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 №4

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.** Массив из 20 чисел заполнить последовательностью, состоящей наполовину из четных чисел (2, 4, ...) и наполовину из этих чисел, умноженных на 5; организовать вывод массива на экран в виде таблицы 2x10 с фиксированной шириной столбцов:



**Алгоритм программы**

**Шаг 1. Написание сегмента данных**

Объявление в сегменте данных:

1. Массив с именем simple из 30 байтов с неопределенным значением;

2. Строку символов с именем result, содержащую 4 пробела и символ ’$’, завершающий строку;

3. Последовательность управляющих символов с именем nl, состоящую из символа перевода строки с кодом 0Ah, символа возврата каретки с кодом 0Dh и символа конца строки ’$’

**Шаг 2. Заполнение массива числами**

Алгоритм цикла ввода чисел в массив

Предусловие:

1. В регистре CX находится число шагов в цикле, равное 10.

2. В регистре BL находится начальный элемент массива, равный 2.

3. В регистре SI находится индекс начального элемента массива, равный 0.

Тело цикла:

1. Заносим содержимое BL в очередной байт массива simple[SI].

2. Заносим в AL делитель

3. Содержимое BL умножаем на 5

4. Заносим содержимое BL в очередной байт массива simple[SI + 10] - число во второй строке, умноженное на 5.

5. Увеличиваем SI на 1, чтобы перейти к следующему байту массива.

6. Увеличиваем BL на 2, чтобы получить следующее число для занесения в массив.

**Шаг 3. Вывод массива**

Алгоритм цикла вывода

Предусловие внешнего цикла:

1. В регистре CX находится число шагов в цикле, равное 10.

2. В регистре SI находится индекс начального элемента массива, равный 0.

Тело внешнего цикла:

1. Сохраняем CX (счётчик внешнего цикла) в стек

2. Устанавливаем счётчик для внутреннего цикла (вывод 10 чисел в строке)

Тело внутреннего цикла:

1. Заносим очередной байт массива simple[SI] в регистр AL.

2. Заносим основание системы счисления, равное 10, в регистр BL.

3. Вызываем процедуру byte\_asc(шаг 4).

4. Организуем вывод результата на экран:

(a) помещаем номер 9 функции DOS в регистр AH;

(b) заносим в DX адрес result (см. л.р.№1);

(c) вызываем прерывание 21h;

(d) заносим в DX адрес nl (см. л.р.№1);

(e) вызываем прерывание 21h.

5. Увеличиваем SI на 1, чтобы перейти к следующему байту массива.

**Шаг 4. Процедура преобразования числа в строку**

Предусловие: выводимое число находится в регистре AL, основание системы счисления – в регистре BL.

Начало.

1. Заносим длину строки результата (= 4) в регистр SI.

2. Задаем позицию в строке результата для очередной цифры числа (уменьшаем индекс в регистре SI на 1).

3. Заносим 0 в регистр AH, т.к. при делении на байт в качестве делимого выступает регистр AX. (В AL до вызова процедуры уже помещено выводимое число).

4. Выполняем деление AX на BL: неполное частное от деления помещается в AL, а остаток – в AH.

5. Получаем из цифры, соответствующей остатку от деления, код цифры в таблице ASCII (добавляем к цифре (AH) код нуля).

6. Заносим код цифры в result[SI].

7. Проверка условия: если неполное частное (AL) не равно 0, то переход на пункт 2.

Конец.

**Задание 4.1.**

model small

stack 100h

.186 ; Разрешение команд процессора 80186

.data

simple db 20 dup (7) ; Массив из 20 байт для хранения чисел

result db ' $' ; Формат вывода числа с выравниванием по правому краю

nl db 0Dh, 0Ah, '$' ; Перевод строки и возврат каретки

.code

start:

mov ax, @data ; Инициализация сегментного регистра DS

mov ds, ax

; Заполнение массива numbers

mov cx, 10 ; Число шагов в цикле (10)

mov bl, 2 ; Начальное значение (2)

mov si, 0 ; Начальный индекс массива simple

vvod:

mov simple[si], bl ; Запись BL в текущий элемент массива

mov al, 5

mul bl ; Умножение AL на BL (результат в AX)

mov simple[si + 10], al ; Запись результата во вторую половину массива

inc si ; Переход к следующему элементу массива

add bl, 2 ; Увеличение значения BL на 2

loop vvod ; Переход к началу цикла, пока CX != 0

; Вывод массива (2 строки по 10 чисел)

mov cx, 2 ; Внешний цикл выполняется 2 раза

mov si, 0 ; Начальный индекс массива simple

outer\_loop:

push cx ; Сохраняем CX (счётчик внешнего цикла) в стек

mov cx, 10 ; Устанавливаем счётчик для внутр. цикла (10 чисел)

vyvod:

mov al, simple[si] ; Загружаем элемент массива в AL

mov bl, 10 ; Основание системы счисления (10)

call byte\_asc ; Вызов процедуры преобразования числа в строку

; Вывод числа на экран

mov ah, 9 ; Функция DOS для вывода строки

mov dx, offset result ; Адрес строки result

int 21h ; Вызов DOS прерывания

inc si ; Переход к следующему элементу массива

loop vyvod ; Повторяем внутренний цикл 10 раз

; Переход на новую строку после вывода 10 чисел

mov dx, offset nl ; Адрес строки nl (новая строка)

mov ah, 9 ; Функция DOS для вывода строки

int 21h ; Вызов DOS прерывания

pop cx ; Восстанавливаем CX из стека для внешнего цикла

loop outer\_loop ; Повторяем внешний цикл 2 раза

mov ax, 4C00h ; Завершение программы (корректный код выхода)

int 21h

; Процедура преобразования числа в строку

byte\_asc proc

pusha ; Сохраняем регистры

mov si, 3 ; Конец строки результата (позиция 3)

next\_digit:

dec si ; Переход к следующей позиции справа налево

mov ah, 0 ; Обнуляем AH для деления AX на BL

div bl ; Деление AX на BL (10)

add ah, '0' ; Преобразуем цифру в ASCII-код

mov result[si], ah ; Сохраняем ASCII-символ в строке result

cmp al, 0 ; Проверяем, не нулевое ли частное

jne next\_digit ; Если не ноль, продолжаем делить

popa ; Восстанавливаем регистры

ret

byte\_asc endp

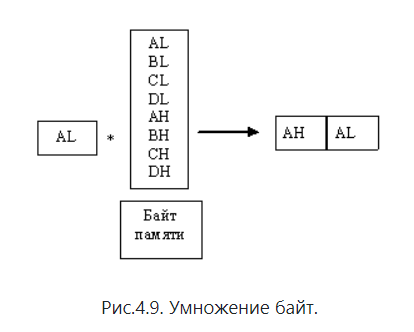
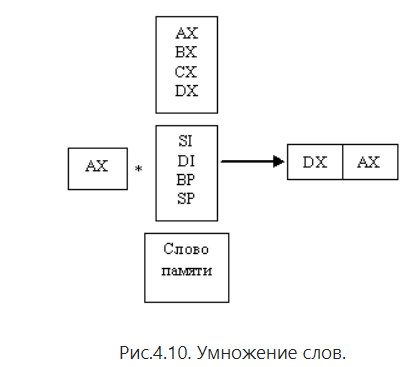
end start

**Контрольные вопросы**

**1. Какой командой можно выделить в памяти место под одномерный массив байтов array размерностью 20?**

array DB 20 dup (?)

**2. Опишите команды умножения на байт и на слово.**

** **

*Байт:* в случае умножения на байт, 1-й множитель содержится в al, а второй задаётся как аргумент в команду mul

mul bl ;(al)\*(bl) -> ax

результат записывается в ax.

*Слово:* в случае умножения на слово, 1-й множитель содержится в ax, а второй задаётся как аргумент в команду mul

mul bx ;(ax)\*(bx)-> dx:ax

результат записывается в dx:ax (dx - старшая часть, ax - младшая часть)

**3. Какое максимальное беззнаковое число можно хранить в элементе массива размером в 1 байт?**

255

**4. Пусть имеется массив: array DW 50 DUP(?). Для доступа к отдельным элементам массива используется адресное выражение array[SI]. Как называется этот способ адресации и как с его помощью будет вычисляться адрес элементов массива?**

Данный способ адресации называется прямой с индексированием, адрес элементов массива вычисляется по формуле:

mov ax, mas[si] ;(ds)+(si) + mas

**5. Каким образом осуществляется перебор элементов некоторого массива A с помощью адресного выражения A[SI], если массив состоит из байтов, слов или двойных слов?**

Для байтов, инкремент к SI (inc SI)

Для слов, необходимо прибавлять 2 (add SI, 2)

Для двойных слов, необходимо прибавлять 4 (add SI, 4)

**6. Для некоторого массива A каким будет результат выполнения команды mov DI, A и команды mov DI, offset A?**

mov DI, A - копирует данные в регистр DI

mov DI, offset A - копирует в DI адрес начала массива A. (Оператор offset возвращает адрес (смещение) некоторой метки данных относительно начала сегмента)