

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 № 15311

***Выполнил:***

Студент группы Р3115

Барсуков Максим

Андреевич

***Преподаватель:***

Абузов Ярослав

Александрович

# Содержание

Текст задания .....	3
Описание программы.....	4
Таблица трассировки .....	5
Вывод.....	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.

455:	046C	463:	0380
456:	A000	464:	0200
457:	E000	465:	0280
458:	0200	466:	2EF1
459: +	0200	467:	0400
45A:	EEFD	468:	EEEE
45B:	AF03	469:	8457
45C:	EEFA	46A:	CEF4
45D:	4EF7	46B:	0100
45E:	EEF7	46C:	1000
45F:	ABF6	46D:	845D
460:	0480	46E:	0740
461:	F401		
462:	0480		

Адрес	Код команды	Мнемоника	Комментарии
455	046C	-- arr_first_elem	Адрес первого элемента
456	A000	-- arr_last_elem	Адрес текущего элемента (начиная с последнего)
457	E000	-- arr_length	Количество элементов массива
458	0200	-- result	Результат
459	0200	CLA	Очистка аккумулятора
45A	EEFD	ST IP-3	Прямое отн. сохранение (Очистка ячейки 458) AC -> M (458)
45B	AF03	LD #3	Прямая загрузка 0003 -> AC
45C	EEFA	ST IP-6	Прямое относительное сохранение AC -> M (457)
45D	4EF7	ADD IP-9	Прямое относительное сложение M(455) + AC -> AC
45E	EEF7	ST IP-9	Прямое относительное сохранение AC -> M (456)
45F	ABF6	LD -(IP-10)	Косвенная автодекрементальная загрузка: 3n(456) -= 1; 3n(456) -> A
460	0480	ROR	Циклический сдвиг вправо
461	F401	BCS IP+1	Если C == 1, то IP = IP + 1 + 1 -> IP
462	0480	ROR	Циклический сдвиг вправо
463	0380	CMC	Инверсия флага C: ^C -> C
464	0200	CLA	Очистка аккумулятора
465	0280	NOT	^AC -> AC
466	2EF1	AND IP-15	Прямое относительное логическое И: M(458) & AC -> AC
467	0400	ROL	Циклический сдвиг влево
468	EEEE	ST IP-17	Прямое относительное сохранение AC -> M (458)
469	8457	LOOP 457	M(457) - 1 -> M(457); Если M(457) <= 0, то IP + 1 -> IP
46A	CEF4	JUMP IP-12	Прямой относительный прыжок IP - 12 + 1 -> IP ()
46B	0100	HLT	Останов
46C	1000	--	Элементы массива
46D	845D	--	
46E	0740	--	

# Описание программы

Программа находит элементы массива, которые делятся на 4 и сохраняет информацию о них в биты ячейки результата. Формула результата:

$$\text{MEM}(458) = \sum_{i=0}^{\text{MEM}(457)} \begin{cases} 2^i & \text{если } \text{MEM}(46C + i) \div 4 \\ 0 & \text{если } \text{MEM}(46C + i) \nmid 4 \end{cases}$$

## Область представления

- arr\_first\_elem, arr\_last\_elem, arr\_length, result – 16-ти разрядные целые числа в прямом коде.
- arr[i] – 16-ти разрядные целые числа в дополнительном коде

## Область допустимых значений

- arr\_length  $\in [1; 16]$  (т. к. при arr\_length > 16 битов результата не будет хватать для данных о делимости элементах и он будет ошибочным)
- result  $\in [0; 2^{16} - 1]$  (т. к. max =  $1 + 2 + 4 + \dots + 2^{15}$ )
- arr\_first\_elem  $\in [0; 455 - \text{arr\_length}] \cup [46B; 7FF - \text{arr\_length}]$
- arr\_last\_elem  $\in [\text{arr\_first\_elem}; \text{arr\_first\_elem} + \text{arr\_length} - 1]$
- Элементы массива arr[i]  $\in [-32768; 32767]$  (т. е.  $[-2^{15}; 2^{15}-1]$ )

## Расположение данных в памяти

- 455, 457, 46C, 46D, 46E – исходные данные;
- 456 – промежуточный результат;
- 458 – итоговый результат;
- 459 – 46B – команды.

## Адреса первой и последней выполняемой команды

- Адрес первой команды: 459
- Адрес последней команды: 46B

# Таблица трассировки

## Значения:

Arr[0] = 0x1000, Arr[1] = 0x845D, Arr[2] = 0x0740, arr\_length = 3, arr\_first\_elem = 0x046C

Адр	Знчн	IP	CR	AR	DR	SP	BR	AC	PS	NZVC	Адр	Знчн
459	200	459	0	0	0	0	0	0	4	100		
459	200	45A	200	459	200	0	459	0	4	100		
45A	EEFD	45B	EEFD	458	0	0	FFFD	0	4	100	458	0
45B	AF03	45C	AF03	45B	3	0	3	3	0	0		
45C	EEFA	45D	EEFA	457	3	0	FFFA	3	0	0	457	3
45D	4EF7	45E	4EF7	455	046C	0	FFF7	046F	0	0		
45E	EEF7	45F	EEF7	456	046F	0	FFF7	046F	0	0	456	046F
45F	ABF6	460	ABF6	46E	740	0	FFF6	740	0	0	456	046E
460	480	461	480	460	480	0	460	03A0	0	0		
461	F401	462	F401	461	F401	0	461	03A0	0	0		
462	480	463	480	462	480	0	462	01D0	0	0		
463	380	464	380	463	380	0	463	01D0	1	1		
464	200	465	200	464	200	0	464	0	5	101		
465	280	466	280	465	280	0	465	FFFF	9	1001		
466	2EF1	467	2EF1	458	0	0	FFF1	0	5	101		
467	400	468	400	467	400	0	467	1	0	0		
468	EEEE	469	EEEE	458	1	0	FFEF	1	0	0	458	1
469	8457	46A	8457	457	2	0	1	1	0	0	457	2
46A	CEF4	45F	CEF4	46A	045F	0	FFF4	1	0	0		
45F	ABF6	460	ABF6	46D	845D	0	FFF6	845D	8	1000	456	046D
460	480	461	480	460	480	0	460	422E	3	11		
461	F401	463	F401	461	F401	0	1	422E	3	11		
463	380	464	380	463	380	0	463	422E	2	10		
464	200	465	200	464	200	0	464	0	4	100		
465	280	466	280	465	280	0	465	FFFF	8	1000		
466	2EF1	467	2EF1	458	1	0	FFF1	1	0	0		
467	400	468	400	467	400	0	467	2	0	0		
468	EEEE	469	EEEE	458	2	0	FFEF	2	0	0	458	2
469	8457	46A	8457	457	1	0	0	2	0	0	457	1
46A	CEF4	45F	CEF4	46A	045F	0	FFF4	2	0	0		
45F	ABF6	460	ABF6	46C	1000	0	FFF6	1000	0	0	456	046C
460	480	461	480	460	480	0	460	800	0	0		
461	F401	462	F401	461	F401	0	461	800	0	0		
462	480	463	480	462	480	0	462	400	0	0		
463	380	464	380	463	380	0	463	400	1	1		
464	200	465	200	464	200	0	464	0	5	101		
465	280	466	280	465	280	0	465	FFFF	9	1001		
466	2EF1	467	2EF1	458	2	0	FFF1	2	1	1		
467	400	468	400	467	400	0	467	5	0	0		
468	EEEE	469	EEEE	458	5	0	FFEF	5	0	0	458	5
469	8457	46B	8457	457	0	0	FFFF	5	0	0	457	0
46B	100	46C	100	46B	100	0	046B	5	0	0		

## 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.