МИНОБРНАУКИ РОССИИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ

ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

«ЛЭТИ» ИМ. В. И. УЛЬЯНОВА (ЛЕНИНА)

ОТЧЕТ

по лабораторной работе №2

по дисциплине «Организация ЭВМ и систем»

Тема: Исследование видеосистемы (текстовый режим).

Студенты гр. 5371 Бергер Э. Э.

Локкина О. С.

Уруков С. Д.

Преподаватель Рукавицын А. Н.

Санкт-Петербург

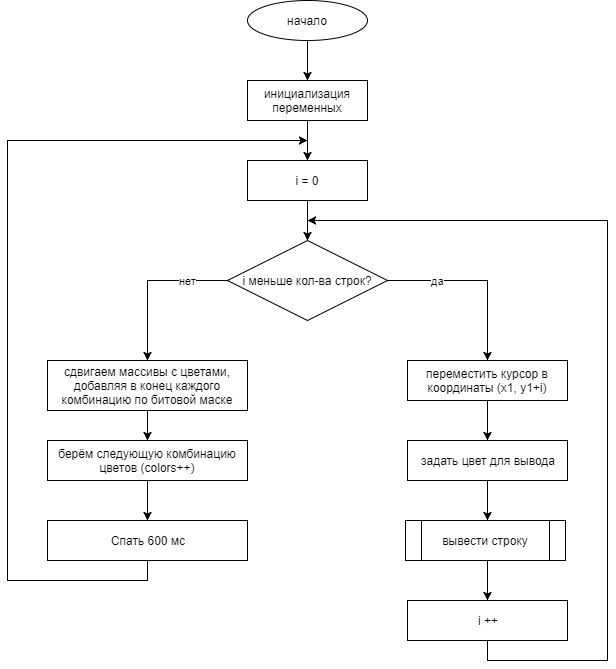
2017

**ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ**

*Вариант 4*

Разработать программу для того, чтобы в окно с координатами x1 = 25, x2 = 55, y1 = 5, y2 = 15 с шагами в T = 0.6 секунд и S = 1 строкой вниз выводилась надпись при всех возможных комбинациях цвета текста и цвета фона. Цвета заданы на английском языке.

**БЛОК-СХЕМА АЛГОРИТМА**



**ПРИМЕРЫ ЗАПУСКА ПРОГРАММЫ**

При запуске программы по заданным координатам рисуется окно, в котором выводятся строки вида «color1 color 2». Первый цвет отвечает за цвет фона, а второй – за цвет текста. С интервалом в 0.6 сек происходит автоматический скроллинг на 1 строку вниз.



**ТЕКСТ ПРОГРАММЫ**

#include "stdafx.h"

#include <iostream>

#include <conio.h>

#include <windows.h>

using namespace std;

const int COLORS\_NUMBER = 16;

void gotoxy(int x, int y) // set cursor position

{

COORD p = { x, y };

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), p);

}

enum COLORS {

BLACK, BLUE, GREEN, CYAN, RED, MAGENTA, BROWN,

LIGHTGRAY, DARKGRAY, LIGHTBLUE, LIGHTGREEN, LIGHTCYAN, LIGHTRED, LIGHTMAGENTA,

YELLOW, WHITE

};

void set\_color(COLORS backgroud, COLORS foreground) // set color for printing

{

HANDLE hConsole;

hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

int x = (backgroud<<4) + foreground;

SetConsoleTextAttribute(hConsole, x);

}

int color\_to\_text(int x) // prints color and return length of word

{

switch (x) {

case BLACK: { cout << "BLACK"; return 5; }

case BLUE: { cout << "BLUE"; return 4; }

case GREEN: { cout << "GREEN"; return 5; }

case CYAN: { cout << "CYAN"; return 4; }

case RED: { cout << "RED"; return 3; }

case MAGENTA: { cout << "MAGENTA"; return 7; }

case BROWN: { cout << "BROWN"; return 5; }

case LIGHTGRAY: { cout << "LIGHTGRAY"; return 9; }

case DARKGRAY: { cout << "DARKGRAY"; return 8; }

case LIGHTBLUE: { cout << "LIGHTBLUE"; return 9; }

case LIGHTGREEN: { cout << "LIGHTGREEN"; return 10; }

case LIGHTCYAN: { cout << "LIGHTCYAN"; return 9; }

case LIGHTRED: { cout << "LIGHTRED"; return 8; }

case LIGHTMAGENTA: { cout << "LIGHTMAGENTA"; return 12; }

case YELLOW: { cout << "YELLOW"; return 6; }

case WHITE: { cout << "WHITE"; return 5; }

default:

{ cout << "undefined"; }

}

}

void row(int b, int f, int width)

{

int least = width;

least -= color\_to\_text(b);

cout << " ";

least--;

least -= color\_to\_text(f);

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

cout << " ";

}

}

void shift(int\* arr, int length, int last\_element)

{

for (int i = 0; i < length - 1; i++) {

arr[i] = arr[i + 1];

}

arr[length - 1] = last\_element;

}

void init()

{

int x1 = 25, x2 = 55;

int y1 = 5, y2 = 15;

int rows = y2 - y1;

int width = x2 - x1;

int color = 0; // Covers all combinations. Changes on each iteration.

int\* background = new int[rows];

int\* foreground = new int[rows];

memset(background, 0, sizeof(int)\*rows);

memset(foreground, 0, sizeof(int)\*rows);

while (true)

{

for (int i = 0; i < rows; i++) { // in this loop renders every row

gotoxy(x1, y1 + i);

set\_color((COLORS)background[i], (COLORS)foreground[i]);

row(background[i], foreground[i], width);

}

// we made shift for each array

shift(background, rows, (color>>4));

shift(foreground, rows, (color&15));

color++;

Sleep(600);

}

}

int main()

{

init();

return 0;

}

**СХЕМА АППАРАТНЫХ СРЕДСТВ**

