Министерство образования Республики Беларусь

Учреждение образования «Белорусский Государственный Технологический Университет»

Факультет информационных технологий

Кафедра программной инженерии

Дисциплина «Основы программной инженерии»

Отчёт к лабораторной работе №6

«Модульное программирование»

Выполнил

Воронович Диана Игоревна

1 курс,6 группа,1 подгруппа

Проверил

Наркевич Аделина

Сергеевна

Минск 2024

**Задание 1**

**Пункт 2**

1. В зависимости от выбранного номера задания выполнить задания 1, 2, 3, или 4 для любого введенного с клавиатуры символа, дополнительно предусмотрев возможность ввода с клавиатуры нескольких символов последовательно, где:

1 – определить разницу значений кодов в ASCII буквы в прописном и строчном написании, если введен символ латинского алфавита, иначе вывод об ошибке;

2 – определить разницу значений кодов в Windows-1251 буквы в прописном и строчном написании, если введен символ русского алфавита, иначе вывод об ошибке;

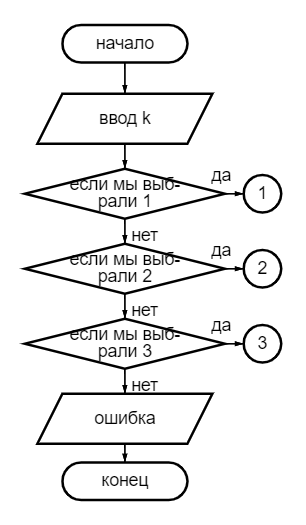
3 – вывести в консоль код символа соответствующего введенной цифре, иначе вывод об ошибке;

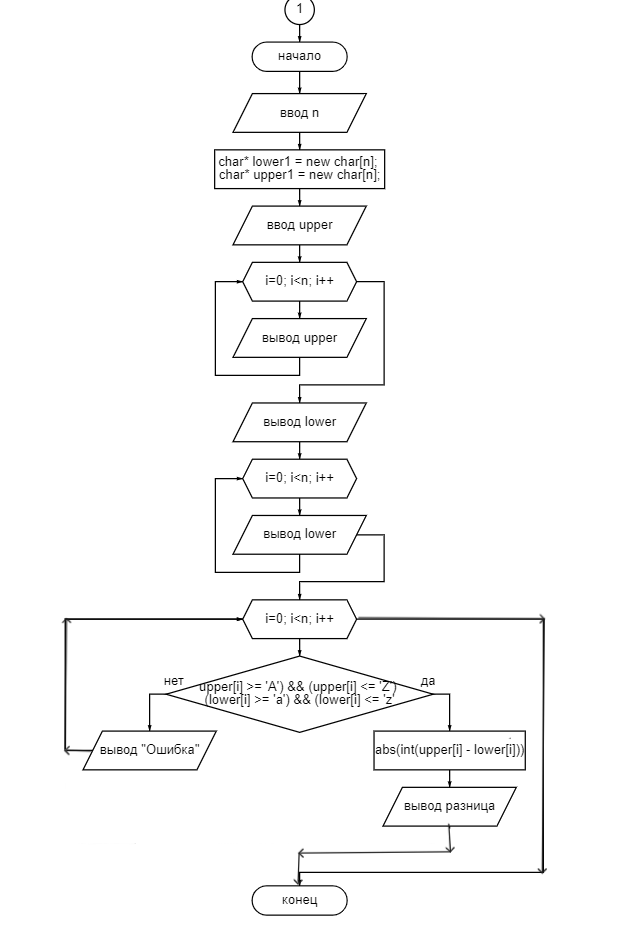
4 – выход из программы.

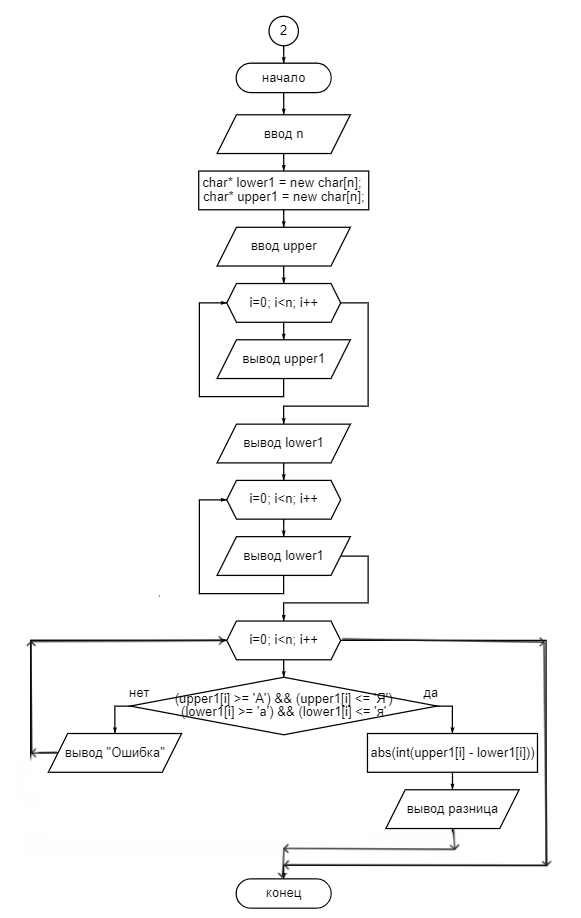
**Пункт 3**

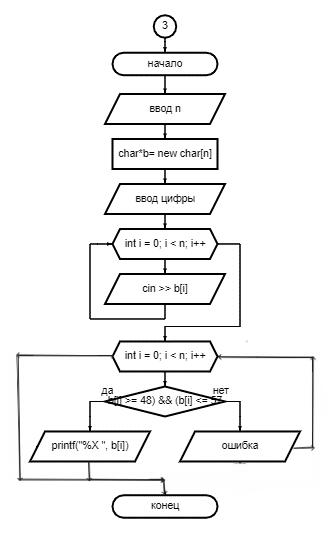
На вход программе задается, в зависимости от выбранного номера задания, символ латинского алфавита, русского алфавита, или цифра, и на выходе программа выводит, в зависимости от выбранного задания, разницу значений кодов в ASCII введенной буквы в прописном и строчном написании, в Windows-1251 введенной буквы в прописном и строчном написании, или код символа введенной цифры.

**Пункт 4**

****

****

****

****

**Пункт 5**

**Состав модуля 1:**

1. Имя модуля (Head).

2. Переменная, которая используется в этом модуле (k), которая будет позволять нам выбирать нужный вариант ответа.

**Назначение модуля:**

1. Вывести на консоль варианты ответов для дальнейшего использования программы.

**Входные данные:**

1. Переменная k для ввода вариантов выбора.

**Выходные данные:**

1. Вывод на консоль предложенных вариантов выбора.

**Псевдокод алгоритма:**

НАЧАЛО

ЗАПРОС k числа у пользователя

ЕСЛИ k=1 ТО использовать модуль case1.h

ЕСЛИ k=2 ТО использовать модуль case2.h

ЕСЛИ k=3 ТО использовать модуль case3.h

ИНАЧЕ ВЫВОД вывести ошибку

КОНЕЦ

**Состав модуля 2:**

1. Имя модуля (Task1). Он состоит из условия, которое проверяет принадлежит ли введенный символ к латинскому алфавиту и, если да, то выводит разницу значений кодов буквы в строчном и прописном написании, иначе выводит сообщение об ошибке.

**Назначение модуля:**

1. Вывести на экран разницу значений кодов или сообщить об ошибке.

**Входные данные:**

1. Переменная для ввода количества элементов массива с клавиатуры.

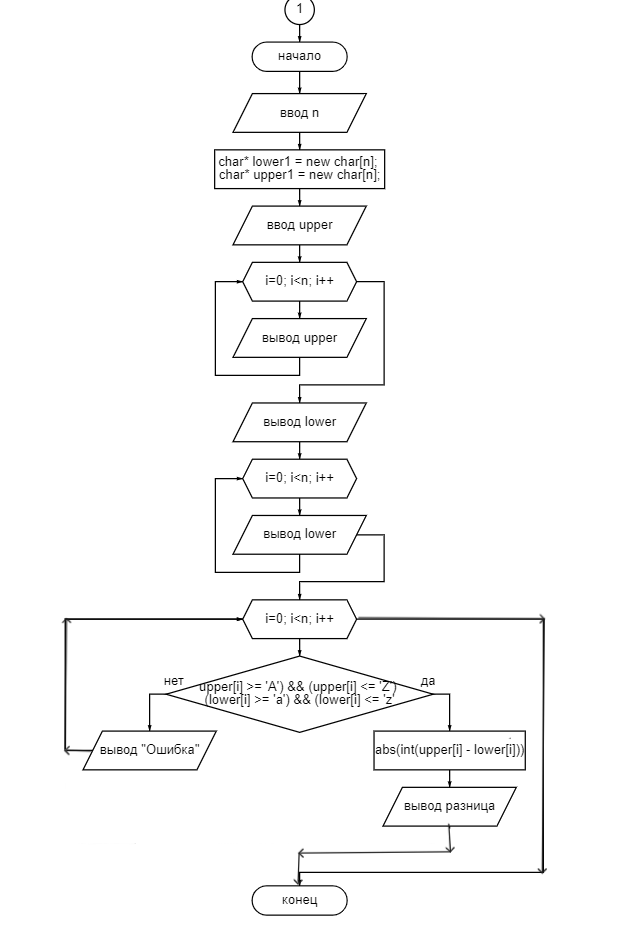
2. Ввод букв латинского алфавита верхнего регистра, используя ТОЛЬКО пробел.

3. Ввод букв латинского алфавита нижнего регистра, используя ТОЛЬКО пробел.

**Выходные данные:**

1. Вывести разницу значений кодов или ошибку.

**Блок-схема алгоритма:**

****

**Состав модуля 3:**

1. Имя модуля (Task2). Он состоит из условия, которое проверяет принадлежит ли введенный символ к русскому алфавиту и, если да, то выводит разницу значений кодов буквы в строчном и прописном написании, иначе выводит сообщение об ошибке.

**Назначение модуля:**

1. Вывести на экран разницу значений кодов или сообщить об ошибке.

**Входные данные:**

1. Переменная для ввода количества элементов массива с клавиатуры.

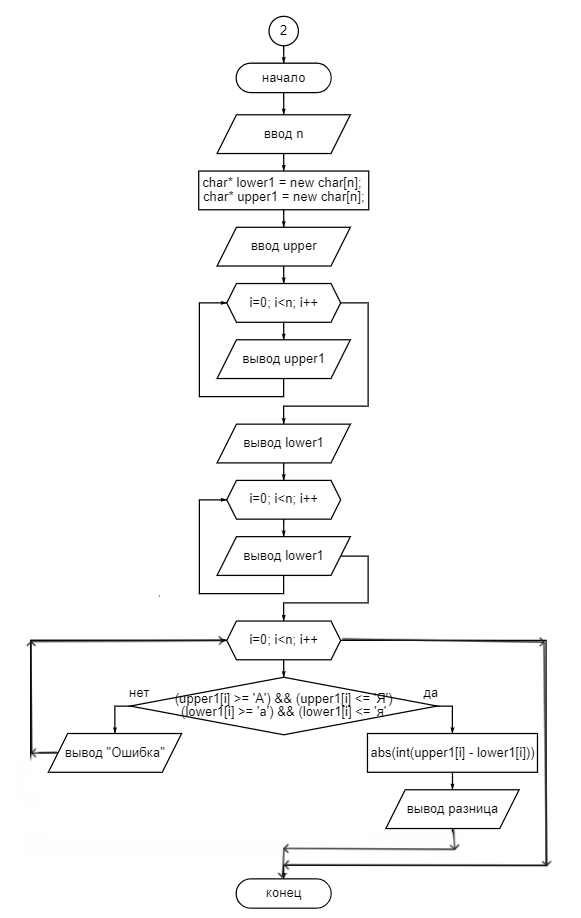
2. Ввод букв русского алфавита верхнего регистра, используя ТОЛЬКО пробел.

3. Ввод букв русского алфавита нижнего регистра, используя ТОЛЬКО пробел.

**Выходные данные:**

1. Вывести разницу значений кодов или ошибку.

**Блок-схема алгоритма:**

****

**Состав модуля 4:**

1. Имя модуля (Task3). Он состоит из условия, которое проверяет является ли введенный символ цифрой, иначе выводит сообщение об ошибке.

**Назначение модуля:**

1. Вывести на экран код цифры в кодировке Windows-1251 или сообщить об ошибке.

**Входные данные:**

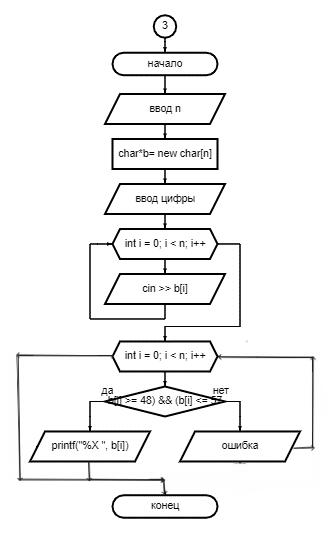
1. Переменная для ввода количества элементов массива с клавиатуры.

2. Ввод цифры.

**Выходные данные:**

1. Вывести какой код у цифры, которую мы ввели, или сообщить об ошибке.

**Блок-схема алгоритма:**

****

**Пункт 6**

****

**Псевдокод:**

**Program** диалоговая программа;

1.Вывести разницу кодов регистров латинского алфавита;

2.Вывести разницу кодов регистров русского алфавита;

3.Вывести код введённой цифры;

4.Выход из программы;

**While** выбрать 1

**Do**

**Begin**

Ввести количество латинских букв, разницу регистров которых мы будем определять, с клавиатуры;

Ввести n-е количество латинских букв верхнего регистра;

Ввести n-е количество латинских букв нижнего регистра;

**If**

Если буквы являются буквами латинского алфавита, то вывести разницу регистров;

**Else**

Вывести ошибку;

**End**

**While** выбрать 2

**Do**

**Begin**

Ввести количество русских букв, разницу регистров которых мы будем определять, с клавиатуры;

Ввести n-е количество русских букв верхнего регистра;

Ввести n-е количество русских букв нижнего регистра;

**If**

Если буквы являются буквами русского алфавита, то вывести разницу регистров;

**Else**

Вывести ошибку;

**End**

**While** выбрать 3

**Do**

**Begin**

Ввести количество цифр, код которых мы будем узнавать, с клавиатуры;

Ввести n-е количество цифр

**If**

Если символы являются цифрами, то вывести код цифры на экран;

**Else**

Вывести ошибку;

**End**

**While** выбрать 4

**Do**

**Begin**

Выход из программы;

**End**

**End program**

**Ответы на вопросы**

1. Парадигма программирования – это совокупность идей и понятий, определяющих стиль написания компьютерных программ (подход к программированию). Структурное программирование, Объектно-ориентированное программирование, Функциональное программирование, Логическое программирование, Визуальное программирование.
2. Методология включает в себя модель вычислений для данного стиля. Методология разработки программного обеспечения – совокупность методов, применяемых на различных стадиях жизненного цикла программного обеспечения.
3. Императивное программирование предполагает, что программа явно описывает алгоритм решения конкретной задачи (действия исполнителя), т.е. описывает как решать поставленную задачу. Декларативное программирование – это предварительная реализация «решателя» для целого класса задач. Тогда для решения конкретной задачи этого класса достаточно декларировать в терминах данного языка только её условие: (исходные данные + необходимый вид результата) «Решатель» сам выполняет процесс получения результата, реализуя известный ему алгоритм решения.
4. 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).
5. − следование; − ветвление; − цикл.
6. нисходящее проектирование (проектирование сверху вниз). модульное (процедурное) программирование. структурное кодирование
7. Модульное программирование – это организация программы как совокупности небольших независимых блоков, называемых модулями. Модуль – функционально законченный фрагмент программы, оформленный в виде отдельного файла с исходным кодом.
8. Функциональная декомпозиция задачи – разбиение большой задачи на ряд более мелких, функционально самостоятельных подзадач – модулей.
9. Модуль – это фрагмент описания процесса, оформленный как самостоятельный программный продукт, пригодный для многократного использования. Модуль, основные характеристики: − 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. Метод восходящей разработки,метод нисходящей разработки.