Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Лабораторная работа 6

По дисциплине «Основы программной инженерии»

На тему «Парадигмы программирования». «Модульное программирование».

Выполнил:

Студент 1 курса 9 группы

Филипчик Иван Владимирович

Преподаватель: Ольга Александровна

2023, Минск

Задание 3

Постановка: Написать диалоговую программу, которая в зависимости от выбранного варианта использования выполняет действия пунктов 1, 2, 3 для любого введенного с клавиатуры символа

1 – определение разницы значений кодов в ASCII буквы в прописном и строчном написании, если введена строка из символов латинского алфавита, иначе вывод сообщения об ошибке;

2 – определение разницы значений кодов в Windows-1251 буквы в прописном и строчном написании, если введена строка из символов русского алфавита, иначе вывод сообщения об ошибке;

3 – вывод в консоль кода символа, соответствующего введенной строке из цифр, иначе вывод сообщения об ошибке;

4 – выход из программы.

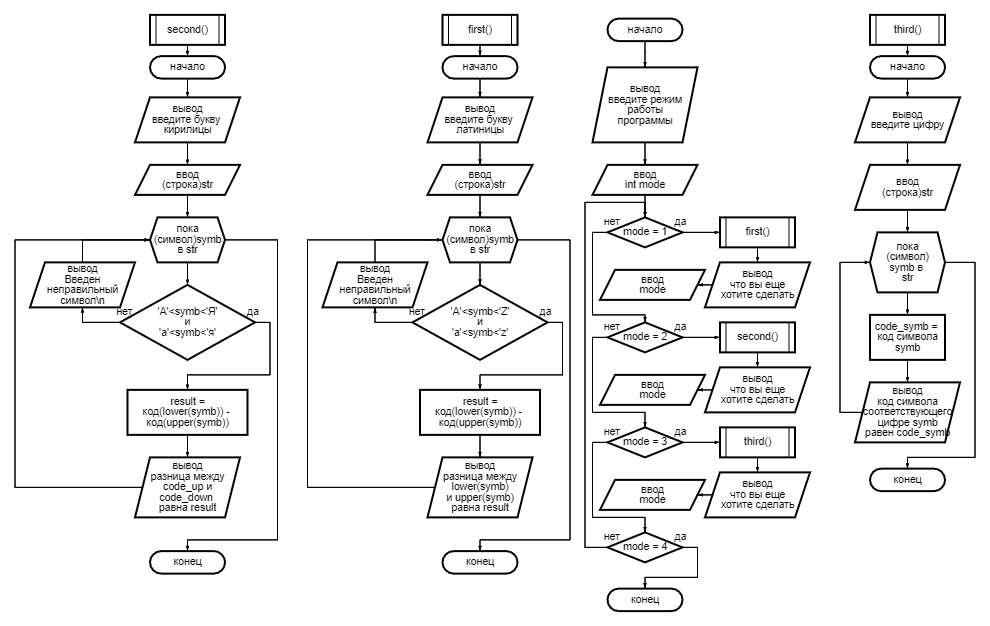
Входные данные:

1. Строка из символов русского алфавита.
2. Строка из символов латинского алфавита.
3. Строка из цифр.

Выходные данные:

1. Разность между верхним и нижним регистром для каждого символа из строки.
2. Разность между верхним и нижним регистром для каждого символа из строки.
3. Код соответствующего символа из таблицы Windows-1251.

Блок-схема:



Модульная схема, псевдокод:

ОПРЕДЕЛЕНИЕ МОДУЛЯ first

ВЫВОД введите букву латиницы

ВВОД (строка)str

ПОКА (символ)symb в str

НАЧАЛО ЦИКЛА

ЕСЛИ ‘A’<symb<’Z’ И ‘a’<symb<’z’ ТО

(integer)result = код(lower(symb)) - код(upper(symb))

ВЫВОД разница между lower(symb) и upper(symb) равна result

КОНЕЦ ЦИКЛА

ОПРЕДЕЛЕНИЕ МОДУЛЯ second

ВЫВОД введите букву кирилицы

ВВОД (строка)str

ПОКА (символ)symb в str

НАЧАЛО ЦИКЛА

ЕСЛИ ‘A’<symb<’Z’ И ‘a’<symb<’z’ ТО

(integer)result = код(lower(symb)) - код(upper(symb))

ВЫВОД разница между lower(symb) и upper(symb) равна result

КОНЕЦ ЦИКЛА

ОПРЕДЕЛЕНИЕ МОДУЛЯ third

ВЫВОД введите цифру

ВВОД (строка)str

ПОКА (символ)symb в str

НАЧАЛО ЦИКЛА

(integer)code = код(symb)

ВЫВОД Код символа, соответствующего введенной цифре symb равен code

КОНЕЦ ЦИКЛА

ОПРЕДЕЛЕНИЕ МОДУЛЯ dop\_1

ВЫВОД введите число

ВВОД (строка)num

ВЫВОД вывожу число без 3ек и 6ок

ПРИСВОИТЬ for\_out = ‘‘

ПОКА i в num

НАЧАЛО ЦИКЛА

ЕСЛИ i != ‘3’ || i != 6 ТО for\_out += i

КОНЕЦ ЦИКЛА

ВЫВОД for\_out

ОПРЕДЕЛЕНИЕ МОДУЛЯ turnip

ПРИСВОИТЬ (словарь)characters = none, ove\_str = 0

ВЫВОД введите количество персонажей

ВВОД char\_num

ПОКА i = 0; i <= char\_num; i++

НАЧАЛО ЦИКЛА

ВЫВОД введите имя персонажа

ВВОД name

ВЫВОД введите силу персонажа

ВВОД streng

(словарь)characters[name] = streng

КОНЕЦ ЦИКЛА

ВЫВОД введите силу необходимую чтобы вытянуть репку

ВВОД turnip

ПОКА (ключ)key в characters

НАЧАЛО ЦИКЛА

ЕСЛИ ove\_str >= turnip ТО ВЫВОД вытянули репку!!!

ИНАЧЕ ove\_str += characters[key]

КОНЕЦ ЦИКЛА

ЕСЛИ ove\_str < turnip ТО ВЫВОД не вытянули репку!!!

ВКЛЮЧИТЬ МОДУЛЬ first

ВКЛЮЧИТЬ МОДУЛЬ second

ВКЛЮЧИТЬ МОДУЛЬ third

ВКЛЮЧИТЬ МОДУЛЬ dop\_1

ВКЛЮЧИТЬ МОДУЛЬ turnip

ВЫВОД введите режим работы программы

ВВОД task

ЕСЛИ task == 1 ТО

ВЫВОД введите режим работы программы:

ВВОД (символ)mode

ПОКА mode != 4

НАЧАЛО ЦИКЛА

ЕСЛИ mode == 1 ТО

ВЫЗВАТЬ first()

ВЫВОД что еще хотите сделать

ВВОД mode

ЕСЛИ mode == 2 ТО

ВЫЗВАТЬ second()

ВЫВОД что еще хотите сделать

ВВОД mode

ЕСЛИ mode == 3 ТО

ВЫЗВАТЬ third()

ВЫВОД что еще хотите сделать

ВВОД mode

ИНАЧЕ

ВЫВОД вы ввели неправильный режим работы программы!!!

ВЫВОД введите режим работы программы:

ВВОД (символ)mode

КОНЕЦ ЦИКЛА

ИНАЧЕ ЕСЛИ task == 2 ТО dop\_1()

ИНАЧЕ ЕСЛИ task == 2 ТО turnip()

Дополнительное задание 2

Постановка задачи:

Составить алгоритм действий всех персонажей сказки «Репка», приводящих к успеху (Вытянули Репку!).

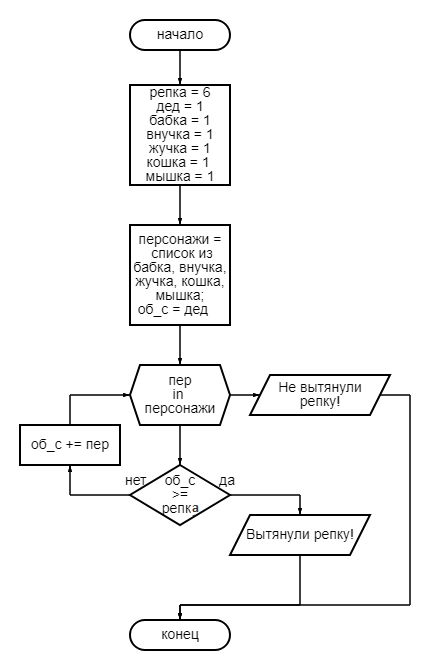
Входные данные:

Количество персонажей, их сила. Количество силы необходимое для вытягивания репки.

Выходные данные:

Строка “Вытянули репку” или “Не вытянули репку”.

Алгоритм решения записанный в блок-схеме(через структурное программирование):



Описание модулей

1. Модуль в котором вызывается функция. Алгоритм: вызвать функцию. Входные данные: None. входные данные: None.
2. Модуль в котором описывается решение задачи. Входные данные: None. Выходные данные: Строка “Вытянули репку!!!” “Не вытянули репку!!!”.