Задание. Для задачи из п.5 лабораторной работы 5:

1) Дополнительно предусмотреть возможность ввода с клавиатуры нескольких символов последовательно.

2) Выполнить постановку задачи.

для задания 5 предусмотреть возможность ввода с клавиатуры нескольких символов последовательно.

3) Определить входные, выходные данные.

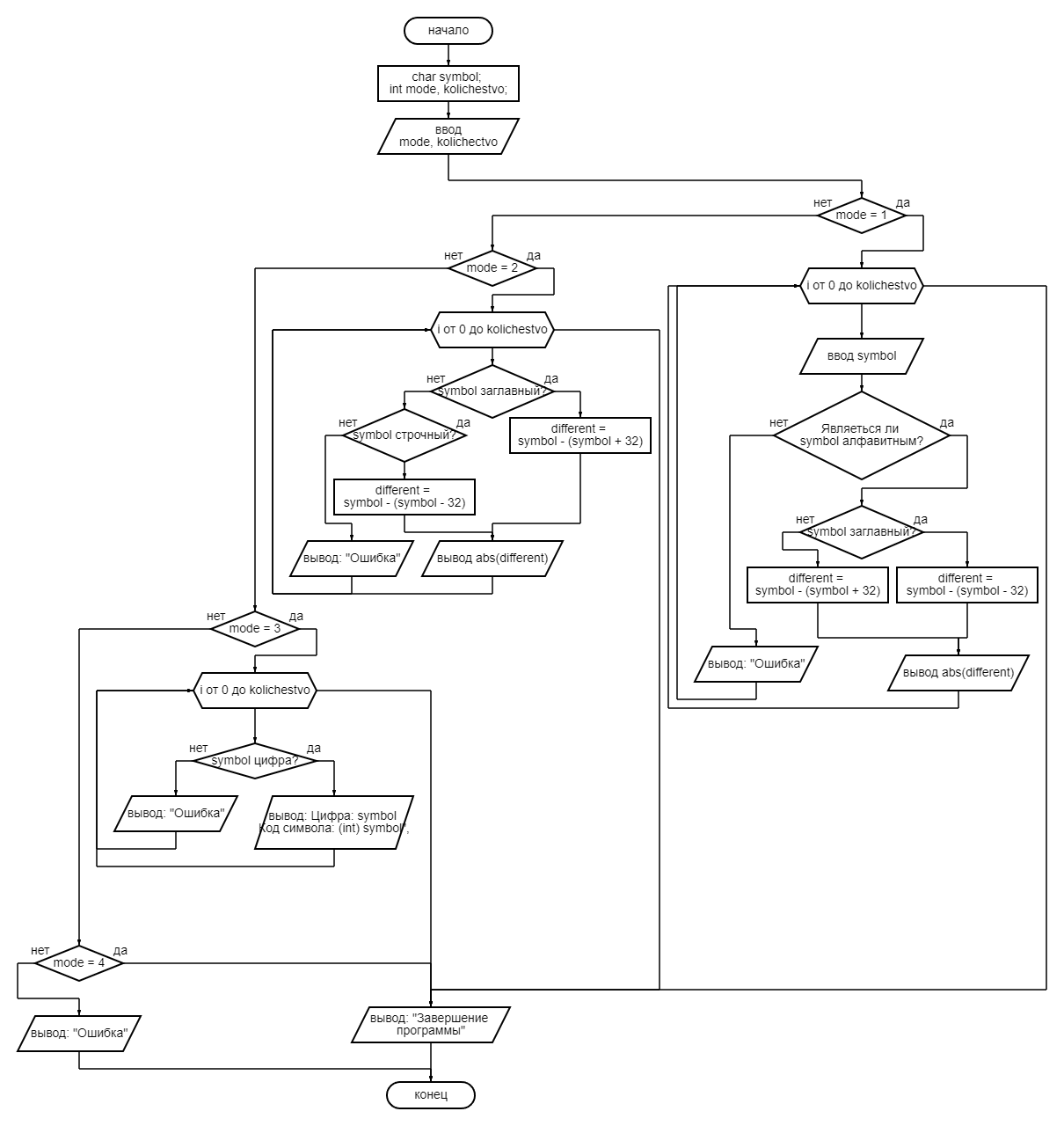
Входные данные:

1. **symbol**: символ, введенный пользователем.
2. **mode**: режим работы с символом, выбранный пользователем.
3. **Kolichestvo**: количество повторений цикла.

Выходные данные:

1. функция **commonCode**, которая выводит сообщение о завершении программы
2. **different**: разницу между строчными и прописными символами в значений кодов в ASCII.
3. **Different**:разницу между строчными и прописными символами в значений кодов в Windows-1251.
4. printf("Цифры и их Коды символов **(symbol, (int)symbol));**

4) Записать алгоритм её решения в виде блок-схемы.



5) Разбить программу на модули. Описать состав, назначение, входные/выходные данные и алгоритм (любым способом) каждого модуля.

В данном случае, программа может быть разделена на следующие модули:

1. **Модуль main**:
   * **Назначение**: Ввод количества символов и режима работы. Запуск обработки символов в соответствии с выбранным режимом.
   * **Входные данные**: Нет.
   * **Выходные данные**: Нет.
   * **Алгоритм**:
     + Ввод количества символов и режима.
     + Запуск функции обработки символов (**processSymbols**) с передачей в нее количества символов и режима.
2. **Модуль processSymbols**:

* **Назначение**: Обработка символов в зависимости от выбранного режима.
* **Входные данные**: Режим работы и количество символов.
* **Выходные данные**: Нет.
* **Алгоритм**:
  + Использование цикла для ввода и обработки каждого символа.
  + Вызов соответствующей функции обработки символа в зависимости от режима.

1. **Модуль processMode1**:

* **Назначение**: Определение разницы значений кодов в ASCII буквы в прописном и строчном написании.
* **Входные данные**: Символ.
* **Выходные данные**: Нет.
* **Алгоритм**:
  + Проверка, является ли символ буквой латинского алфавита.
  + Вывод разницы значений кодов в ASCII.

1. **Модуль processMode2:**

Назначение: Определение разницы значений кодов в Windows-1251 буквы в прописном и строчном написании.

Входные данные: Символ.

Выходные данные: Нет.

Алгоритм:

Проверка, является ли символ буквой русского алфавита. Вывод разницы значений кодов в Windows-1251.

5. **Модуль processMode3**:

* **Назначение**: Вывод в консоль кода символа, соответствующего введенной цифре.
* **Входные данные**: Символ.
* **Выходные данные**: Нет.
* **Алгоритм**:
  + Проверка, является ли символ цифрой.
  + Вывод кода символа.

6. **Модуль commonCode**:

* **Назначение**: Вывод сообщения о завершении программы.
* **Входные данные**: Нет.
* **Выходные данные**: Нет.
* **Алгоритм**:
  + Вывод сообщения о завершении программы.

6) Выполнить нисходящее проектирование программы. Составить модульную схему программы и описать ее, используя псевдокод

Нисходящее проектирование программы подразумевает разбиение программы на модули и описание каждого модуля.

**Модульная схема:**

1. **Модуль main**
   * Ввод режима и количества символов.
   * Вызов функции **processSymbols** с передачей режима и количества символов.
2. **Модуль processSymbols**
   * Повторение для каждого введенного символа:
     + Ввод символа.
     + Вызов функции обработки символа в зависимости от режима.
3. **Модуль processMode1**
   * Проверка, является ли символ буквой латинского алфавита.
   * Если буква в верхнем регистре, преобразование в нижний регистр.
   * Вывод разницы кодов в ASCII.
4. **Модуль processMode2**
   * Проверка, является ли символ буквой русского алфавита.
   * Вывод разницы значений кодов в Windows-1251.
5. **Модуль processMode3**
   * Проверка, является ли символ цифрой.
   * Вывод кода символа.
6. **Модуль commonCode**
   * Вывод сообщения о завершении программы.

НАЧАЛО

ВВОД mode, kolichestvo

ДЛЯ каждого i от 1 до kolichestvo:

ВВОД symbol

ВЫБОР mode:

В случае 1:

processMode1(symbol)

В случае 2:

processMode2(symbol)

В случае 3:

processMode3(symbol)

В случае 4:

commonCode()

По умолчанию:

commonCode()

КОНЕЦ ВЫБОРА

КОНЕЦ ЦИКЛА

КОНЕЦ

Допонительные задания:

− выполнить постановку задачи;

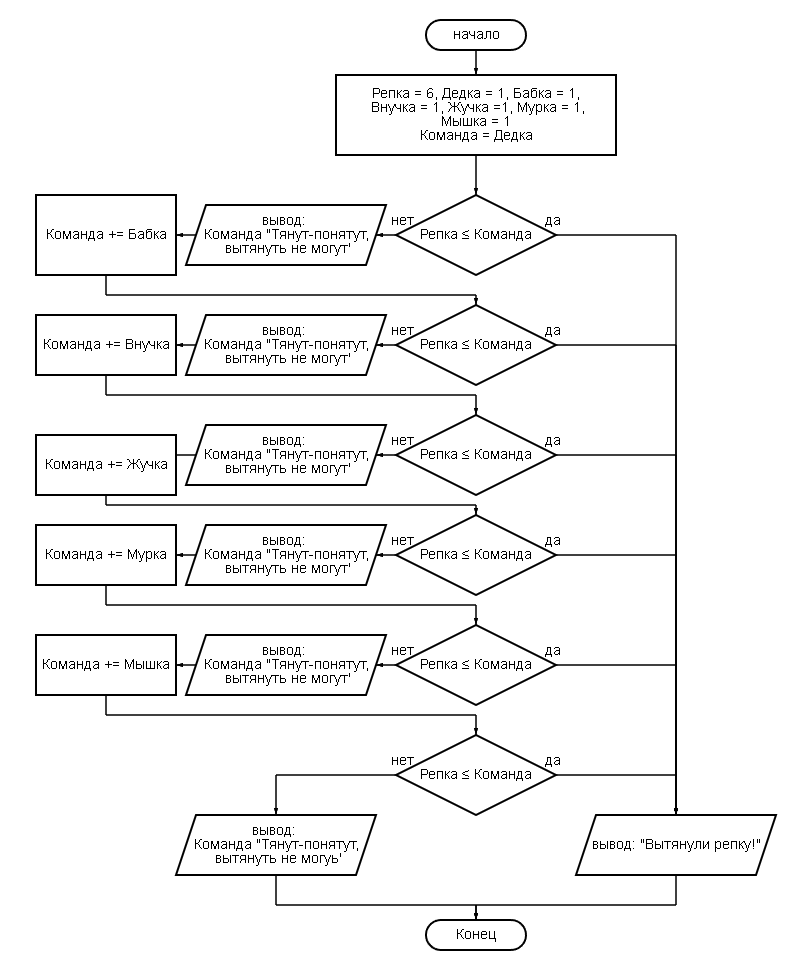
**Постановка задачи:** Реализовать программу на основе сказки "Репка", в которой персонажи поочередно тянут друг друга, чтобы успешно вытянуть репку.

− определить входные, выходные данные;

**Входные данные:** Нет явных входных данных. Программа состоит из последовательности действий персонажей.

**Выходные данные:** Текстовые сообщения, описывающие действия персонажей и успешное вытягивание репки.

− записать алгоритм её решения в виде блок-схемы;



− разбить программу на модули и описать их состав, назначение, входные/выходные данные и привести алгоритм (любым способом) каждого модуля;

1. **Модуль babka**
   * Назначение: Описывает действия бабки.
   * Входные данные: Нет.
   * Выходные данные: Нет.
   * Алгоритм:
     + Вывести сообщение "Бабка тянет Репку."
2. **Модуль dedka**
   * Назначение: Описывает действия дедки.
   * Входные данные: Нет.
   * Выходные данные: Нет.
   * Алгоритм:
     + Вывести сообщение "Дедка тянет бабку."
3. **Модуль vnuchka**
   * Назначение: Описывает действия внучки.
   * Входные данные: Нет.
   * Выходные данные: Нет.
   * Алгоритм:
     + Вывести сообщение "Внучка тянет дедку."
4. **Модуль zhuchka**
   * Назначение: Описывает действия жучки.
   * Входные данные: Нет.
   * Выходные данные: Нет.
   * Алгоритм:
     + Вывести сообщение "Жучка тянет внучку."
5. **Модуль vytyanutRepku**
   * Назначение: Описывает действия по вытягиванию репки.
   * Входные данные: Нет.
   * Выходные данные: Нет.
   * Алгоритм:
     + Вывести сообщение "Вытянули Репку!"

− выполнить нисходящее проектирование программы. Составить модульную схему программы и описать ее, используя псевдокод.

// Модуль babka

Процедура babka()

Вывести "Бабка тянет Репку."

Конец Процедуры

// Модуль dedka

Процедура dedka()

Вывести "Дедка тянет бабку."

Конец Процедуры

// Модуль vnuchka

Процедура vnuchka()

Вывести "Внучка тянет дедку."

Конец Процедуры

// Модуль zhuchka

Процедура zhuchka()

Вывести "Жучка тянет внучку."

Конец Процедуры

// Модуль vytyanutRepku

Процедура vytyanutRepku()

Вывести "Вытянули Репку!"

Конец Процедуры

// Основной модуль main

Процедура main()

Вывести "Начало сказки 'Репка':"

// Вызов модулей

babka()

dedka()

vnuchka()

zhuchka()

vytyanutRepku()

Вывести "Конец сказки 'Репка'."

Конец Процедуры

