МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

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

Кафедра дискретной математики и информатики

ЛАБОРАТОРНАЯ РАБОТА №4

студента 2 курса 221 группы

направления 09.03.01 Информатика и вычислительная техника

факультета компьютерных наук и информационных технологий

Мусатова Федора Алексеевича

Саратов 2025

Вариант 7. Массив из 20 чисел заполнить последовательностью, состоящей наполовину из нечетных чисел (1, 3, ...) и наполовину из квадратов этих чисел; организовать вывод массива на экран в виде таблицы 2x10 с фиксированной шириной столбцов:



Рис.4.8. Вывод в виде таблицы 2x10 массива из 20 чисел.

**Алгоритм программы**

* 1. Вывод фамилии и номера группы
  2. Вывод массива
     1. Вывод первой строки (обход 0-9 индексов)
        1. Вывод числа
           1. Приведение к цифре и сохранение в стеке
           2. Извлечение цифры из стека, её приведение к символу и вывод на экран
     2. Вывод второй строки (обход 10-19 индексов)
        1. Аналогично выводу первой строки
  3. Завершение работы программы

**Текст программы**

.model small

.stack 100h

.data

array dw 20 dup(0) ; Массив из 20 слов (16 бит)

buffer db 6 dup('$') ; Буфер для чисел (максимум 5 цифр)

newline db 13,10,'$' ; Перевод строки

.code

start:

mov ax, @data

mov ds, ax

mov es, ax

; Заполнение нечетных чисел (1,3,...,19)

mov cx, 10

mov ax, 1

lea di, array

fill\_odds:

stosw ; Записываем AX в [DI] и увеличиваем DI на 2

add ax, 2

loop fill\_odds

; Заполнение квадратов

mov cx, 10

lea si, array ; Указатель на нечетные числа

lea di, array+20; Указатель на вторую половину (10\*2=20 байт)

fill\_squares:

lodsw ; Загружаем число в AX

mul ax ; AX = AX \* AX (результат в DX:AX)

stosw ; Записываем AX (старшая часть DX игнорируется)

loop fill\_squares

; Вывод массива

mov cx, 20

lea si, array

xor bx, bx ; Счетчик элементов

print\_loop:

lodsw ; Загружаем число в AX

; Сохраняем CX

push cx

; Преобразование AX в строку

lea di, buffer + 5

mov byte ptr [di], '$'

mov cx, 10

convert:

xor dx, dx

div cx ; AX = DX:AX / 10

add dl, '0'

dec di

mov [di], dl

test ax, ax

jnz convert

; Восстанавливаем CX

pop cx

; Удаляем ведущие нули

cmp byte ptr [di], '0'

jne print

mov byte ptr [di], ' '

print:

; Вывод числа

mov ah, 09h

mov dx, di ; Адрес начала числа в буфере

int 21h

; Вывод разделителя

inc bx

mov dl, 9

cmp bx, 10

je print\_nl

cmp bx, 20

je print\_nl

jmp print\_cont

print\_nl:

mov ah, 02h

mov dl, 13

int 21h

mov dl, 10

int 21h

jmp next

print\_cont:

mov ah, 02h

int 21h

next:

loop print\_loop

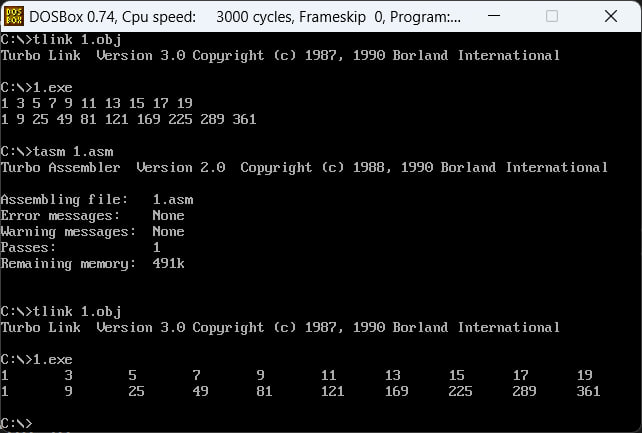
; Завершение

mov ax, 4C00h

int 21h

end start

**Скриншот результата работы программы**



**Ответы на контрольные вопросы**

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

Ответ: можно выделить с помощью директивы определения данных

Array **DB** 25 DUP (?)

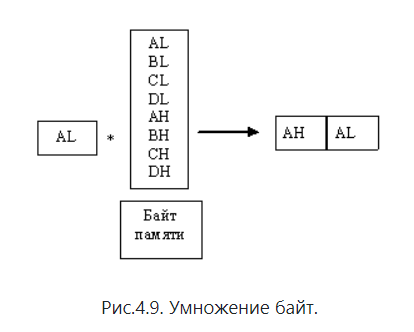
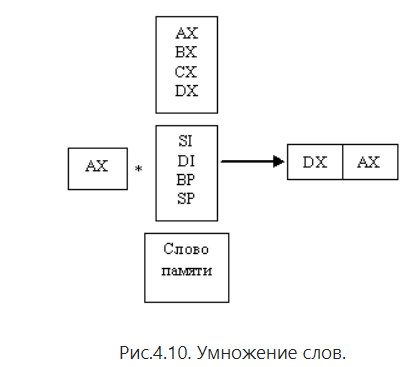
(вместо DB могут быть также DW, DD, DQ, DT)

DUP – оператор повторения

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

Для умножения используется команда MUL: MUL <второй множитель>

Первый множитель будет выбран исходя из размера второго множителя (это будет AL и AX).

При умножении 8-битовых операндов результат всегда помещается в AX. При умножении 16-битовых данных результат, который может быть длиною до 32 бит, помещается в пару регистров: В регистре DX содержатся старшие 16 бит, а в регистре AX – младшие 16 бит. Умножение не допускает непосредственного операнда.

Пример:

MUL BX ;DX:AX=AX\*BX  
MUL AX ;DX:AX=AX2  
MUL BL ;AX=AL\*BL

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

Ответ: число 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?**

Данная адресация называется **индексной адресацией со смещением**.

Имя массива определяет начало массива, а переход от одного элемента к другому в массиве осуществляется с помощью содержимого индексного регистра. То есть к адресу массива данных прибавляется содержимое регистра SI.

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

В массиве байтов SI += 1 (смещение по байту).

В массиве слов SI += 2 (смещение по слову, где слово = 2 байта).

В массиве двойных слов SI += 4 (смещение по двойному слову, где двойное слово = 4 байта).

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

В первом случае в DI будет помещено значение нулевого элемента массива A, а во втором адрес элемента A (значение метки (имя элемента массива) в памяти).