Министерство науки и высшего образования Российской Федерации  
Пензенский государственный университет  
Кафедра вычислительная техника

**ОТЧЕТ**

по лабораторной работе №4

по дисциплине «ЭВМ и переферийные устройства»

на тему «Работа процессора с подпрограммами»

Выполнили:

студенты группы 22ВВП1

Хоссейни Нежад С.А.С.М.

Захаров А. С.

Приняли:

Никишин К.И.

Патунин Д.В.

Пенза 2025

**Цель работы**

Изучение особенностей выполнения команд работы с подпрограммами и передачи параметров в подпрограммы.

**Задание**

**Задание1:** Составить программу, выполняющую обработку массива, аналогично заданию предыдущей работы, но оформленной в виде "ближней" процедуры, параметры в которую передаются через регистры.

**Задание2:** Составить программу, выполняющую обработку массива, аналогично заданию предыдущей работы, но оформленной в виде "ближней" процедуры, параметры в которую передаются через стек.

****

**Ход работы**

**Задание 1**

**Листинг:**

stseg segment para stack

dw 16 dup(?)

stseg ends

dseg segment para

x dw 1, 0, 1, 2, 3, 0, 4, 5

y dw ?

dseg ends

cseg segment para

main proc far

; Инициализация сегмента данных

assume cs:cseg, ds:dseg, ss:stseg

push ds

mov ax, dseg

mov ds, ax

mov bx, offset x

mov cx, 8

; Инициализируем y значением 0

mov ax, 0

mov y, ax

call set\_y\_to\_one

end\_proc:

pop ds

mov ax, 4c00h

int 21h

main endp

is\_there\_null proc near

loop\_start:

mov dx, [bx] ; Загружаем текущий элемент массива

cmp dx, 0

je set\_y\_to\_one ; Если элемент равен 0, то переходим в set\_y\_to\_one

add bx, 2 ; Переход к следующему элементу (одно слово = 2 байта)

loop loop\_start ; Уменьшаем CX и повторяем цикл, если CX не равен 0

ret ; Возврат в основную программу

set\_y\_to\_one:

mov ax, 1

mov y, ax ; Присваиваем y значение 1

ret ; Возврат в основную программу

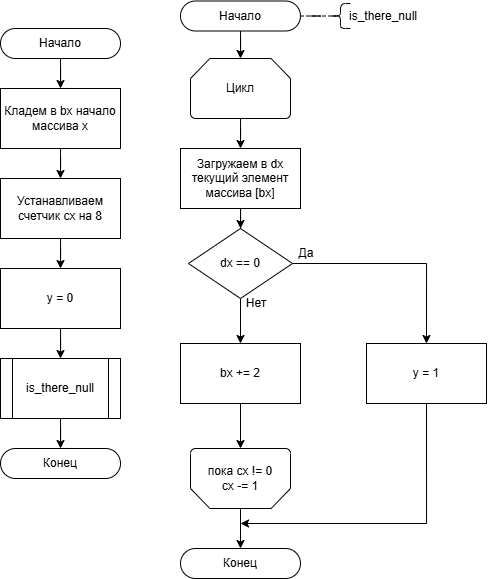
is\_there\_null endp

cseg ends

end main

**Отладка**

**Блок-схема**

****

**Задание 2**

**Листинг**

stseg segment para stack

dw 16 dup(?)

stseg ends

dseg segment para

x dw 1, 0, 1, 2, 3, 0, 4, 5

y dw ?

dseg ends

cseg segment para

main proc far

; Инициализация сегмента данных

assume cs:cseg, ds:dseg, ss:stseg

push ds

mov ax, dseg

mov ds, ax

; Сначала push второй параметр (число элементов), затем первый (адрес массива)

push 8

push offset x

; Инициализация переменной y значением 0

mov ax, 0

mov y, ax

call is\_there\_null

end\_proc:

pop ds

mov ax, 4c00h

int 21h

main endp

is\_there\_null proc near

push bp

mov bp, sp

; Извлечение параметров из стека

mov bx, [bp+4] ; адрес массива

mov cx, [bp+6] ; число элементов массива

loop\_start:

mov dx, [bx] ; загружаем текущий элемент массива

cmp dx, 0

je set\_y\_to\_one ; если элемент равен 0, то переходим в set\_y\_to\_one

add bx, 2 ; переход к следующему элементу (одно слово = 2 байта)

loop loop\_start ; уменьшаем CX и повторяем цикл, если CX не равен 0

jmp done

set\_y\_to\_one:

mov ax, 1

mov y, ax ; присваиваем переменной y значение 1

done:

pop bp

ret 4 ; возврат из подпрограммы с очисткой параметров (2 параметра по 2 байта)

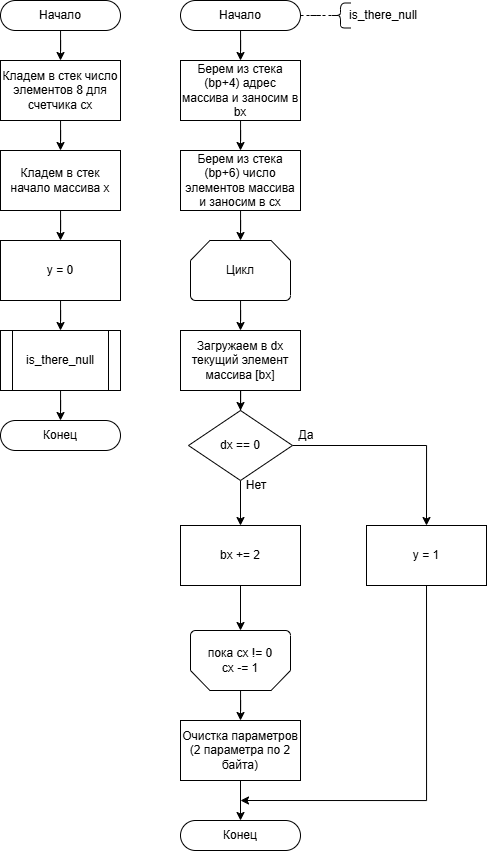
is\_there\_null endp

cseg ends

end main

**Отладка**

**Блок-схема**

****

**Вывод**

Изучили особенности выполнения команд работы с подпрограммами и передачи параметров в подпрограммы.