

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«Санкт-Петербургский национальный исследовательский университет информационных  
технологий, механики и оптики»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

**ЛАБОРАТОРНАЯ РАБОТА №3**  
по дисциплине  
**«Основы профессиональной деятельности»**

Вариант № 3496

*Выполнил:*  
Студент группы Р3132  
Юксель Хамза

*Преподаватель:*  
Абузов Ярослав  
Александрович

# Содержание

<u>Текст задания</u> .....	3
<u>Описание программы</u> .....	3
<u>Таблица трассировки</u> .....	5
<u>Вывод</u> .....	6

# Текст задания

По выданному преподавателем варианту восстановить текст заданного варианта программы, определить предназначение и составить описание программы, определить область представления и область допустимых значений исходных данных и результата, выполнить трассировку программы.

2CC:	02E0		2DA:	7EF4
2CD:	A000		2DB:	F901
2CE:	E000		2DC:	EEF2
2CF:	0200		2DD:	82CE
2D0:	+ AF80		2DE:	CEF9
2D1:	0740		2DF:	0100
2D2:	0680		2E0:	F600
2D3:	EEFB		2E1:	FD00
2D4:	AF03		2E2:	72DE
2D5:	EEF8			
2D6:	4EF5			
2D7:	EEF5			
2D8:	ABF4			
2D9:	F203			

Адрес	Код команды	Мнемоника	Комментарии	
2CC	02E0	A	First Element of Array	
2CD	A000	B	POINTER to the last element of array(memory keep decreasing with loop so does array elements)	
2CE	E000	C	THE LOOP COUNTER == 3 (number of the elements in array)	
2CF	0200	R	RESULT	
2D0	AF80	LD #80	Прямая загрузка M -> AC (FF80)	With immediate (direct) operand loading. bits 0-7 as an operand. Direkt olarak accumulatora operandi 0-7 arasındaki değeri verir.
2D1	0740	DEC	AC – 1 ->AC (FF7F)	Adressless.
2D2	0680	SWAB	Изменяет верхний и нижний байты аккумулятора. [AC15-AC8 <->AC7-AC0]	Adressless. AC = FF7F → 7FFF.
2D3	EEFB	ST #FB (IP+1+FB)	(0xE command) → Operand = IP + FB Operand = 2D4 +FB(= -5) = 2CF Прямое отн. сохранение (Очистка ячейки 2CF) AC -> M (2CF)	0xE Direct Relative Addressing Прямое отн. Сохранение (offsets are signed bits) 2CF = 7FFF
2D4	AF03	LD #03	Прямая загрузка M -> AC (0003)	Immediate operand loading 03 > AC
2D5	EEF8	ST #F8 (IP+1+F8)	Прямое относительное сохранение AC -> M (2CE) (IP-8)	AC değeri 2CE ye yazıldı
2D6	4EF5	ADD #F5 #(IP+1+F5)	Прямое относительное сложение M((IP-11)2CC) + AC -> AC	

2D7	EEF5	ST #F5 Signed F5 =(IP+1+F5)	Прямое относительное сохранение AC -> M (2CD)(IP-11)	
2D8	ABF4	LD (IP+1+F4)	Kосвенная автодекрементальная загрузка: (Value(IP-12=2CD)) - 1 -> AC  Decrease the value of cell 2CD by 1. Go to this address and write the value to the accumulator.  2CD de bulunan veri -1 > AC	Indirect auto-decrement. First decrement then writes
2D9	F203	BMI (IP+1+03)	Если N == 1, то IP = IP + 3 + 1 -> IP	Check the negativity
2DA	7EF4	CMP (IP+1+F4)	Subtracts the value of cell (2DB-12=2CF) from the AC, the result sets the flags.	Direct relative addressing
2DB	F901	BGE (IP+1)	If N xor V ==0; IP+1+1-> IP	Большой veya eşitse geç, küçükse normal devam
2DC	EEF2	ST #F2 Signed F2 = (IP+F2+1)	Прямое относительное сохранение AC -> M (2CF) (IP-14).	
2DD	82CE	LOOP (2CE)	Value(2CE) – 1 => (2CE); Если Value(2CE) <= 0, то IP + 1 -> IP	If the checked data 2CE is less than or equal to 0, the following jump section is skipped and the program is completed. If it is negative or 0, continue (to the 2DE).
2DE	CEF9	JUMP (IP+1+F9)	Прямой относительный прыжок (IP)2DF – 7 -> IP (2D8)	Direct relative jump.
2DF	0100	HLT		Останова
2E0	F600	---	M[0]	Arrays
2E1	FD00	---	M[1]	
2E2	72DE	---	M[2]	

## Описание программы

Программа считывает элементы массива (по адресам 2E0-2E2) по одному от конца к началу, учитывает только неотрицательные (т.е. положительные) значения, находит среди них «наименьшее положительное» значение и сохраняет его по адресу 2CF.

## Область представления

- M[0], M[1], M[2] – 16-разрядные знаковые числа
- A fist element address – 11-разрядные беззнаковые числа
- B last element address – 11-разрядные беззнаковые числа
- C (Количество элементов массива) – 8-разрядные беззнаковые числа
- R – 16-разрядные знаковые числа

## Область допустимых значений

- $-2^{15} \leq M[I] \leq 2^{15} - 1$
- $C \in [0, 2^8 - 1]$
- $-2^{15} \leq R \leq 2^{15} - 1$
- $A, B \in [0, 2CC - C] \cup [2E0, 7FF]$  max memory address 7ff

(ОДЗ пишется довольно просто, обычно ограничение упирается лишь в то, что во всей памяти БЭВМ может хранится всего 7FF значений и в ограничение ввода данных. Например, у косвенной и прямой адресации это от 0 до 127.)

## Расположение данных в памяти

- 2CC - 2CE, 2E0 – 2E2 – исходные данные;
- 2CF – результат;
- 2D0 – 2DF – команды.

## Адреса первой и последней выполняемой команды

- Адрес первой команды: 2D0
- Адрес последней команды: 2DF

## Таблица трассировки

Адр	Знчн	IP	CR	AR	DR	SP	BR	AC	NZVC	Адр	Знчн
2D0	AF80	2D1	AF80	2D0	FF80	000	FF80	FF80	1000	-	-
2D1	0740	2D2	0740	2D1	0740	000	02D1	FF7F	1001	-	-
2D2	0680	2D3	0680	2D2	0680	000	02D2	7FFF	0001	-	-
2D3	EEFB	2D4	EEFB	2CF	7FFF	000	FFF8	7FFF	0001	2CF	7FFF
2D4	AF03	2D5	AF03	2D4	0003	000	0003	0003	0001	-	-
2D5	EEF8	2D6	EEF8	2CE	0003	000	FFF8	0003	0001	2CE	0003
2D6	4EF5	2D7	4EF5	2CC	02E0	000	FFF5	02E3	0000	-	-
2D7	EEF5	2D8	EEF5	2CD	02E3	000	FFF5	02E3	0000	2CD	02E3
2D8	ABF4	2D9	ABF4	2E2	72DE	000	FFF4	72DE	0000	-	-
2D9	F203	2DA	F203	2D9	F203	000	02D9	72DE	0000	-	-
2DA	7EF4	2DB	7EF4	2CF	7FFF	000	FFF4	72DE	1000	-	-

<b>Адр</b>	<b>Знчн</b>	<b>IP</b>	<b>CR</b>	<b>AR</b>	<b>DR</b>	<b>SP</b>	<b>BR</b>	<b>AC</b>	<b>NZVC</b>	<b>Адр</b>	<b>Знчн</b>
2DB	F901	2DC	F901	2DB	F901	000	02DB	72DE	1000	-	-
2DC	EEF2	2DD	EEF2	2CF	72DE	000	FFF2	72DE	1000	2CF	72DE
2DD	82CE	2DE	82CE	2CE	0002	000	0001	72DE	1000	-	-
2DE	CEF9	2D8	CEF9	2DE	02D8	000	FFF9	72DE	1000	-	-
2D8	ABF4	2D9	ABF4	2E1	FD00	000	FFF4	FD00	1000	2CD	02E1
2D9	F203	2DD	F203	2D9	F203	000	0003	FD00	1000	-	-
2DD	82CE	2DE	82CE	2CE	0001	000	0000	FD00	1000	2CE	0001
2DE	CEF9	2D8	CEF9	2DE	02D8	000	FFF9	FD00	1000	-	-
2D8	ABF4	2D9	ABF4	2E0	F600	000	FFF4	F600	1000	2CD	02E0
2D9	F203	2DD	F203	2D9	F203	000	0003	F600	1000	-	-
2DD	82CE	2DF	82CE	2CE	0000	000	FFFF	F600	1000	2CE	0000
2DF	0100	2E0	0100	2DF	0100	000	02DF	F600	1000	-	-

## Вывод

Во время выполнения лабораторной работы я научился работать в БЭВМ с массивами, ветвлением и циклами. Я изучил прямую и косвенную адресацию и цикл выполнения таких команд, как LOOP и JUMP.