**ЛАБОРАТОРНАЯ РАБОТА 3**

**Тема:** Организация обработки числовых одномерных массивов.

**Цель:** Изучение команд организации циклов и способов косвенной адресации данных памяти в микропроцессоре *i486*; приобретение практических навыков составления программ обработки одномерных массивов, освоение методов анализа трудоемкости и ресурсной сложности алгоритмов обработки одномерных числовых массивов.

**Задание. Вариант 23.**

1. На основании индивидуального задания (табл. 3.4) составить программу для обработки элементов одномерного массива. Длина элементов исходного массива равна DW. Значения элементов исходного массива задать в сегменте данных (см. пример выше). Длину элементов результирующего массива, если он необходим, выбрать самостоятельно.
2. Получить загрузочный модуль и протестировать выполнение программы.
3. Выполнить расчет времени выполнения программы.

Задание на обработку: Дан массив А[15]. Сдвинуть элементы массива циклическим сдвигом так, чтобы минимальный элемент оказался на первом месте

**Ход работы.**

Определим значения массива A в сегменте данных (рисунок 1).

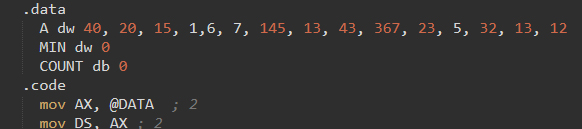


Рисунок 1 - представление массива в сегменте данных

Ниже представлен листинг программы для обработки элементов одномерного массива.

.model small

.stack 100h

.data

A dw 40, 20, 15, 1,6, 7, 145, 13, 43, 367, 23, 5, 32, 13, 12

MIN dw 0

COUNT db 0

.code

mov AX, @DATA ; 2

mov DS, AX ; 2

; Вычисление минимального элемента

MOV CX, 13

xor SI, SI

mov BX, [a + SI]

mov MIN, BX

add SI, 2

LAB: xor BX, BX

mov BX, [a + SI]

add SI, 2

mov AX, MIN

cmp BX, AX

JG LABEX

mov MIN, BX

LABEX:

LOOP LAB

;Перестановка элементов массива до тех пор,

;Пока первый элемент не станет минимальным.

MOV CX, 14

xor BX,BX

LAB2:MOV COUNT, 14;15

xor SI, SI

mov AX, [a + SI]

cmp AX, MIN

JE LAB2EXIT

L1: cmp COUNT, 1

JE L2

mov DI, SI

add SI, 2

mov BX, [a + SI]

mov [a + DI], BX

sub count, 1

MOV DI, 1

cmp DI, 1

JE L1

L2: mov DI, SI

MOV [a + DI], AX

LOOP LAB2

LAB2EXIT:

MOV AH,4Ch ; 4

INT 21h ; 51

END

Протестируем работу программы при помощи TurboDebugger.

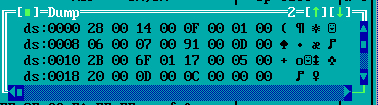


Рисунок 2 - Массив в дампе памяти перед началом работы программы

На рисунке 2 представлен массив A на начало работы программы. Теперь последовательно выполним все действия программы.

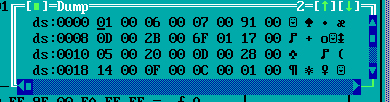


Рисунок 3 - Массив в дампе памяти на момент окончания программы

На рисунке 3 мы можем увидеть массив, элементы которого начинаются с единицы. Единица – минимальный элемент массива, значит программа работает корректно.

Исполнение программы занимает 2415-2807 такт процессора.

**Вывод:** в ходе работы получены базовые навыки обработки одномерных массивов целых чисел, построения алгоритмов с использованием ветвления и множественного перебора элементов массива, вывода массива, а также циклов, условных и безусловных переходов к меткам.