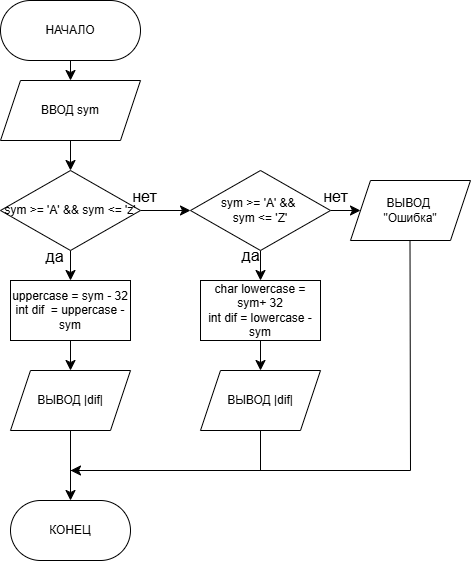
**Зад. 2**

**Постановка задачи:**

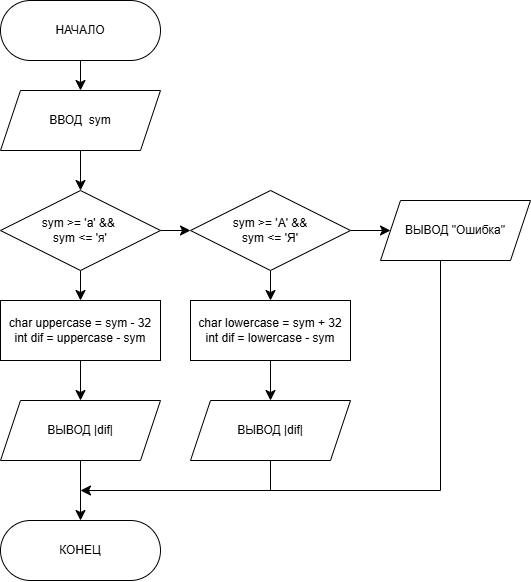
1. Описать логику работы трех модулей
2. Описать логику первого модуля
3. Описать логику второго модуля
4. Описать логику третьего модуля
5. Совместить все функции модулей в файле modules.h
6. Импортировать файл modules.h в основной файл с функцией main
7. Проверить работоспособность программы

**Зад. 4**

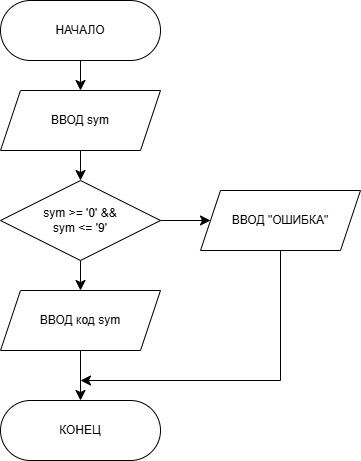
**Блок-схема 1 модуля:**

****

**Блок-схема 2 модуля:**

****

**Блок-схема 3 модуля:**

****

**Зад. 6**

**1)Модуль 1:**

**Состав:** импортирования файла modules.h, функция module1

**Назначение:** отвечает за то, что будет выполнено при n = 1

**Входные/Выходные данные:** char sym/dif

**Алгоритм:** см. выше

**2)Модуль 2:**

**Состав:** импортирования файла modules.h, функция module2

**Назначение:** отвечает за то, что будет выполнено при n = 2

**Входные/Выходные данные:** char sym/dif

**Алгоритм:** см. выше

**3) Модуль 3:**

**Состав:** импортирования файла modules.h, функция module3

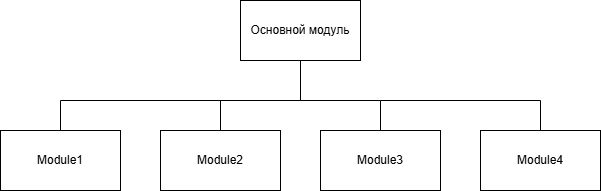
**Назначение:** отвечает за то, что будет выполнено при n = 3

**Входные/Выходные данные:** char sym/ код переменной char

**Алгоритм:** см. выше

**Зад. 7**

**Нисходящее проектирование:**

****

**Зад. 8**

**Модуль 1:**

НАЧАЛО

ВВОД sym

ЕСЛИ <sym >= ‘a’ && sym <= ‘z’> ТО <

ПРИСВОИТЬ uppercase = sym – 32

ПРИСВОИТЬ dif = uppercase – sym

ВЫВОД dif >

ИНАЧЕ <

ВЫВОД «неверный ввод»>

КОНЕЦ

**Модуль 2:**

НАЧАЛО

ВВОД sym

ЕСЛИ <sym >= ‘a’ && sym <= ‘z’> ТО <

ПРИСВОИТЬ uppercase = sym – 32

ПРИСВОИТЬ dif = uppercase – sym

ВЫВОД dif >

ИНАЧЕ <

ЕСЛИ <sym >= ‘А’ && sym <= ‘Я’> ТО <

ПРИСВОИТЬ lowercase = sym + 32

ПРИСВОИТЬ dif = lowercase – sym

ВЫВОД dif> ИНАЧЕ <

ВЫВОД «неверный ввод!»>

>

КОНЕЦ

**Модуль 3:**

НАЧАЛО

ВВОД sym

ЕСЛИ <’9’ >= sym >= ‘0’ > ТО <ВЫВОД код sym> ИНАЧЕ <ВЫВОД «неверный ввод!»>

КОНЕЦ

**Главный модуль:**

НАЧАЛО

ВВОД n

ЕСЛИ <n = 1> ТО <Модуль 1>

ЕСЛИ <n = 2> ТО <Модуль 2>

ЕСЛИ <n = 3> ТО <Модуль 3>

ЕСЛИ <n = 4> ТО КОНЕЦ