

## Лабораторная работа № 12 (4)

**Тема:** Разработка программ вычисления целочисленных выражений, работы с видеобуфером в текстовом и графических режимах.

**Цель:** Освоить написание программ с выводом информации напрямую в видеобуфер в текстовом и графических режимах.

### Краткая теория

Вывод на информации экран может осуществляться в двух основных типах видеорежимов: текстовом и графическом. Реально в видеоадаптерах CGA/EGA/VGA/SVGA может быть несколько текстовых и графических видеорежимов различающихся по разрешению и количеству цветов. В рамках лабораторной работы будут использоваться два видеорежима:

- текстовый (номер 3h): 25 строк на 80 столбцов, каждый символ кодируется двумя байтами: 1 – код символа, 2 – параметры символа (цвет символа и цвет фона);
- графический (номер 13h): 200 строк на 320 столбцов, каждый пиксель (точка на экране) кодируется одним байтом (256 цветов).

Для переключения режимов при программировании в реальном режиме (под ОС MS DOS) используется сервис BIOS реализованный в прерывании 10h. Нулевая функция этого прерывания (содержимое регистра АН равно 0) осуществляет переключение видеорежимов, номер режима должен быть записан в регистр AL.

Например, переключение в видеорежим 13h осуществляется следующим образом:

```
mov    ax, 13h
int    10h
```

а переключение в текстовый режим:

```
mov    ax, 13h
int    10h
```

Пространство памяти видеобуфера в текстовом режиме располагается в сегменте, расположенном по адресу В800h, а в графическом режиме – А000h. Смещение 0 – это адрес левого верхнего угла экрана. Вычисление смещения символа с координатами (X,Y) на экране в текстовом режиме 25 строк 80 столбцов осуществляется по следующей формуле:

$$OFFSET = (Y \times 80 + X) \times 2$$

где константа 80 – ширина экрана в знаках, константа 2 – для учета того, что каждое знакоместо на экране кодируется двумя байтами: код символа и параметры цвета.

Вычисление смещения пикселя с координатами (X,Y) на экране в графическом режиме 320x200 256 цветов осуществляется по следующей формуле:

$$OFFSET = Y \times 320 + X$$

где константа 320 – ширина экрана в пикселях.

В общем случае при вычислении смещения в видеобуфере, по которому располагается элемент с координатами (X,Y) необходимо умножить координату Y на ширину экрана и добавить значение координаты X. Если размер одного элемента более 1 байта (2 байта, 4 байта и т.д.), то полученное значение необходимо умножить на это значение. Также необходимо учитывать, что нумерация начинается с нуля, т.е. если у нас установлен графический режим 320X200, то координата X лежит в диапазоне [0;319], а координата Y – [0;199]. Начало координат располагается в левом верхнем углу экрана.

Несколько примеров при работе с текстовым видеобуфером.

**ПРИМЕР 1:** Разработать программу, выводящую на экран количество символов 'А' находящихся на экране. Количество вывести в левом верхнем углу экрана в десятичной системе счисления.

```

.486
model tiny
Code SEGMENT use16
    ASSUME cs:Code, ds:Code, ss:Code
    org 100h
start:
    mov ax, 0b800h ;Загрузка в AX адреса сегмента видеобуфера
    mov ds, ax     ;Установка DS на видеобуфер
    xor ax, ax     ;Очистка AX
    mov si, ax     ;Запись начала видеобуфера в SI
    mov dx, ax     ;Инициализация счетчика
    mov cx, 25*80  ;Запись в CX количества повторений
loop0:
    lodsw          ;Загрузка в AX из DS:[SI]
    cmp al, 'A'    ;Сравнение с 'A'
    jnz next0      ;Если не 'A', то переход на next0
    inc dx         ;Инкремент DX
next0:
    loop loop0     ;Цикл
    push ds        ;Установка регистра ES на адрес видеобуфера
    pop es        ;из регистра DS
    xor di, di     ;Очистка DI
    push cs        ;Установка сегмента данных DS
    pop ds        ;на сегмент кода CS
    lea si, Strk   ;В SI загружаем смещение Strk
    add si, 6      ;Переход в конец строки
    mov cx, 10     ;В регистр CX записываем 10
    mov ax, dx     ;В AX заносим количество символов 'A' из DX
next1:
    xor dx, dx     ;Очистка DX
    div cx         ;Делим DX:AX на CX
    add dl, 48     ;Прибавляем к остатку код 0
    mov ds:[si], dl ;Записываем в строку символ числа
    or ax, ax      ;Проверка AX на ноль
    jz next2       ;Если AX=0, то переход на next2
    dec si         ;Декремент SI
    jmp next1      ;Переход на следующую итерацию
next2:
    mov ah, 0Fh    ;Загрузка в AH цвет символов
next4:
    lodsb          ;В AL из DS:SI
    or al, al      ;Проверка AL на ноль
    jz next3       ;Если ноль, то переход на next3
    stosw          ;Запись в ES:[DI] из AX
    jmp next4      ;Переход на next4
next3:
    in al, 60h     ;Чтение в AL из порта 60h
    cmp al, 1      ;Проверка на 1 (ESC)
    jnz next3      ;Если неравно, то переход на next3
    int 20h        ;Завершение программы
Strk db 7 dup(0), 0 ;Резервирование под строку
Code ENDS
end start

```

**ПРИМЕР 2:** Вычислить значение выражения  $(7 \cdot A + B) / C$ . Результат вывести на экран (в левый верхний угол) в десятичной системе счисления. A, B и C – целые знаковые числа разрядностью 16 бит.

```

.486
model small
Data SEGMENT use16
    ASSUME ds:Data
ValA dw -120      ;Значение A
ValB dw 350       ;Значение B
ValC dw -30       ;Значение C
Sign db 0         ;Признак отрицательного значения

```

```

Strk db 7 dup(0),0 ;Резервирование памяти под строку
Data ENDS
Stk SEGMENT use16 STACK
    ASSUME ss:Stk
    db 256 dup(0)
Stk ENDS
Code SEGMENT use16
    ASSUME cs:Code
start:
    mov ax, seg ValA ;Запись в AX сегмента Data
    mov ds, ax ;Запись в DS AX
    mov ax, ValA ;Запись в AX значение A
    mov cx, 7 ;Запись в CX значение 7
    imul cx ;Умножение AX*CX -> DX:AX
    add ax, ValB ;Сложение AX со значением B
    adc dx, 0 ;Учет переноса в DX
    mov cx, ValC ;Запись в CX значения C
    idiv cx ;Деление DX:AX / CX
    or ax, ax ;Проверка AX на ноль
    jns next0 ;Если число > 0, то переход на next0
    neg ax ;Преобразуем в положительное число
    mov byte ptr Sign, 1 ;Сохраняем в Sign признак
    ;отрицательного результата
next0:
    lea si, Strk ;Загружаем в SI смещение Strk
    add si, 6 ;Переходим в конец строки
    mov cx, 10 ;Заносим в CX значение 10
next1:
    xor dx, dx ;Очистка DX
    div cx ;Деление DX:AX / CX
    add dl, 48 ;Суммируем остаток с кодом '0'
    mov ds:[si], dl ;Записываем в строку
    or ax, ax ;Проверка AX на ноль
    jz next2 ;Если ноль, то переходим на next2
    dec si ;Уменьшаем SI на 1
    jmp next1 ;Переход на next1
next2:
    mov ax, 0b800h ;Заносим в AX адрес сегмента видеобуфера
    mov es, ax ;Устанавливаем ES на видеобуфер
    xor di, di ;Очистка DI
    mov ah, 0Fh ;В AH записываем код цвета
    cmp byte ptr Sign, 1 ;Если результат был положительным, то
    jnz next4 ;переход на next4
    mov al, '-' ;Записываем в AL код '-'
    stosw ;Запись AX в ES:[DI]
next4:
    lodsb ;Загрузка в AL из DS:[SI]
    or al, al ;Проверка AL на ноль
    jz next3 ;Если ноль, то переход на next3
    stosw ;Запись AX в ES:[DI]
    jmp next4 ;Переход на next4
next3:
    in al, 60h ;Чтение в AL из порта 60h
    cmp al, 1 ;Сравнение с 1 (ESC)
    jnz next3 ;Если не равно, то переход на next3
    mov ax, 4c00h ;Запись в AX номер функции выхода
    int 21h ;Завершение программы
Code ENDS
end start

```

Пример программы в графическом режиме. Построить график функции  $y = \frac{3}{5} \cdot x + 5$ .

Центр координат графика должен располагаться в левом нижнем углу экрана. Цвет фона графика и цвет линии должен кодироваться в виде переменных, описанных в сегменте данных.

```
.486
model small
Data SEGMENT use16
    ASSUME ds:Data
BkCol db 16 ;Цвет фона
LnCol db 15 ;Цвет линии
Data ENDS
Stak SEGMENT use16 STACK
    ASSUME ss:Stak
    db 256 dup(0)
Stak ENDS
Code SEGMENT use16
    ASSUME cs:Code
start:
    mov ax, 13h ;Запись в AH - 0 (инициализация видеорежима),
                ;в AL - 13h (режим 320X200 256 цветов)
    int 10h ;Вызов прерывания BIOS - 10h
    mov ax, seg BkCol ;Запись в AX адреса сегмента Data
    mov ds, ax ;Установка DS на сегмент Data
    mov ax, 0a000h ;Запись в AX адреса сегмента видеобuffers
    mov es, ax ;Установка ES на сегмент видеобuffers
    xor di, di ;Очистка DI
    mov al, BkCol ;Запись в AL код цвета фона
    mov ah, al ;Запись в AH код цвета фона
    mov cx, 32000 ;Заносим в CX количество повторений (320*200/2)
    rep stows ;Закрашивание экрана цветом фона
    xor ax, ax ;Инициализация координаты X в регистре AX
next0:
    push ax ;Сохранение координаты X в стеке
    mov cx, 3 ;Запись в CX константы 3
    imul cx ;Вычисление 3*X
    mov cx, 5 ;Запись в CX константы 5
    idiv cx ;Вычисление (3*X)/5
    add ax, 5 ;Вычисление ((3*X)/5)+5 (AX содержит значение Y)
    mov bx, 199 ;Перевод Y из системы координат с центром в левом
    sub bx, ax ;нижнем углу в левый верхний угол
    mov ax, bx ;Записываем в AX координату Y в новой системе
    mov cx, 320 ;Запись в CX константы 320 для вычисления смещения
    mul cx ;Умножение на количество столбцов на экране
    pop bx ;Восстанавливаем X в регистр BX
    add ax, bx ;Складываем смещение строки с координатой X
    mov di, ax ;Запись в DI значения AX
    mov al, LnCol ;Запись в AL код цвета линии
    mov es:[di], al ;Запись точки в видеобuffer
    mov ax, bx ;Записываем в AX значение координаты X (из BX)
    inc ax ;Инкремент регистра AX
    cmp ax, 320 ;Сравнение с правой границей экрана
    jnz next0 ;Если не достигли предела, то переход на next0
wait0:
    in al, 60h ;Чтение значения из порта клавиатуры
    cmp al, 1 ;Сравнение AL с кодом клавиши ESC
    jnz wait0 ;Если не равны, то переход на wait0
    mov ax, 3 ;Запись в AH - 0 (инициализация видеорежима),
                ;в AL - 3h (текстовый режим строк 25, 80 столбцов)
    int 10h ;Вызов прерывания BIOS 10h
    mov ax, 4c00h ;Запись в AX номер функции завершения программы
    int 21h ;Вызов прерывания MS DOS 21h
Code ENDS
end start
```

## Ход работы

Во всех заданиях этой лабораторной работы все входные данные инициализируются в тексте программы при объявлении. Завершение работы программ осуществляется по нажатию клавиши ESC.

### Задание 1

Разработать программу, согласно варианту задания. Результат (если он есть) вывести в левом верхнем углу экрана. Это задание выполняется в текстовом режиме. Варианты заданий:

1	Подсчитать количество строк на экране, в которых встречаются цифры.
2	Поменять на экране местами строки с номерами N и M. Значения N и M кодируются в тексте программы в сегменте данных.
3	Подсчитать количество строк на экране, в которых нет печатных символов.
4	Переписать символы строки с номером N на экране в обратном порядке. Значение N кодируется в тексте программы в сегменте данных.
5	Вывести вверху экрана строку, содержащую наибольшее количество букв латинского алфавита в нижнем регистре.
6	Переписать все строки экрана в обратном порядке: первая строка становится последней, а последняя – первой и т.д.
7	Выделить на экране красным цветом все цифры.
8	Определить номер столбца, в котором больше всего букв латинского алфавита.
9	Выделить на экране синим цветом фона все знаки пунктуации.
10	Определить каких символов на экране больше: заглавных букв латинского алфавита, или прописных букв латинского алфавита.
11	Очистить строку, содержащую наибольшее количество цифр.
12	Поменять на экране все заглавные буквы на экране на прописные буквы.
13	Посчитать количество цифр на экране.
14	Вывести все пробелы на экране желтым фоном.
15	Выделить на экране зеленым цветом все буквы латинского алфавита.
16	Увеличить все цифры на экране на 1. Цифра 9 переходит в 0.
17	Определить символ, встречающийся на экране больше всего раз.
18	Сдвинуть циклически все символы строки с номером N на M символов влево. Значения N и M кодируются в тексте программы в сегменте данных.
19	Заменить все символы пунктуации на экране символом пробела с красным фоном.
20	Вывести вверху экрана строку, содержащую наименьшее количество букв латинского алфавита в верхнем регистре.
21	Выделить на экране все гласные буквы латинского алфавита зеленым цветом.
22	Поменять на экране все прописные буквы на экране на заглавные.
23	Сдвинуть циклически все символы столбца с номером N на M символов вверх. Значения N и M кодируются в тексте программы в сегменте данных.
24	Выделить слова на экране фоном зеленого цвета.
25	Поменять на экране местами столбцы с номерами N и M. Значения N и M кодируются в тексте программы в сегменте данных.
26	Определить количество знаков препинания на экране.
27	Заменить на экране все точки на запятые, а запятые на точки.
28	Вывести вверху экрана строку, содержащую большее количество букв латинского алфавита в нижнем регистре.
29	Уменьшить все цифры на экране на 1. Цифра 0 переходит в 9.
30	Сдвинуть циклически все символы строки с номером N на M символов вправо. Значения N и M кодируются в тексте программы в сегменте данных.

## Задание 2

Разработать программу вычисления арифметического выражения. Результат вывести в левом верхнем углу экрана в текстовом режиме. Все значения целые знаковые числа разрядностью 16 бит. Это задание выполняется в текстовом режиме. Варианты заданий:

№	Выражение	№	Выражение
1	$(2 \cdot C - D + 23) / (A / 4 - 1)$	16	$(C / D + 3 \cdot A / 2) / (C - A + 1)$
2	$(C + 4 \cdot D - 123) / (1 - A / 2)$	17	$(4 \cdot C + D \cdot 31) / (D - A - 1)$
3	$(-2 \cdot C + D \cdot 82) / (A / 4 - 1)$	18	$(5 \cdot C + D / \$ + 27) / (A \cdot A - 1)$
4	$(2 \cdot C + D - 52) / (A / 4 - 1)$	19	$(2 \cdot C - D / 2 + 1) / (A \cdot A + 7)$
5	$(C / 4 - D \cdot 62) / (A \cdot A + 1)$	20	$(2 \cdot C / D + 2) / (D - A \cdot A - 1)$
6	$(-2 \cdot C - D + 53) / (A / 4 - 1)$	21	$(1200 / C - D / 4 + 54) / (A \cdot A - 1)$
7	$(2 \cdot C - D / 4) / (A \cdot A + 1)$	22	$(2 \cdot C / A - D \cdot D) / (D + A - 1)$
8	$(2 + C - D \cdot 23) / (2 \cdot A \cdot A - 1)$	23	$(2 \cdot B - A + B \cdot C) / (C / 4 - 1)$
9	$(2 \cdot C - D / 3) / (B - A / 4)$	24	$(41 - D / 4 - 1) / (C / B + A \cdot D)$
10	$(4 \cdot C + D - 1) / (C - A / 2)$	25	$(A - B / 4 - 1) / (C / 35 + A \cdot B)$
11	$(2 \cdot C - D \cdot 42) / (C + A - 1)$	26	$(B / A + 4 \cdot C) / (C - B + 1)$
12	$(25 / C - D + 2) / (B + A \cdot A - 1)$	27	$(76 - A \cdot C / 4) / (1 + C / A + B)$
13	$(C - D / 2 + 33) / (2 \cdot A \cdot A - 1)$	28	$(C - 33 + B / 4) / (A \cdot C / B - 1)$
14	$(4 \cdot C - D / 2 + 23) / (A \cdot A - 1)$	29	$(2 \cdot B - 38 \cdot C) / (B + A / C + 1)$
15	$(C \cdot D + 78) / (A / 2 - 4 \cdot D - 1)$	30	$(C / 4 + 28 \cdot D) / (A / D - C - 1)$

## Задание 3

Разработать программу построения графика функции. Данное задание выполняется в графическом режиме 320X200 256 цветов (режим VGA № 13h). Центр координат графика должен располагаться в левом нижнем углу экрана. Цвет фона графика и цвет линии должен кодироваться в виде переменных, описанных в сегменте данных. Дополнительно программа должна нарисовать горизонтальную линию, равную среднему значению функции на отрезке [0;319]. При завершении программа должна «возвращаться» в текстовый режим. Варианты заданий:

№	Функция	№	Функция
1	$Y = \begin{cases} \frac{X}{4} + 20, & X < 160 \\ -\frac{X}{3} + 113, & 160 \leq X < 320 \end{cases}$	16	$Y = \begin{cases} \frac{X}{2} + 50, & X < 160 \\ -X + 260, & 160 \leq X < 320 \end{cases}$
2	$Y = \begin{cases} \frac{2 \cdot X}{3} + 54, & X < 100 \\ -\frac{X}{2} + 160, & 100 \leq X < 320 \end{cases}$	17	$Y = \begin{cases} \frac{X}{5} + 120, & X < 200 \\ -\frac{X}{4} + 210, & 200 \leq X < 320 \end{cases}$
3	$Y = \begin{cases} \frac{3 \cdot X}{2}, & X < 120 \\ -\frac{3 \cdot X}{4} + 270, & 120 \leq X < 320 \end{cases}$	18	$Y = \begin{cases} 2 \cdot X, & X < 50 \\ -\frac{X}{3} + 216, & 50 \leq X < 320 \end{cases}$
4	$Y = \begin{cases} \frac{2 \cdot X}{5} + 110, & X < 200 \\ -\frac{X}{4} + 240, & 200 \leq X < 320 \end{cases}$	19	$Y = \begin{cases} \frac{4 \cdot X}{6} + 127, & X < 80 \\ -\frac{X}{4} + 200, & 80 \leq X < 320 \end{cases}$

5	$Y = \begin{cases} \frac{5 \cdot X}{3} + 34, X < 100 \\ -\frac{X}{3} + 233, 100 \leq X < 320 \end{cases}$	20	$Y = \begin{cases} \frac{X}{5} + 154, X < 180 \\ -\frac{X}{2} + 280, 180 \leq X < 320 \end{cases}$
6	$Y = \begin{cases} \frac{4 \cdot X}{3} + 30, X < 120 \\ -\frac{X}{4} + 220, 120 \leq X < 320 \end{cases}$	21	$Y = \begin{cases} \frac{X}{4} + 130, X < 200 \\ -\frac{2 \cdot X}{3} + 313, 200 \leq X < 320 \end{cases}$
7	$Y = \begin{cases} 2 \cdot X + 10, X < 90 \\ -\frac{X}{3} + 220, 90 \leq X < 320 \end{cases}$	22	$Y = \begin{cases} \frac{4 \cdot X}{5} + 26, X < 180 \\ -\frac{X}{4} + 215, 180 \leq X < 320 \end{cases}$
8	$Y = \begin{cases} -\frac{2 \cdot X}{3} + 190, X < 100 \\ \frac{X}{3} + 91, 100 \leq X < 320 \end{cases}$	23	$Y = \begin{cases} -\frac{X}{4} + 160, X < 150 \\ \frac{2 \cdot X}{5} + 63, 150 \leq X < 320 \end{cases}$
9	$Y = \begin{cases} -\frac{X}{3} + 190, X < 180 \\ \frac{X}{2} + 40, 180 \leq X < 320 \end{cases}$	24	$Y = \begin{cases} -\frac{X}{5} + 120, X < 200 \\ \frac{X - 200}{2} + 80, 200 \leq X < 320 \end{cases}$
10	$Y = \begin{cases} -\frac{3 \cdot X}{4} + 170, X < 150 \\ \frac{X - 150}{3} + 58, 150 \leq X < 320 \end{cases}$	25	$Y = \begin{cases} -2 \cdot X + 180, X < 90 \\ \frac{X}{4} - 22, 90 \leq X < 320 \end{cases}$
11	$Y = \begin{cases} -\frac{4 \cdot X}{3} + 180, X < 120 \\ \frac{4 \cdot (X - 120)}{3} + 20, 120 \leq X < 320 \end{cases}$	26	$Y = \begin{cases} -\frac{X}{5} + 144, X < 220 \\ \frac{X - 220}{2} + 100, 220 \leq X < 320 \end{cases}$
12	$Y = \begin{cases} -\frac{2 \cdot X}{5} + 120, X < 150 \\ \frac{X}{3} + 10, 150 \leq X < 320 \end{cases}$	27	$Y = \begin{cases} -3 \cdot X + 180, X < 90 \\ \frac{4 \cdot X}{3} - 120, 90 \leq X < 320 \end{cases}$
13	$Y = \begin{cases} -\frac{2 \cdot X}{3} + 160, X < 180 \\ \frac{X}{4} - 5, 180 \leq X < 320 \end{cases}$	28	$Y = \begin{cases} -\frac{3 \cdot X}{5} + 120, X < 200 \\ \frac{X}{4} - 50, 200 \leq X < 320 \end{cases}$
14	$Y = \begin{cases} -\frac{X}{5} + 100, X < 200 \\ \frac{X - 200}{2} + 60, 200 \leq X < 320 \end{cases}$	29	$Y = \begin{cases} -\frac{5 \cdot X}{3} + 190, X < 120 \\ \frac{X - 120}{2} - 10, 120 \leq X < 320 \end{cases}$
15	$Y = \begin{cases} -\frac{X}{2} + 180, X < 160 \\ \frac{X - 160}{4} + 100, 160 \leq X < 320 \end{cases}$	30	$Y = \begin{cases} -\frac{5 \cdot X}{2} + 180, X < 70 \\ \frac{X}{4} - 12, 70 \leq X < 320 \end{cases}$

#### **Задание 4 (необязательное)**

Разработать программу, которая в графическом режиме по заданным координатам двух точек чертит на экране отрезок, соединяющий эти точки. Координаты точек задаются в программе и гарантированно лежат внутри области экрана.

#### **Контрольные вопросы**

1. С помощью какого прерывания BIOS осуществляется управления выводом информации на экран?
2. Какая функция прерывания BIOS 10h осуществляет переключение видеорежимов?
3. Опишите параметры текстового режима с номером 3h.
4. Опишите параметры графического режима с номером 13h.
5. С какого адреса начинается сегмент видеобuffers в текстовом режиме?
6. С какого адреса начинается сегмент видеобuffers в графическом режиме?
7. Как осуществляется вычисление смещения символа с координатами (X,Y) в текстовом режиме?
8. Как осуществляется вычисление смещения пикселя с координатами (X,Y) в графическом режиме?
9. Какие особенности следует учитывать при вычислении смещения символа (пикселя) с заданными координатами в общем случае?