МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

ЛАБОРАТОРНАЯ РАБОТА №5

студента 2 курса 251 группы

направления 09.03.04 Программная инженерия

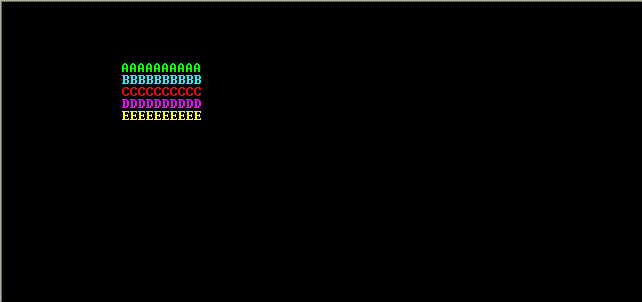
факультета компьютерных наук и информационных технологий

Карасева Вадима Дмитриевича

Саратов 2025

**Текст задания**

**Вариант 6.** Изображение показано на рисунке 5.9. и состоит из 5 строк символов начиная с символа A (ASCII 41h) и далее по алфавиту с разными атрибутами начиная с 0Ah и далее плюс один. В каждой строке по 10 символов, начальная позиция вывода 5:15. Надо выполнить задание, используя прямую работу с видеопамятью (см. пример 5.2).



**Алгоритм программы**

*Алгоритм подпрограммы B10DISPLAY:*

1. В стек помещаем регистры общего назначения
2. В регистр AL помещаем код символа «а» по таблице ASCII
3. В регистр AH помещаем значение атрибута
4. В регистр DI помещаем адрес начального смещения в видеопамяти, который рассчитывается по формуле (№ строки \* 80 + № столбца) \* 2
5. В CX заносим количество строк, которые нужно вывести (5)
6. Создаем цикл для перемещения по строке и вывода символов в каждом столбце строки
7. Загружаем из стека все регистры общего назначения
8. Выполняем возврат из программы в точку вызова

*Алгоритм цикла для перемещения по строке и вывода символов (Row):*

1. В стеке сохраняем количество строк, которые будем выводить
2. Заносим в CX количество столбцов (10)
3. Создаем цикл для вывода символов в каждом столбце строки
4. Записываем символ из регистра AX по адресу DI (адрес в видеопамяти) на экране
5. Переходим на следующий символ (увеличение на 2, так как каждый символ состоит из 2 байт)
6. Уменьшаем индекс для перемещения на следующий столбец, до тех пор, пока не обработаем все столбцы
7. К значению в регистре DI добавляем n положений до начала следующей строки (n = (80 – (кол-во столбцов) \* 2))
8. Восстанавливаем значение CX из стека
9. Переходим к следующему символу
10. Переходим к следующему атрибуту
11. Уменьшаем индекс для перемещения на следующую строку

**Задание 5.1.**

.model small ; Модель памяти SMALL использует сегменты размером не более 64КБ

.stack 100h ; Сегмент стека размером 100h (256 байт)

.186 ; Разрешаем использовать команды процессора 80186

.data ; Начало сегмента данных

first\_symbol db 41h; Строка, в которой находится код символа "A" по таблице ASCII

coloring db 0Ah ; Строка с атрибутами символа (цвет и фон)

.code ; Сегмент кода

start: ; Точка входа в программу

mov AX, @data ; Помещаем указатель на сегмент данных в AX

mov DS, AX ; Помещаем указатель на сегмент данных в DS

mov AX, 0B800h ; Используя сегментный регистр ES,

mov ES, AX ; организовать запись данных в видеопамять по адресу B800h:0000h

; Теперь в ES у нас хранится адрес нулевой страницы экрана.

mov AH, 00h ; Установка видеорежима (перевод экрана в текст. режим с номером 03)

mov AL, 03h ; Стандартный цветной текстовый режим

int 10h ; Вызвать обработчик прерывания

mov AH, 05h ; Установка выводимой (текущей, активной) страницы

mov AL, 00h ; Нулевая страница

int 10h ; Вызвать обработчик прерывания

call B10DISPLAY ; Вызов функции вывода символов на экран

mov AX, 4C00h ; Код завершения программы

int 21h ; Вызвать обработчик прерывания

B10DISPLAY proc

pusha

mov AL, first\_symbol ; Загружаем символ 'A' (код 41h) в AL

mov AH, coloring ; Загружаем атрибут цвета (0Ah — зеленый на черном) в AH

mov DI, 830 ; Начальное смещение в видеопамяти: (строка \* 80 + столбец) \* 2

mov CX, 5 ; Количество строк (5)

Row:

push CX ; Сохраняем счетчик строк в стеке

mov CX, 10 ; Количество столбцов (10)

Col:

mov ES:word ptr[DI], AX ; Выводим символ с атрибутом в видеопамять

add DI, 2 ; Переходим к следующей позиции (+2 байта)

loop Col ; Повторяем для всех столбцов в строке

add DI, 140 ; Переход к началу следующей строки: (80 - 10) \* 2 = 140

pop CX ; Восстанавливаем счетчик строк

inc AL ; Переходим к следующему символу (B, C, D, ...)

inc AH ; Изменяем атрибут цвета (следующий цвет)

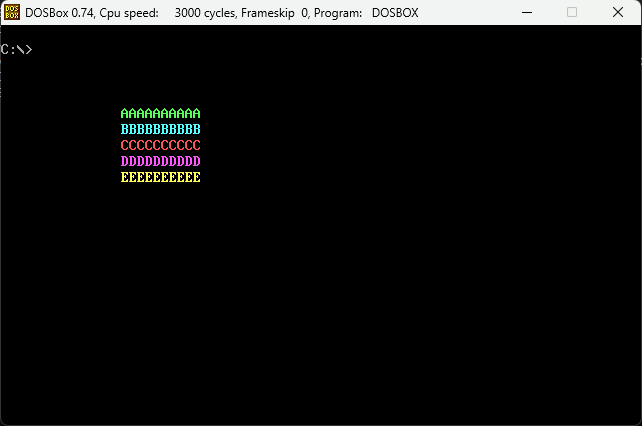
loop Row ; Повторяем для всех строк

popa

ret ; Возврат из процедуры

B10DISPLAY endp

end start



**Контрольные вопросы**

**Каков адрес области видеоданных для режимов 00h - 06h; монохромного текстового режима?**

Область видеоданных устанавливается в зависимости от выбранной программистом активной страницы, по умолчанию это значение эквивалентно 0-й странице и равняется B800h.

**Укажите число страниц, разрешение и число цветов для видеорежима 03**

Количество страниц 0-3, разрешение 720x400, 16 цветов.

**Укажите в двоичной форме содержимое байтов атрибутов для**

* **светло-зелёных символов на синем фоне;**
* **чёрных символов на голубом мигающем фоне.**

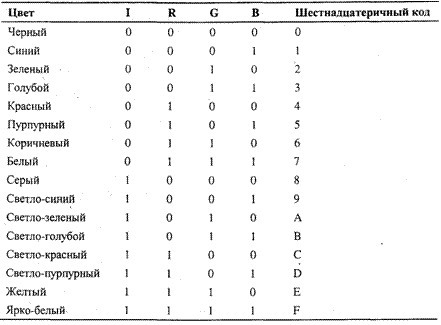
1. 0001 1010 – светло-зелёные символ, синий фон

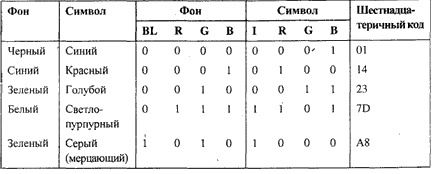
2. 1010 0000 – чёрные символы, голубой-мигающий фон

**Объясните, как ограничивается количество доступных цветов для символа и для фона структурой байта атрибутов.**

Количество цветов для символа и для фона ограничивается с помощью структуры байта атрибутов, который сохраняет свои значения до следующего явного изменения структуры. Фон может иметь 1 из 8-ми цветов, а цвет самих символов – 1 из 16-ти.

Доступные фоны и цвета изображены в таблицах.



****

**Укажите инструкции, необходимые для вывода на экран с помощью функции 09h прерывания INT 10h**

* **12 пурпурных сердечек (ASCII 03h) на зелёном фоне;**
* **8 серых звездочек (ASCII 2Ah) на красном фоне.**

Для данного задания необходимо отредактировать исходную структуру b10display по следующему шаблону:

1. В al заносится предварительно отредактированный char\_str, в котором хранится выводимый символ;

2. В ah заносится предварительно выбранное свойство (цвет, фон) в 16-ричной системе;

3. В cx вместо 8-ми заносится необходимое количество символов, которое

мы хотим вывести.

*Для первого пункта:*

char\_str: 03h

mov ah, 25h

mov cx, 12

*Для второго пункта:*

char\_str: 2Ah

mov ah, 48h

mov cx, 8