Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Лабораторная работа

По дисциплине «Основы алгоритмизации и программирования»

На тему «Структурное программирование»

Выполнил:

Студент 1 курса 7 группы

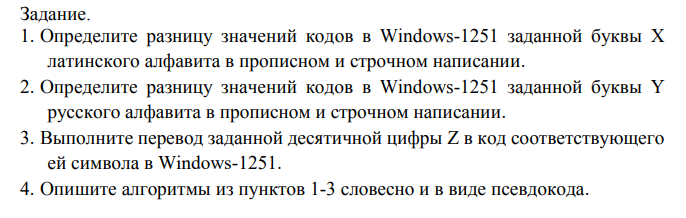
Ленкевич Павел Андреевич

Преподаватель: асс. Якубенко К.Д.

2023, Минск

**Структурное программирование**

**X: e, Y: а, Z: 5**

****

**Задание 1.**

**X.** e – 0065(16) = 101(10), E – 0045(16) = 69 (10). Разница составляет 32(10) символа

**Задание 2.**

**Y.** а - 0430(16) = 1072(10), А – 410(16) = 1040 (10). Разница составляет 0020(16), что равно 32(10) символам

**Y.** Е – 0415, е – 0435. Разница в коде составляет 0020(16) = 32(10) = 100000(2)

**Задание 3.**

**Z.** 5(10) = 5(10) + 48(10) = 5(16) + 30(16) = 35

**Задание 4.**

**1-2.**

**Словесно:**

1. Вводим прописной символ (Upper)
2. Вводим строчный символ (Lower)
3. Вычисляем разницу в их кодах: Difference = Lower – Upper
4. Вывод Difference
5. Конец

**Псевдокод:**

НАЧАЛО

ПРИСВОИТЬ Upper = 0 и Lower = 0

ВВОД Строчной буквы - Lower

ВВОД Прописной буквы – Upper

ВЫЧИСЛИТЬ Difference = Lower – Upper

ВЫВОД Difference

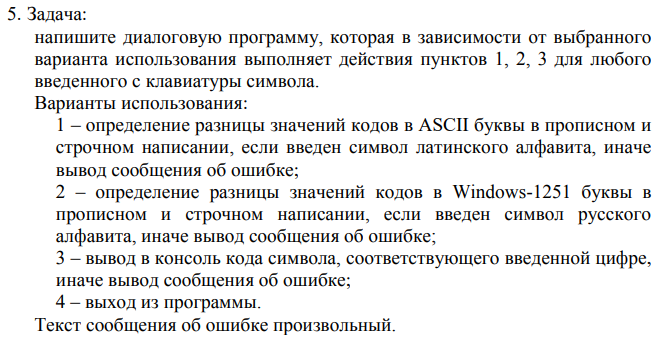
КОНЕЦ

**3.**

**Словесно:**

1. Вводим цифру number
2. number = number + 48
3. Переводим number в шестнадцатеричную систему
4. Выводим number

**Задача 5.**

****

**Постановка задачи:** создать программу, которая в зависимости от варианта будет выводить код / разницу кодов между строчным и прописным символом. Вариант 1 должен по введённому символу выводить разницу кодов латинских символов. Вариант 2 по введённому символу должен выводить разницу русских символов. Вариант 3 должен отображать код введённого символа

**Программа:**

#include <iostream>

#include <locale.h>

#include <Windows.h>

using namespace std;

void main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setlocale(LC\_CTYPE, "Russian");

wchar\_t lowercase, uppercase, symbol;

int variant, exit\_number;

for (;;) {

cout << "Введите номер варианта использования: "; cin >> variant;

switch (variant)

{

case 1:

cout << "Введите заглавную латинскую букву: \n"; wcin >> uppercase;

cout << "Введите строчную латинскую букву: \n"; wcin >> lowercase;

if (64 < uppercase && uppercase < 91 && 96 < lowercase && lowercase < 123)

cout << "Разница значений кодов: " << lowercase - uppercase << endl;

else

cout << "Ошибка. Введена не латинская буква" << endl;

break;

case 2:

wcout << L"Введите заглавную русскую букву: \n"; wcin >> uppercase;

wcout << L"Введите строчную русскую букву: \n"; wcin >> lowercase;

if ((1039 < uppercase) && (uppercase < 1072) && (1071 < lowercase) && (lowercase < 1104))

wcout << L"Разница значений кодов: " << int(lowercase - uppercase) << endl;

else

wcout << L"Ошибка. Введена не русская буква" << endl;

break;

case 3:

wcout << L"Введите любой символ: "; wcin >> symbol;

wcout << L"Код введённого вами символа: " << int(symbol) << endl;

break;

case 4:

wcout << L"До свидания!" << endl;

break;

default:

wcout << L"Нет такого варианта!!!" << endl;

break;

}

cout << "Если хотите выйти, нажмите 0, если хотите продолжить нажмите 1: "; cin >> exit\_number;

if (exit\_number == 0) {

break;

}

if (exit\_number == 1) {

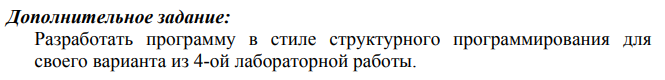
continue;

}

}

}

**Задание 6.**

****

**Программа:**

#include <iostream>

#include <locale.h>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

float cost = 0, bus\_ticket, metro\_ticket;

int transfer, variant1;

cout << "Введите стоимость билета на автобус: "; cin >> bus\_ticket;

cout << "Введите стоимость билета на метро: "; cin >> metro\_ticket;

for (int i = 0; i < 2; i++) {

cout << "Поездка до ";

if (i == 0)

cout << "дома только на автобусе? (1 - да, 2 - нет)" << endl;

else

cout << "цирка только на автобусе? (1 - да, 2 - нет)" << endl;

cin >> variant1; cout << endl;

switch (variant1)

{

case 1:

cout << "Есть пересадки? (1 - да, 2 - нет)" << endl; cin >> variant1; cout << endl;

switch (variant1)

{

case 1:

cout << "Введите количество пересадок: "; cin >> transfer; cout << endl;

cost = cost + bus\_ticket\* (transfer + 1);

break;

case 2:

cost = cost + bus\_ticket;

break;

}

break;

case 2:

cout << "Поездка до ";

if (i == 0)

cout << "дома только на метро? (1 - да, 2 - нет)" << endl;

else

cout << "цирка только на метро? (1 - да, 2 - нет)" << endl;

cin >> variant1; cout << endl;

switch (variant1)

{

case 1:

cost = cost + metro\_ticket;

break;

case 2:

break;

}

case 3:

cout << "Вывод: Поездка до ";

if (i == 0)

cout << "дома разу на двух видах транспорта" << endl;

else

cout << "цирка cразу на двух видах транспорта" << endl;

cout << endl;

cout << "Введите количество автобусных пересадок: "; cin >> transfer; cout << endl;

cost = cost + bus\_ticket \* (transfer + 1) + metro\_ticket;

break;

}

}

cout << "Стоимость поездки по городу: " << cost; cout << endl;

}

**Структурное программирование:**

1. Структурное программирование – стиль написания программ без goto.

Структурное программирование – методология и технология разработки программных средств, основанная на трёх базовых конструкциях: − следование; − ветвление; − цикл

1. Принципы разработки:
   1. − программирование «сверху-вниз» (нисходящее программирование);
   2. − модульное программирование с иерархическим упорядочением связей между модулями/подпрограммами «От общего к частному
2. Этапы проектирования:

− формулировка целей (результатов) работы программы;

− представление процесса работы программы (модель);

− выделение из модели фрагментов: определение переменных и их назначения, стандартных программных контекстов.

1. Этапы структурного проектирования:
   1. Формулировка целей производится на естественном языке.
   2. Создается модель происходящего процесса.
   3. Сбор информации, касающейся характеристик алгоритма.
   4. В модели выделяется наиболее существенная часть, для которой строится алгоритм.
   5. Определяются переменные, необходимые для формального представления алгоритма и формулируется их назначение.
   6. Выбирается одна из конструкций - простая последовательность действий, условная или цикл.
   7. Для вложенных конструкций необходимо вернуться на предыдущие этапы проектирования.
2. Технология структурного программирования базируется на следующих методах:
   1. − нисходящее проектирование;
   2. − пошаговое проектирование;
   3. − структурное проектирование (программирование без goto);
   4. − одновременное проектирование алгоритма и данных;
   5. − модульное проектирование;
   6. − модульное, нисходящее, пошаговое тестирование.