МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ «ПОЛОЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ ЕВФРОСИНИИ ПОЛОЦКОЙ»

Факультет информационных технологий Кафедра технологий программирования

**Отчет по выполнению лабораторной работы №2**

**по курсу «Конструирование программного обеспечения»**

«Работа с массивами в языке низкого уровня»

Выполнил: студент группы 22-ИТ-зд

Шастовская М.С.

Проверил:

Скуковская А.А.

Полоцк, 2024

**Цель:** целью работы является приобретение навыков использования массивов при программировании на языке ассемблера.

# Задание 1

# Подсчитать в массиве количество элементов, равных введённому N;

.model small

.stack 100h

.data

array db 1, 3, 4, 6, 4, 8, 9, 4, 5

n db 4

count db 0

.code start:

mov ax, @data mov ds, ax

mov cx, 9 ; количество элементов в массиве mov si, offset array ; адрес начала массива

mov al, n ; загрузить значение N в AL mov bl, 0 ; сбросить счетчик

mov dx, cx ; сохранить количество элементов в DX

loop\_start:

cmp byte ptr [si], al ; сравнить текущий элемент с N

jne loop\_end ; если элемент не равен N, перейти к концу цикла add bl, 1 ; увеличить счетчик на 1

loop\_end:

inc si ; увеличить указатель массива на размер элемента loop loop\_start ; перейти к следующему элементу

; сохранить результат в переменной count mov count, bl

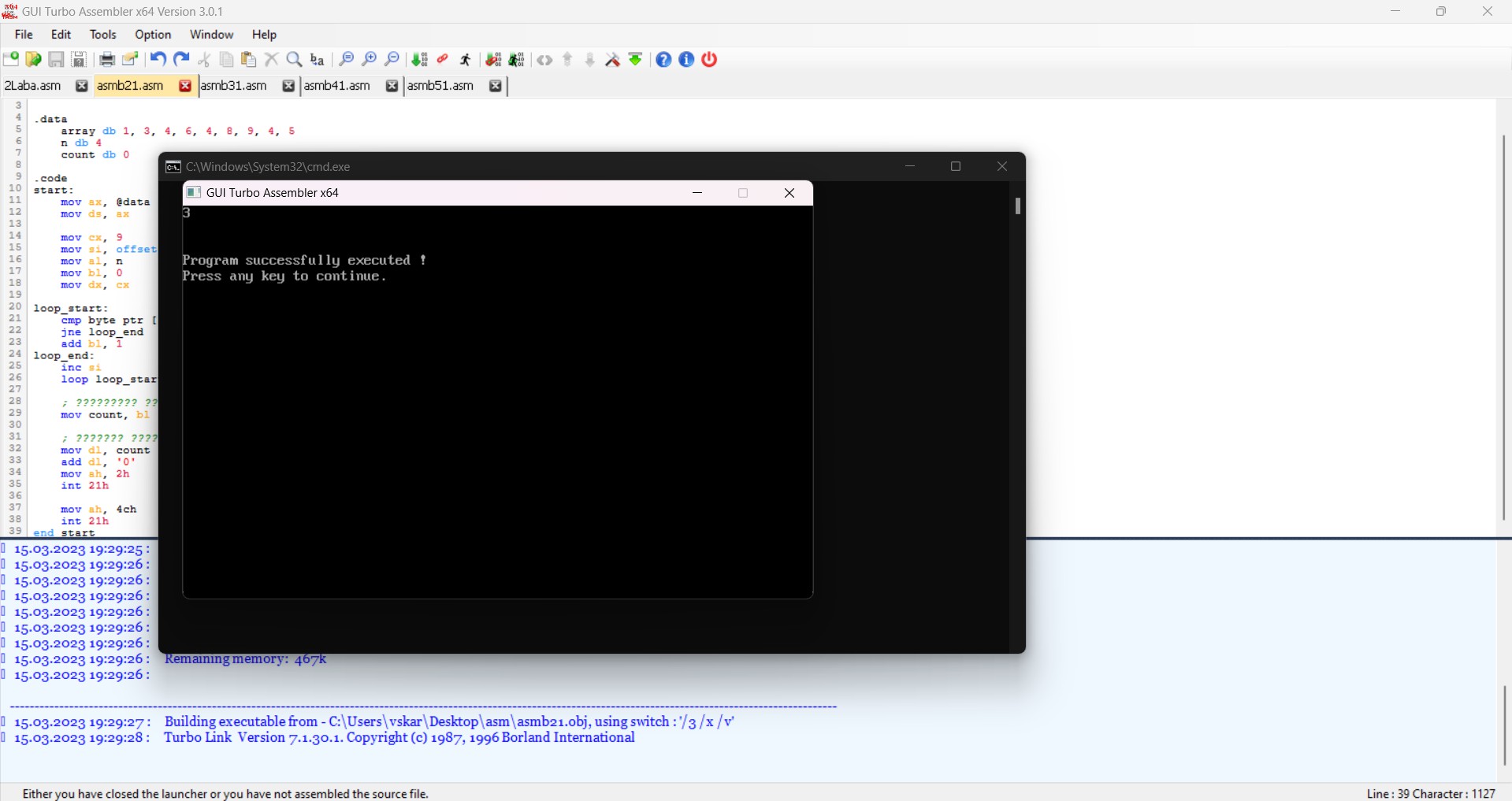
; вывести результат

mov dl, count ; загрузить значение счетчика в DL add dl, '0' ; преобразовать число в символ

mov ah, 2h ; системный вызов для вывода символа int 21h

mov ah, 4ch ; завершение программы

int 21h end start



# Задание 2

# Заменить в массиве все четные элементы нулями;

.model small

.stack 100h

.data

array db 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

count equ 10

.code start:

mov ax, @data mov ds, ax

mov cx, count lea si, array xor bx, bx

loop\_start:

mov al, [si]

test al, 1 ; проверка на нечетность jnz odd\_number

mov byte ptr [si], 0 ; замена на ноль jmp next\_number

odd\_number: inc bx

next\_number:

inc si

loop loop\_start

; вывод массива mov cx, count

lea si, array

mov ah, 02h ; настройка прерывания для вывода символа print\_loop:

mov dl, [si]

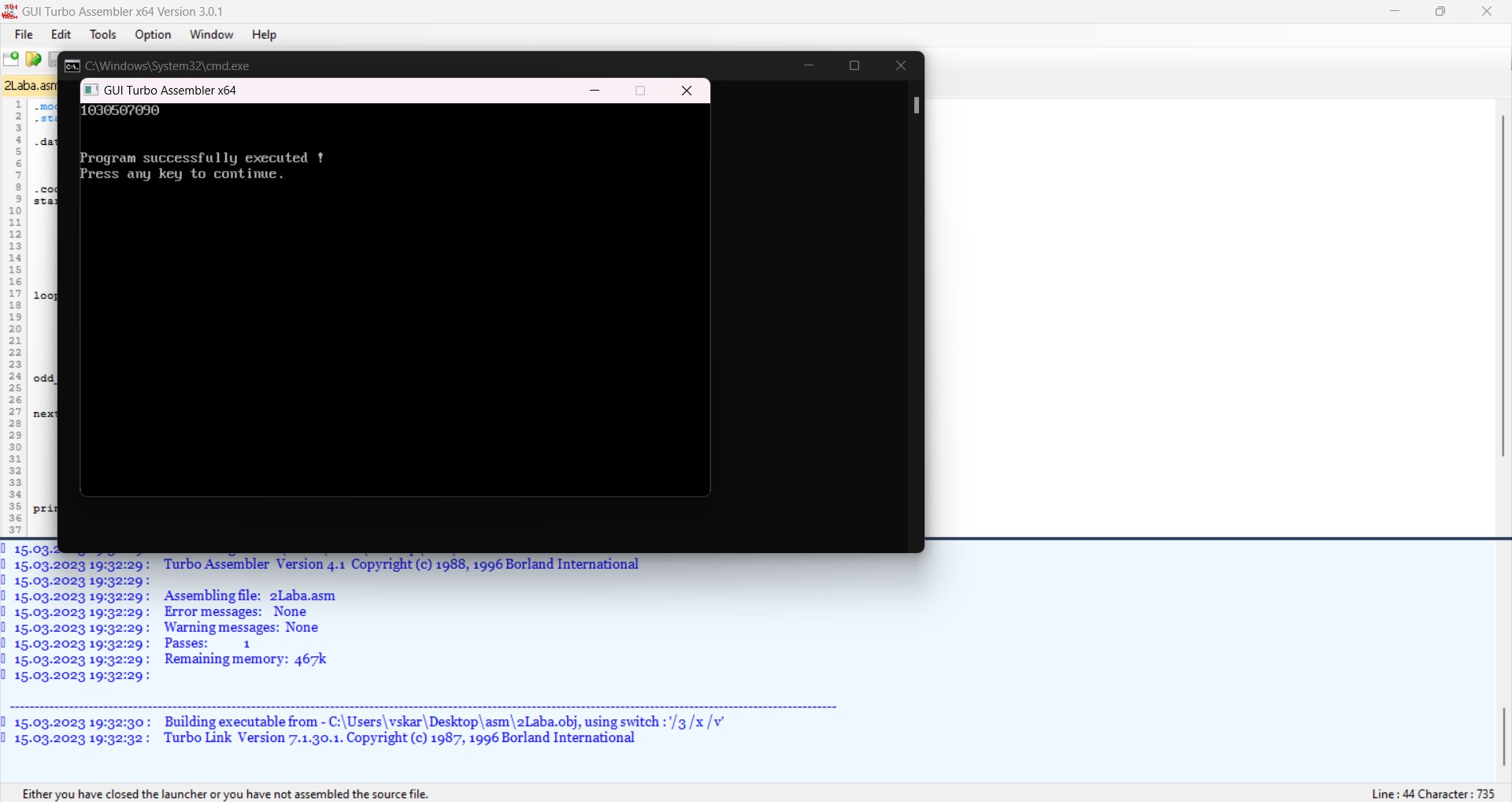
add dl, '0' ; преобразование числа в символ int 21h

inc si

loop print\_loop

mov ah, 4Ch int 21h

end start



# Задание 3

# Определить номер первого равного нулю элемента;

.model small

.stack 100h

.data

array db 3, 8, 0, 6, 4, 0, 9, 2, 5

count equ 9

.code start:

mov ax, @data mov ds, ax

mov cx, count ; количество элементов в массиве

mov si, offset array ; адрес начала массива

mov bx, 0 ; номер первого равного нулю элемента search\_loop:

cmp byte ptr [si], 0 ; сравнить текущий элемент с нулем

jz found\_zero ; если элемент равен нулю, перейти к найденному метке inc bx ; увеличить номер текущего элемента

add si, 1 ; увеличить указатель массива на размер элемента loop search\_loop ; перейти к следующей итерации цикла

not\_found:

mov ah, 4Ch ; функция завершения программы int 21h

found\_zero:

mov bx, bx ; сохранить номер элемента с нулем в регистре BX

; здесь можно выполнить дополнительную обработку найденного элемента

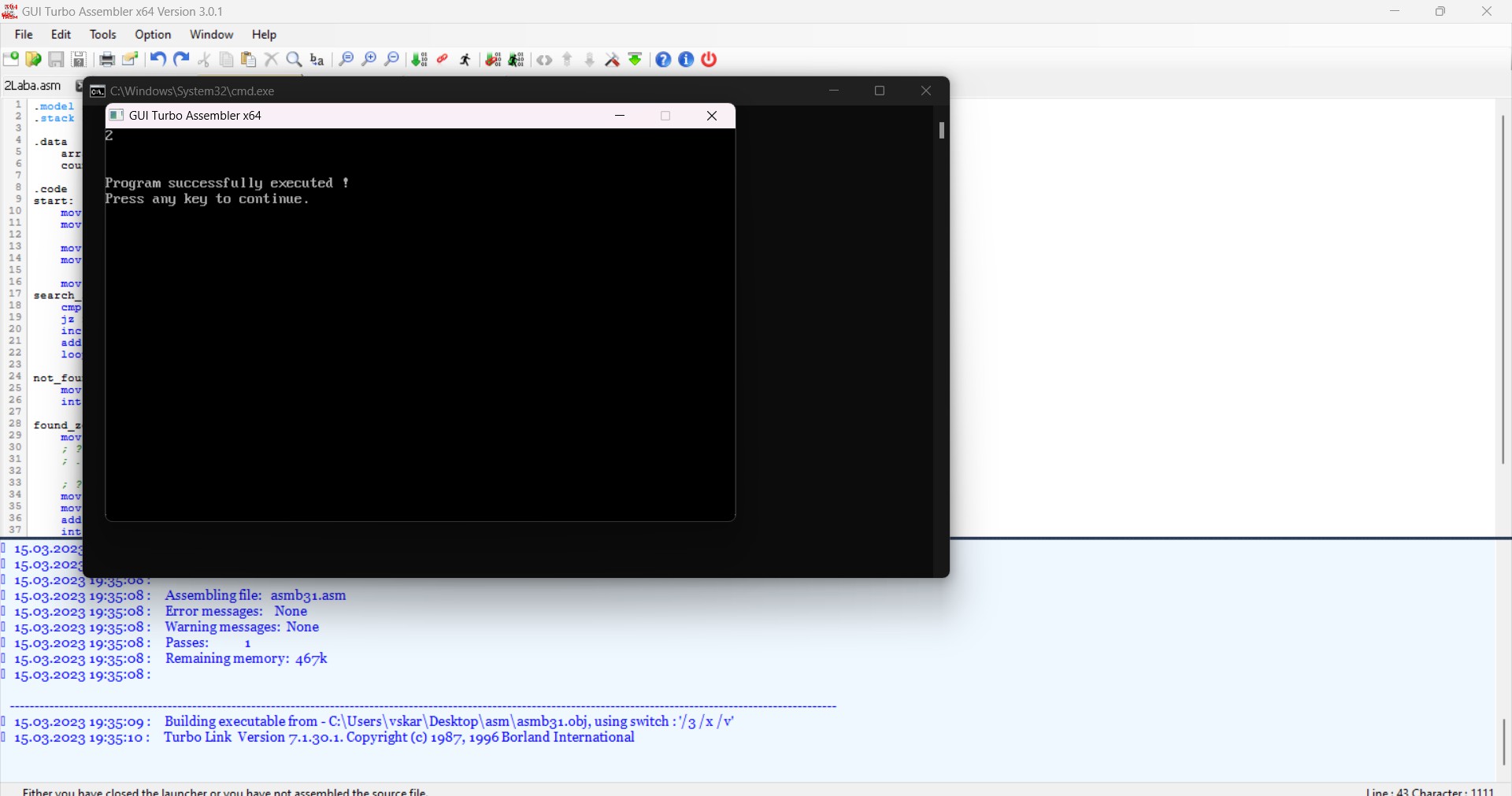
; ...

; вывести номер найденного элемента mov ah, 02h

mov dl, '0' add dl, bl int 21h

; завершить программу mov ah, 4Ch

int 21h end start



# Задание 4

# Вывести на экран все нечетные элементы массива;

.model small

.stack 100h

.data

array db 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

count equ 10

.code start:

mov ax, @data mov ds, ax

mov cx, count ; количество элементов в массиве mov si, offset array ; адрес начала массива

print\_loop:

mov al, byte ptr [si] ; загрузить текущий элемент в AL test al, 1 ; проверить, является ли он нечетным

jz skip\_print ; если элемент четный, перейти к следующему add al, '0' ; преобразовать число в символ

mov dl, al ; поместить символ в DL

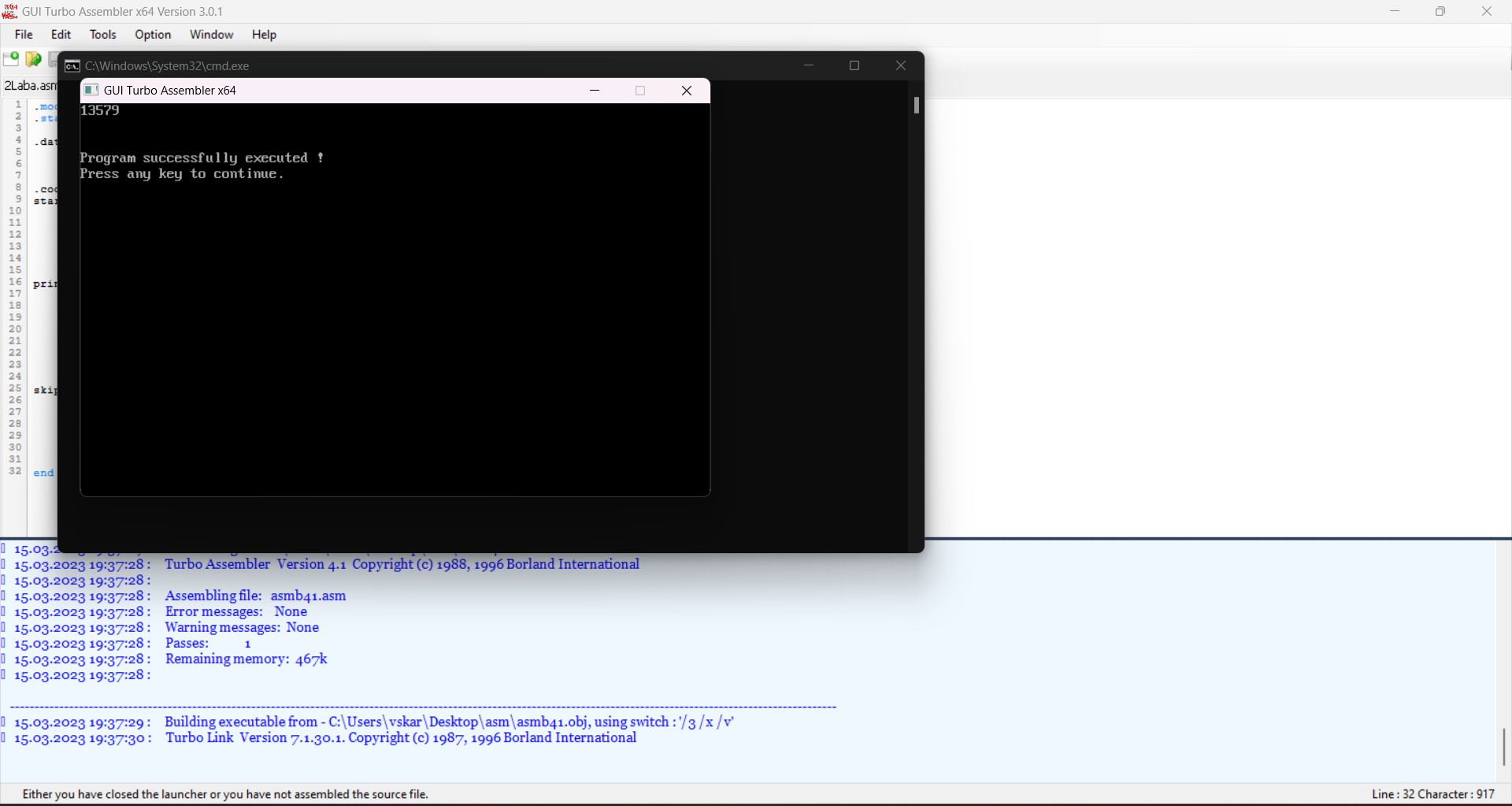
mov ah, 02h ; функция вывода символа int 21h ; вывод символа на экран

skip\_print:

add si, 1 ; увеличить указатель массива на размер элемента loop print\_loop ; перейти к следующему элементу

mov ah, 4Ch ; функция завершения программы int 21h

end start



# Задание 5

Отсортировать массив по возрастанию, используя любой известный

алгоритм сортировки

.model small

.stack 100h

.data

array db 1, 6, 3, 8, 2, 9, 4, 5, 10, 7

count equ 10

.code start:

mov ax, @data mov ds, ax

mov cx, count

mov si, offset array

sort:

mov bx, si add bx, 1

inner\_loop:

mov al, [si] cmp al, [bx]

jbe next\_iteration

xchg al, [bx] mov [si], al

next\_iteration: add bx, 1

cmp bx, count jne inner\_loop

add si, 1 loop sort

; вывод массива mov cx, count

lea si, array

mov ah, 02h ; настройка прерывания для вывода символа print\_loop:

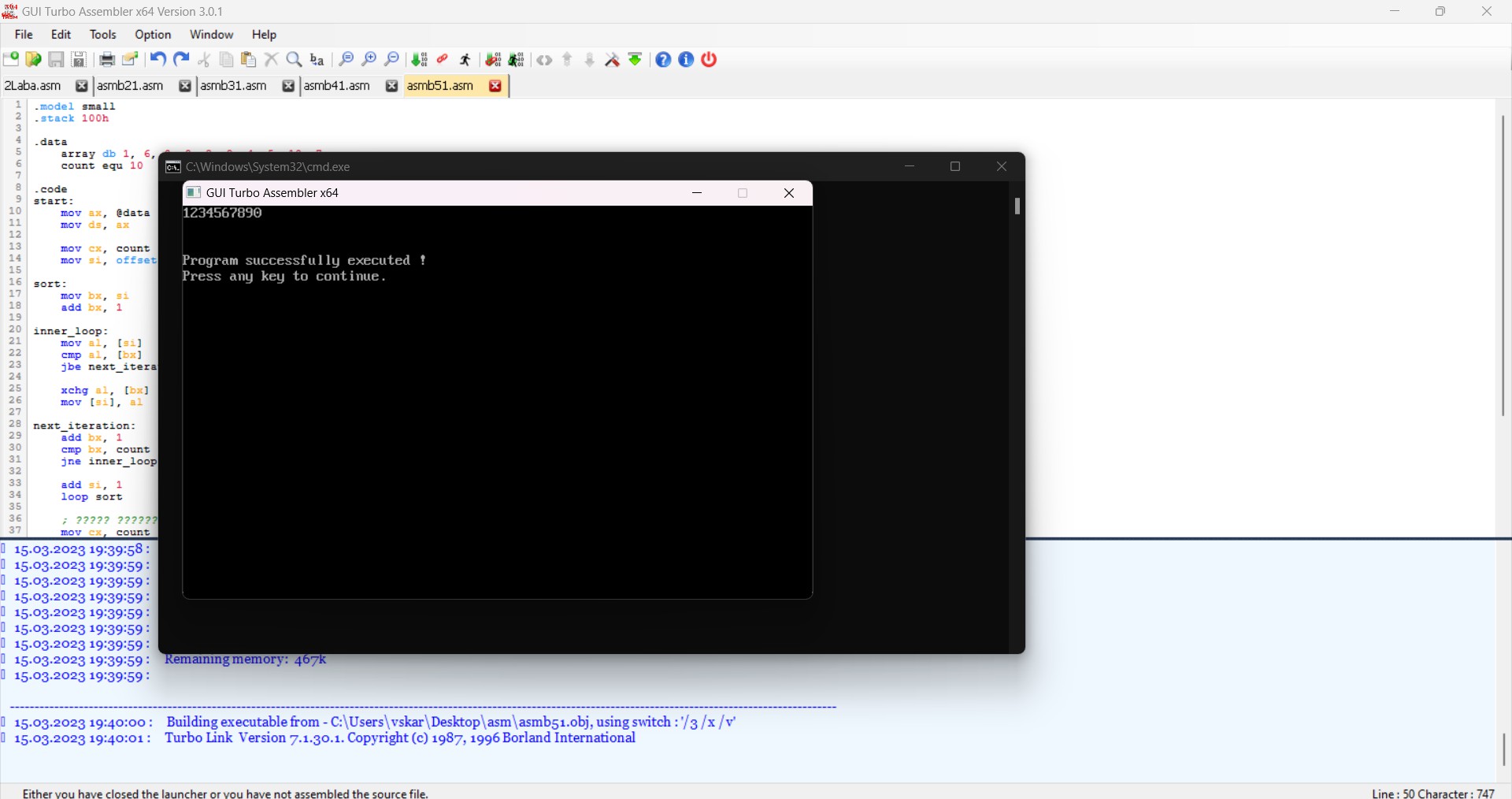
mov dl, [si]

add dl, '0' ; преобразование числа в символ int 21h

inc si

loop print\_loop

mov ah, 4Ch int 21h

end start

**Вывод:** приобрела навыки использования массивов при программировании на языке ассемблера.