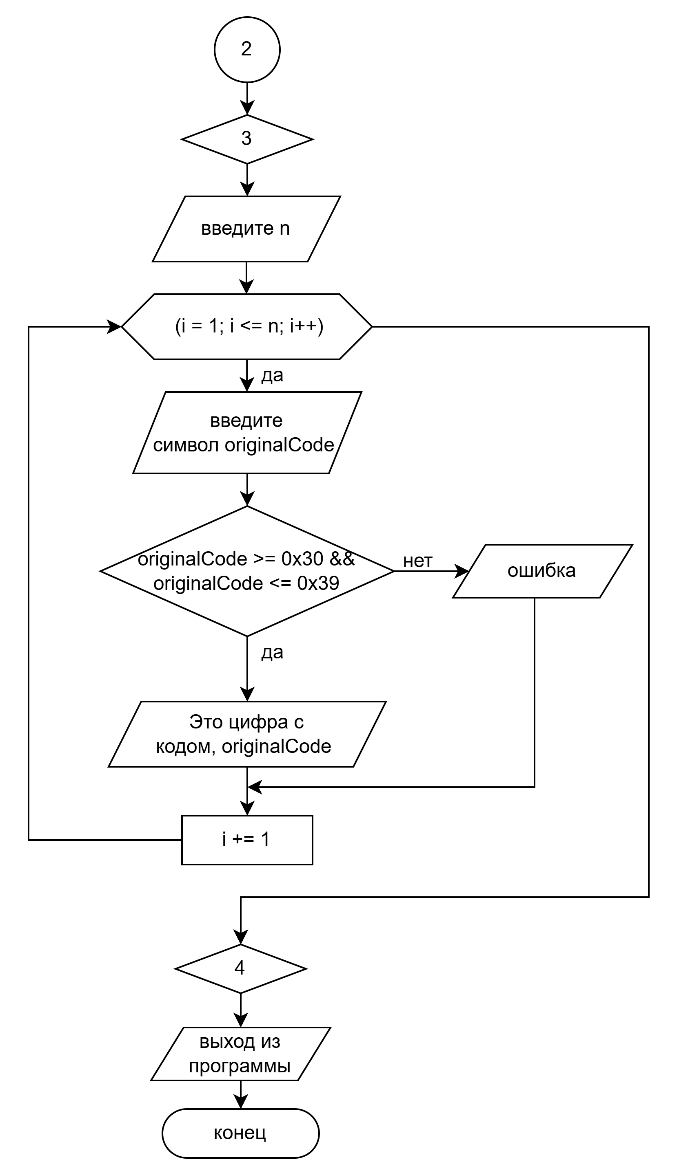
}

**Назначение модуля** – модуль служит для определения кода введенной цифры в кодировке Windows-1251

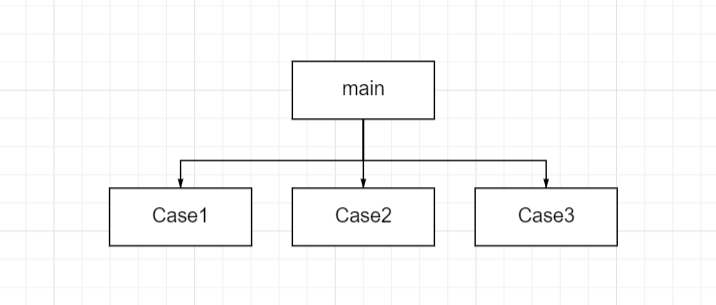
**Входные данные –** символ

**Выходные данные –** код цифры в кодировке Windows-1251 или ошибка

**Блок-схема:**



**7)**



НАЧАЛО

(Модуль 1)

НАЧАЛО ЦИКЛА

ДЛЯ (i = 1; i <= n; i++)

ПОВТОРЯТЬ

ВВОД origcode

ЕСЛИ <символ находится в диапазоне от 0x41 до 0x5A (включительно)> ТО

ВЫВОД Это прописная латинская буква

<код\_строчной\_буквы = код\_прописной\_буквы + 0x20>

ВЫВОД Это прописная латинская буква

<разница = код\_прописной\_буквы – код\_строчной\_буквы>

ВЫВОД разница символов

ИНАЧЕ

ЕСЛИ <символ находится в диапазоне от 0x61 до 0x7A (включительно)> ТО

ВЫВОД Это строчная латинская буква

< код\_прописной\_буквы = код\_строчной\_буквы - 0x20>

ВЫВОД Это латинская буква

<разница = код\_прописной\_буквы – код\_строчной\_буквы>

ВЫВОД разница символов

ИНАЧЕ ВЫВОД <Ошибка>

(Модуль 2)

НАЧАЛО ЦИКЛА

ДЛЯ (i = 1; i <= n; i++)

ПОВТОРЯТЬ

ВВОД origcode

ЕСЛИ <символ находится в диапазоне от 0xC0 до 0xDF (включительно)> ТО

ВЫВОД Это прописная русской буква

<код\_строчной\_буквы = код\_прописной\_буквы + 0x20>

ВЫВОД Это прописная латинская буква

<разница = код\_прописной\_буквы – код\_строчной\_буквы>

ВЫВОД разница символов

ИНАЧЕ

ЕСЛИ <символ находится в диапазоне от 0xЕ0 до 0xFF (включительно)> ТО

ВЫВОД Это строчная русская буква

< код\_прописной\_буквы = код\_строчной\_буквы - 0x20>

ВЫВОД Это латинская буква

<разница = код\_прописной\_буквы – код\_строчной\_буквы>

ВЫВОД разница символов

ИНАЧЕ ВЫВОД <Ошибка>

(Модуль 3)

НАЧАЛО ЦИКЛА

ДЛЯ (i = 1; i <= n; i++)

ПОВТОРЯТЬ

ВВОД origcode

ЕСЛИ <символ находится в диапазоне от 0x30 до 0x39 (включительно)> ТО

ВЫВОД Это цифра с кодом

ИНАЧЕ ВЫВОД <Ошибка>

(Модуль 4)

ВЫВОД Выход из программы

КОНЕЦ

**1. Дать определение парадигмы программирования. Перечислить основные парадигмы программирования.**

Парадигма программирования – это совокупность идей и понятий, определяющих стиль написания компьютерных программ (подход к программированию).

**2. Методология разработки программного обеспечения. Определение, назначение.**

Методология включает в себя модель вычислений для данного стиля. Методология разработки программного обеспечения – совокупность методов, применяемых на различных стадиях жизненного цикла программного обеспечения.

**3. Отличие императивного программирования программирования от декларативного**

В декларативном программировании разработчик описывает желаемый результат, а компьютер самостоятельно определяет последовательность действий, которые нужно выполнить для его достижения. В императивном программировании разработчик указывает конкретные шаги, которые должен выполнить компьютер для достижения результата.

**4. Назовите цели и основные принципы структурного программирования.**

Структурное программирование – методология и технология разработки программных средств, основанная на трёх базовых конструкциях:

− следование;

− ветвление;

− цикл.

**Цели структурного программирования:**

− повысить надежность программ (улучшить структуру программы);

− создание понятной, читаемой программы, которая исполняется, тестируется, конфигурируется, сопровождается и модифицируется без участия автора (создание ПП).

**Принципы разработки:**

− программирование «сверху-вниз» (нисходящее программирование);

− модульное программирование с иерархическим упорядочением связей между модулями/подпрограммами «От общего к частному»

**5. Какие базовые конструкции языка программирования лежат в основе структурного программирования.**

− следование;

− ветвление;

− цикл.

**6. Перечислите методы структурного программирования.**

Типичными методами структурного программирования являются:

* нисходящее проектирование (проектирование сверху вниз);
* модульное (процедурное) программирование;
* структурное кодирование.

**7. Какой принцип лежит в основе модульного проектирования?**

Принципы разработки:

− программирование «сверху-вниз» (нисходящее программирование);

− модульное программирование с иерархическим упорядочением связей между модулями/подпрограммами «От общего к частному»

**8. Что означает функциональная декомпозиция задачи?**

Функциональная декомпозиция задачи – разбиение большой задачи на ряд более мелких, функционально самостоятельных подзадач – модулей.

**9. Назовите основные характеристики модуля.**

− один вход и один выход (на вход программный модуль получает набор исходных данных, выполняет их обработку и возвращает набор выходных данных);

− функциональная завершенность (модуль выполняет набор определенных операций для реализации каждой отдельной функции, достаточных для завершения начатой обработки данных);

− логическая независимость (результат работы данного фрагмента программы не зависит от работы других модулей); − слабые информационные связи с другими программными модулями (обмен информацией между отдельными модулями должен быть минимален);

− размер и сложность программного элемента должна быть в разумных рамках.

**10. Перечислите методы модульного программирования.**

− методы нисходящего проектирования (назначение – декомпозиция большой задачи на меньшие так, чтобы каждую подзадачу можно было рассматривать независимо.);

− методы восходящего проектирования.