МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования

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

ЛАБОРАТОРНАЯ РАБОТА №4

Отчёт о практике

студента 2 курса 251 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Тюменцева Радомира Александровича

Проверено:

Старший преподаватель _____ Е. М. Черноусова

Саратов 2025

1 Задание 4

Вариант 1. Массив из 20 чисел заполнить последовательностью, состоящей наполовину из чётных чисел (2, 4, …) и наполовину из квадратов этих чисел; организовать вывод массива на экран в виде таблицы 2x10 с фиксированной шириной столбцов:

```
 2   4   6   8  10  12  14  16  18  20
 4  16  36  64 100 144 196 256 324 400
```

Рис. 1 – Вывод в виде таблицы 2x10 массива из 20 чисел.

1.1 Краткий словестный алгоритм программы

1. Вывод фамилии, имени и номера группы.
2. Заполнение в цикле первых 10 элементов массива `simple` чётными числами: от 2 до 20.
3. Вычисление квадратов этих чисел и заполнение вторых 10 элементов массива `simple` квадратами чётных чисел: от 4 до 400.
4. Перевод первых 10 элементов массива `simple` в символы ASCII и вывод первой строки чётных чисел.
5. Перевод первых 10 элементов массива `simple` в символы ASCII и вывод второй строки квадратов чётных чисел.
6. Завершение работы программы.

1.2 Текст программы на языке ассемблера с комментариями

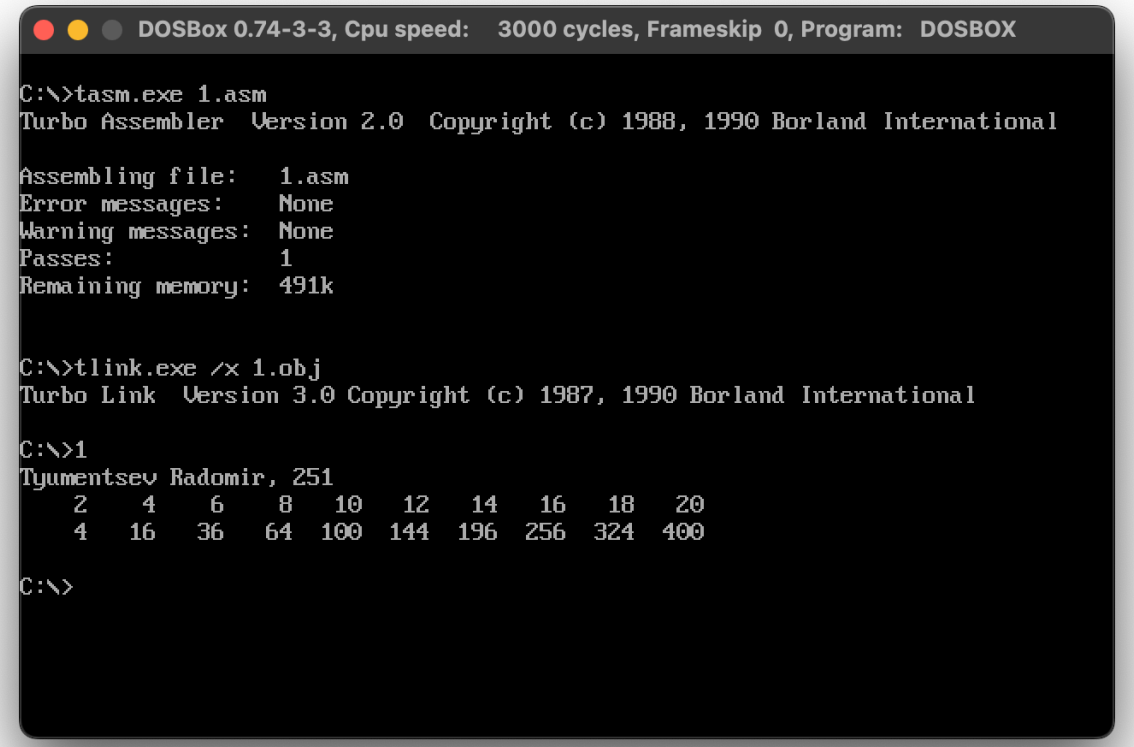
```
.model small
.stack 100h
.186

.data
  my_name db "Tyumentsev Radomir, 251$"
  simple dw 20 dup (?) ; Массив из 20 слов (неинициализированный)
  result db 5 dup (' '), '$' ; Буфер для вывода одного числа: 5 символов + '$'
  nl     db 0AH, 0Dh, '$' ; Символы перехода на новую строку

.code
start:
  mov AX, @data
  mov DS, AX

  ;   Вывод фамилии, имени и номера группы
  lea DX, my_name
  mov AH, 09h
```

```
  int 21h

  lea DX, nl
  mov AH, 09h
  int 21h

  mov CX, 10; Количество чисел
  mov BX, 2; Первое число
  mov SI, 0; Смещение в байтах (0)

fill_even_loop:
  mov  simple[SI], BX
  add  SI, 2
  add  BX, 2
  loop fill_even_loop

  ;   Заполнение второй половины массива квадратами
  mov CX, 10
  mov BX, 2
  mov SI, 20; Смещение к 11-му элементу (10 слов * 2 байта)

fill_sq_loop:
  mov  AX, BX
  mul  BX; AX = BX*BX
  mov  simple[SI], AX
  add  SI, 2
  add  BX, 2
  loop fill_sq_loop

  ;   Вывод первой строки (чётные числа)
  mov CX, 10
  mov SI, 0; Начинаем с первого элемента

print_first_row:
  mov  AX, simple[SI]
  mov  BX, 10
  call word_asc
  mov  AH, 9
  lea  DX, result
  int  21h
  add  SI, 2
  loop print_first_row

  ;   Перенос строки
  mov AH, 9
  lea DX, nl
  int 21h

  ;   Вывод второй строки (квадраты)
  mov CX, 10
```

```
69    mov SI, 20; Начинаем с 11-го элемента
70
71  print_second_row:
72    mov  AX, simple[SI]
73    mov  BX, 10
74    call word_asc
75    mov  AH, 9
76    lea  DX, result
77    int  21h
78    add  SI, 2
79    loop print_second_row
80
81    ;   Перевод строки
82    mov AH, 9
83    lea DX, nl
84    int 21h
85
86    ;   Завершение программы
87    mov AX, 4C00h
88    int 21h
89
90    word_asc proc
91      pusha
92      mov SI, 0; Смещение в байтах, изначально 0, затем увеличивается до 5
93      mov CX, 5; Длина строки result (5 символов)
94
95    ; Заполняем буфер пробелами для очистки
96    fill_spaces:
97      mov  result[SI], ' '
98      inc  SI
99      loop fill_spaces
100
101   convert_loop:
102     dec SI
103     mov DX, 0
104     div BX; AL = частное, AH = остаток
105     add DL, '0'
106     mov result[SI], DL
107     cmp AX, 0
108     jne convert_loop
109
110     popa
111     ret
112   word_asc endp
113
114 end start
```

Текст программы

1.3 Скриншоты запуска программ



Запуск программы

2 Ответы на контрольные вопросы

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

Для выделения памяти существуют директивы db, dw, dd, dq, dt, выделяющие, в зависимости от конкретной директивы, байт, слово, двойное слово, и так далее. К ним применим оператор dup, позволяющий выделить несколько экземпляров:

```
array db 20 dup (?)
```

При использовании `(?)` ячейки выделенной памяти не будут инициализироваться конкретными значениями.

В результате выполнения приведённой в примере директивы будет выделено 20 байт памяти с начальным адресом под меткой `array`.

Перечислив значения через запятую, можно проинициализировать конкретные байты, слова, и так далее, соответственно.

2. Опишите команды умножения на байт и на слово.

Для умножения значения регистра AL или AX на число используется инструкция `mul` для беззнаковых чисел или `imul` для знаковых:

```
1 mul <второй множитель>
2 mul BL; AX = AL * BL
3 mul BX; DX:AX = AX * BX
```

Команда умножения может устанавливать только флаги переноса и переполнения.

Команда `mul` устанавливает оба флага, если старшая половина результата не нулевая. Если умножаются два байта, установка флагов переполнения и переноса показывает, что результат умножения больше 255 и не может содержаться в одном байте. В случае умножения слов флаги устанавливаются, если результат больше 65535.

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

Число $255 = 2^8 - 1$.

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?

Прямая адресация с индексированием. `array` определяет адрес начала массива, а значение в SI - индекс элемента, прибавляющийся к адресу (смещение).

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

Задаётся CX, равный количеству элементов массива, а затем в цикле постепенно увеличивается SI на размер элемента массива (байт, слово или двойное слово, соответственно), и происходит постепенное смещение по элементам.

```
1 mov CX, 5; Если в массиве 5 элементов
2 mov SI, 0
3 array_loop:
4     mov AX, A[SI]
5     add SI, 2; Для слова
6     loop array_loop
```

6. Для некоторого массива array, объявленного следующим образом: array DW 20 DUP(?), каким будет результат выполнения команды mov DI, array и команды mov DI, offset array?

В первом случае в DI будет помещено значение первого элемента массива `array`, а во втором - адрес начала массива `array`.