|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.03 Прикладная информатика**

**Отчет**

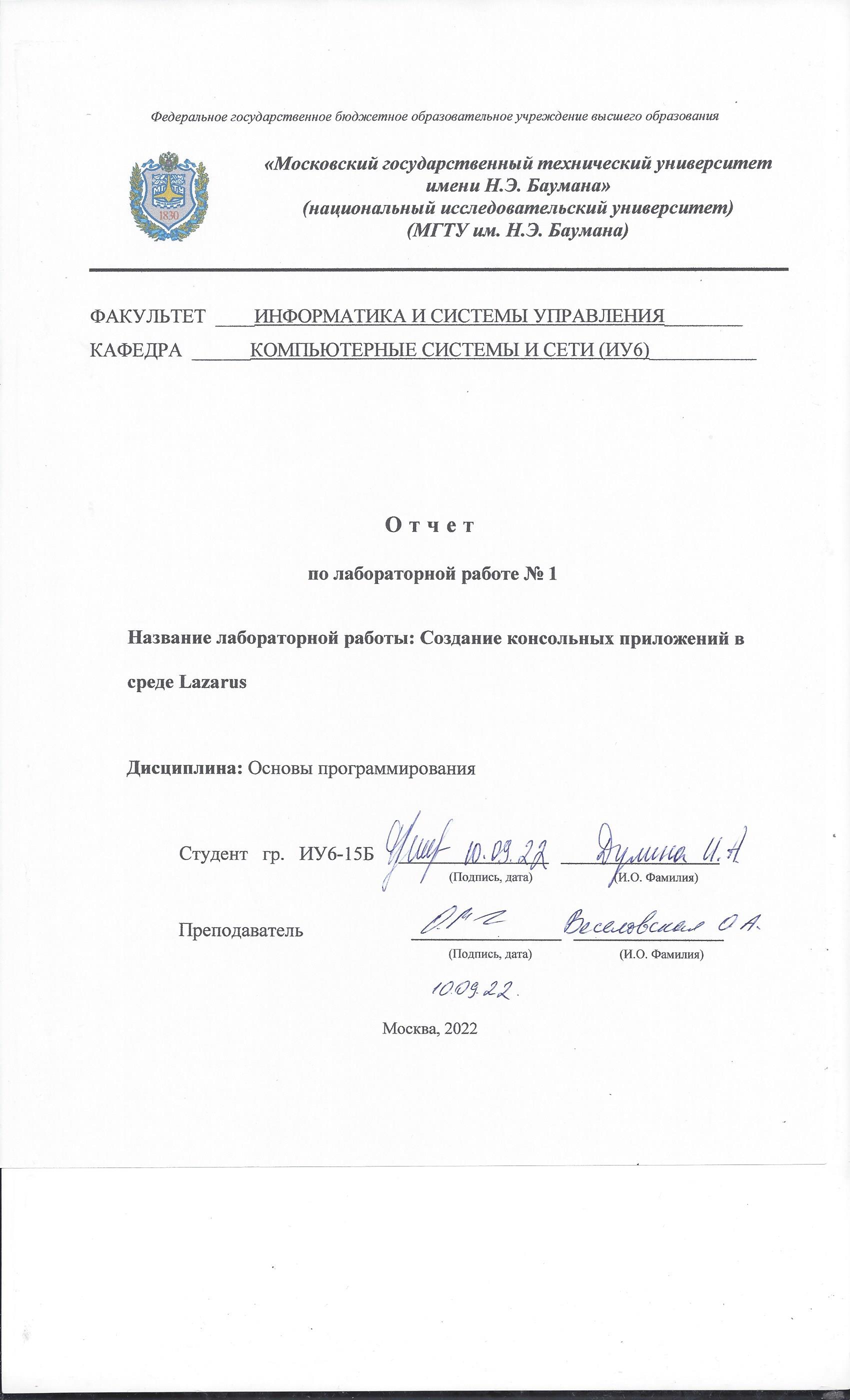
|  |  |
| --- | --- |
| **по домашней работе №** | 2 |

**Название:**

Программирование на С++

**Дисциплина:** Объектно-ориентированное программирование





|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-25 Б |  |  |  |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  | 20.03.2023 |  |
| Преподаватель |  |  |  |  |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

20.03.2023

Москва, 2023

**Вариант 8**

**Часть 2.1. Функции**

**Задание:** Написать программу, определяющую все меньшие n (n<=106) натуральные числа, которые являются палиндромами как в десятичной, так и в двоичной системе. Вывести на печать все найденные числа в десятичной и двоичной системе. Например: 33 – 100001. При программировании использовать функции.

Код программы:

#include <string.h>

#include <iostream>

#include <stdlib.h>

using namespace std;

bool pal (char \*s) {

int i, j, k=0;

j = strlen(s);

for (i = 0; i < j/ 2; i++)

if (s[i] == s[j - i - 1]) k++;

if (k == j / 2) return true;

else return false;

}

char\* rev(char\*str) {

int i = 0;

while (str[i]) i++; //считаем длину строки

for (int j = 0; j < i / 2; j++)

{

swap(str[j], str[i - j - 1]);

}

return str;

}

int main() {

int i, n;

char str1[11], str[11], ch[3];

for (i = 1; i <= 106; i++) {

strcpy\_s(str1, "");

n = i;

while (n != 0) {

\_itoa\_s(n % 2, ch, sizeof(ch), 10);

strcat\_s(str1, ch);

n /= 2;

}

rev(str1);

\_itoa\_s(i, str, sizeof(str), 10);

if (pal(str) == pal(str1)&&pal(str)==true)

cout << i << " " << str1 << "\n";

}

return 0;

}

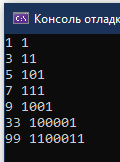


Рисунок 1 – пример работающей версии программы



Рисунок 2 – блок-схема

**Вывод**: мы научились создавать подпрограммы (функции) на языке C++

**Часть 2.2. Текстовая обработка**

**Задание:** Дана последовательность строк. Каждая строка состоит из групп букв, цифр и знаков +,-,\*. Группой букв называется группа, которой не предшествует и за которой не следует буква. Группой цифр называется группа, которой не предшествует и за которой не следует цифра. Группой знаков называется группа, которой не предшествует и за которой не следует знак. Написать программу, определяющую в каждой строке количество групп букв, цифр и знаков и осуществляющую замену группы букв, начинающуюся с гласной буквы (если таковая найдется) на группу букв ‘aaaa’. Вывести на печать исходную и скорректированную последовательности строк.

#include <iostream>

#include <stdlib.h>

#include <string.h>

/\* sjsad323kw+djh\*\*+-hs293891adf

32aa227sj+s-\*wy2+3g59agd

+sie93323+dfiegeife2+\_32\*

dsf343+dsfs43awdaw++dsa

sadad4ew++aasasd834e\*/

using namespace std;

const char b[] = "qwertyuiopasdfghjklzxcvbnm";

const char ch[] = "1234567890";

const char s[] = "+-\*";

void kol(char\*\* list, int k) {

char\* str, \* f;

int cb ,cch,cs;

bool fb, fch, fs;

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

str = list[i];

cb = cch = cs = 0;

fb = fch = fs = false;

f = str;

while (strcmp(f, "")!=0 ){

if (strchr(b, f[0]) != nullptr) {

if (fch == true) cch++;

if (fs == true) cs++;

fb = true; fch = false; fs = false;

f++;

}

else {

if (fb == true) cb++;

if (strchr(ch, f[0]) != nullptr) {

if (fs == true) cs++; fch = true; fb = false; fs = false; f++;

}

else {

if (fch == true) cch++;

fs = true;

fch = false;

fb = false;

f++;

}

}

}

if (fb == true) cb++;

if (fch == true) cch++;

if (fs == true) cs++;

cout << "The " << i+1 << " string: " << cb<<" "<<cch << " " << cs << "\n";

}

}

void zam(char\*\* list, int k) {

char\* str, \*f, \*r;

char g[] = "eyuioa";

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

char str1[50] = ""; char str2[2]="";

str = list[i];

while (strcmp(str, "") != 0) {

f = str;

if (strchr(g, f[0]) != nullptr) {

strcat\_s(str1, "aaaa");

do

f++;

while (strchr(ch, f[0]) == nullptr && strchr(s, f[0]) == nullptr&& strcmp(f, "") != 0);

}

else {

do {

strncpy\_s(str2, f, 1);

strcat\_s(str1, str2);

f++;

} while ((strchr(b, f[0]) != nullptr || strchr(g, f[1]) == nullptr)&& strcmp(f, "") != 0);

}

if (strchr(g, f[1]) != nullptr){

strncpy\_s(str2, f, 1);

strcat\_s(str1, str2);

f++;

}

str = f;

}

puts(str1);

}

}

int main() {

int kolstr;

cout << "The amount: \n";

cin >> kolstr;

char\*\* listnew = new char\* [kolstr];

cout << "The former strings: \n";

cin.ignore();

for (int i = 0; i < kolstr; i++)

{

char\* str = new char[51];

gets\_s(str, 50);

listnew[i] = str;

}

kol(listnew, kolstr);

cout << "The result string \n";

zam(listnew, kolstr);

for (int i = 0; i < kolstr; i++) delete[] listnew[i];

delete[]listnew;

}

**Часть 2.3. Файлы**

**Задание:** Сформировать программным способом символьный файл F. Записать в файл H с сохранением порядка следования те символы файла F, которым в этом файле предшествует буква а.

Вывести на экран оба файла.

Код программы:

#include <iostream>

#include <stdlib.h>

#include <string.h>

#include <stdio.h>

using namespace std;

int main() {

FILE\* f, \* h;

int ch; bool fl = false; int i = 0;

char \*str;

str = new char[40];

fopen\_s(&f, "f.txt", "w");

cout << "Input string: \n";

gets\_s(str, 40);

fprintf\_s(f, "%s", str);

fclose(f);

fopen\_s(&f, "f.txt", "r");

fopen\_s(&h, "h.txt", "w");

cout << "The f file\n";

while (ch = fgetc(f), ch!= EOF) {

if (fl == true) {fputc(ch, h); fl = false;}

if (ch == 'a') fl = true;

putc(ch, stdout);}

cout << "\nThe h file\n";

fclose(h);

fopen\_s(&h, "h.txt", "r");

if (!feof(h)) {

fscanf\_s(h, "%s", str, sizeof(str));

printf("%s", str);

}

else cout << "No elements";

fclose(f);

fclose(h);

delete[]str;

return 0;

}

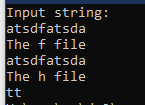


Рисунок 1 – работающая версия программы

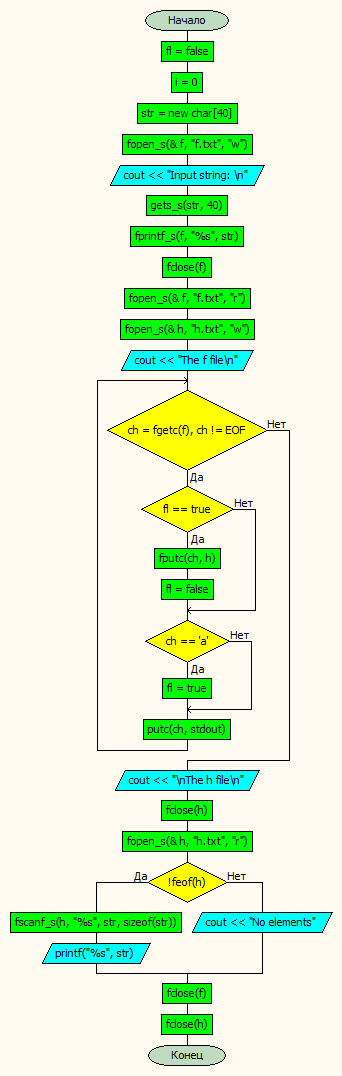


Рисунок 2 – блок-схема

**Вывод**: мы научились обрабатывать файлы на языке C++.