Лабораторная работа №4

Модульное программирование. Стили программирования (стиль оформления кода)

Дополнительно предусмотреть возможность ввода с клавиатуры нескольких символов последовательно.

#include <iostream>

#include <cctype>

#include <windows.h>

#include "(имя твоего файла).h"

using namespace std;

int main()

{

setlocale(LC\_CTYPE, "Russian");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int choice;

while (true)

{

cout << "Выберите вариант:" << endl;

cout << "1 – определение разницы значений кодов в ASCII буквы в прописном и строчном написании (для символов латинского алфавита)" << endl;

cout << "2 – определение разницы значений кодов в Windows-1251 буквы в прописном и строчном написании (для символов русского алфавита)" << endl;

cout << "3 – определение кода символа, соответствующего введенной цифре" << endl;

cout << "4 – выход из программы" << endl;

cin >> choice;

switch (choice)

{

case 1:

{

calcLatinDiff();

break;

}

case 2:

{

calcRusDiff();

break;

}

case 3:

{

getNumCode();

break;

}

case 4:

{

cout << "Программа завершена" << endl;

return 0;

}

default:

{

cout << "Неверный выбор" << endl;

break;

}

}

}

}

Входные и выходные данные.

Входные данные:

1. Объявление переменной char symbol и переменной выбора int choice;

2. Выбор операции (1, 2, 3, 4)

3. Латинская буква, русская буква или цифра в зависимости от выбора операции.

Выходные данные:

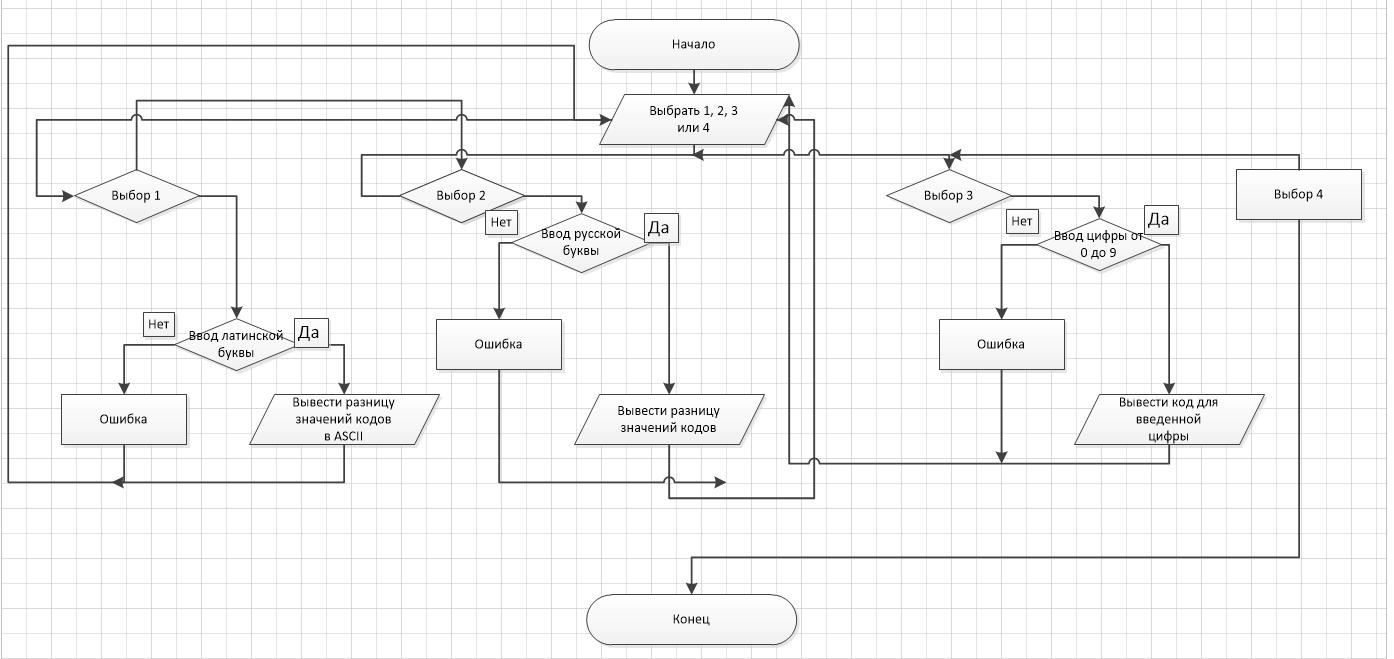
1. Для операции 1: разница между прописной и строчной буквой в ASCII

2. Для операции 2: разница между прописной и строчной буквой в кодировке Windows-1251

3. Для операции 3: код цифры в кодировке Windows-1251

4. Программа завершается при выборе операции номер 4.

Блок-схема алгоритма решения задачи:



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

Главный модуль:

Состав: функция main().

- Назначение: управление всей программой, вывод сообщения пользователю, обработка выбора пользователя и вызов соответствующих модулей для выполнения дальнейших операций

- Входные данные: выбор пользователя (ввод с клавиатуры цифры)

- Выходные данные: результат выполнения выбранной операции или, при выборе цифры 4, завершение программы

- Алгоритм:

1. Вывести сообщение с выбором операций для пользователя

2. Считать выбор пользователя

3. В зависимости от выбора вызвать соответствующий модуль для выполнения операции

4. Повторять шаги 1-3 до выбора завершения программы (цифры 4)

Модуль 2:

Состав: ASCIIDifference

- Назначение: вычисление разницы между значениями ASCII-кодов для прописной и строчной буквы латинского алфавита

- Входные данные: латинская буква (ввод с клавиатуры)

- Выходные данные: разница между значениями ASCII-кодов или сообщение об ошибке

- Алгоритм:

1. Считать введенную латинскую букву

2. Проверить, является ли символ буквой

3. Если да, то вычислить разницу между ASCII-кодами прописной и строчной буквы

4. Вывести результат или сообщение об ошибке

Модуль 3: CyrillicDifference

- Назначение: вычисление разницы между значениями кодов в Windows-1251 для прописной и строчной буквы русского алфавита

- Входные данные: русская буква (ввод с клавиатуры)

- Выходные данные: разница между значениями кодов в Windows-1251 или сообщение об ошибке

- Алгоритм:

1. Считать введенную русскую букву

2. Проверить, является ли символ буквой

3. Если да, то вычислить разницу между значениями кодов в Windows-1251 для прописной и строчной буквы

4. Вывести результат или сообщение об ошибке

Модуль 4: digit

- Назначение: определение кода символа по введенной цифре

- Входные данные: цифра (ввод с клавиатуры)

- Выходные данные: сообщение об ошибке

- Алгоритм:

1. Считать введенную цифру

2. Проверить, является ли введенный символ цифрой

3. Если да, вычислить код цифры в кодировке Windows-1251

4. Вывести сообщение об ошибке и выйти из консоли

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

М1

М2 М3

Псевдокод:

Этап 1: задаем заголовок программы, соответствующий ее основной функции

int main()

Этап 2: while(true)

{

Case 1: подсчет разницы символов латинского алфавита (\*)

Case 2: подсчет разницы символов русского алфавита ()

Case 3: вывод кода для заданной цифры(\*\*\*)

Case 4: завершение программы

}

Этап 3: пошагового уточнения: детализация фрагмента \*

Ввод последовательности символов

Определение вида буквы(заглавная/маленькая)

Вычисление разницы значений кода

Вывод результата

Этап 4: пошагового уточнения: детализация фрагмента

Ввод последовательности символов

Определение вида буквы(заглавная/маленькая)

Вычисление разницы значений кода

Вывод результата

Этап 5: пошагового уточнения: детализация фрагмента \*\*\*

Ввод последовательности символов

Определение кода цифры

Вывод результата

ДОП. ЗАДАНИЯ:

#include <iostream>

Using namespace std;

int main() {

int A; //Задаем переменную А

cout << "Enter an integer: ";

cin >> A;

for (int i = 3; i <= 7; i++) { //Инвертирование битов с 4 по 8

A ^= (1 << i);

}

cout << "Inverted number: " << A << endl; //Вывод инвертированного номера

return 0;

}

Рекомендации по стилю оформления кода:

1. Используйте пробелы для отступов, а не табуляцию. Обычно используется отступ в 4 пробела.

2. Имена переменных и функций должны быть осмысленными и описательными. Например, вместо "B" лучше использовать более описательное имя (если это конечно не переменная), чтобы было более понятно

3. Добавьте комментарии к коду, чтобы объяснить его логику и функциональность.

4. Используйте const там, где это возможно, чтобы указать, что переменная не будет изменяться.

5. Разделяйте длинные строки кода на несколько строк для улучшения читаемости.

6. Используйте операторы сравнения (==, !=, <, >) вместо битовых операций для сравнения значений.

7. Используйте пространства имен (namespace) для организации кода.

Разбить на модули:

1. Ввод исходного числа:

#include <iostream>

using namespace std;

int getInput() {

int num;

cout << "Enter an integer: ";

cin >> num;

return num;

}

2. Инвертирование битов:

int invertBits(int num) {

for (int i = 3; i <= 7; i++) {

num ^= (1 << i);

}

return num;

}

3. Вывод результата:

void outputResult(int num) {

cout << "Inverted number: " << num << endl;

}

4. Основная функция:

int main() {

int inputNum = getInput();

int invertedNum = invertBits(inputNum);

outputResult(invertedNum);

return 0;

}

