

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
Национальный исследовательский университет ИТМО

МЕГАФАКУЛЬТЕТ ТРАНСЛЯЦИОННЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ПРОГРАММИРОВАНИЯ

ЛАБОРАТОРНАЯ РАБОТА №4
По дисциплине «Архитектура ЭВМ»
Исследование работы ЭВМ при выполнении комплекса программ

Выполнил Фадеев Артём Владимирович
(Фамилия Имя Отчество)
Проверил _____
(Фамилия Имя Отчество)

Санкт-Петербург, 2020 г.

Текст исходной программы:

Адрес	Код команды	Мнемоника	Комментарии
00A	0010	ISZ 010	Link to numbers
00B	0000	-	-
00C	0000	-	-
00D	0000	-	-
00E	0000	-	-
00F	0000	-	-
010	8080	-	Number1
011	ABDA	-	Number2
012	630D	-	Number3
013	71BD	-	Number4
014	FFFC	-	Loop counter
015	0000	-	answer
016	F200	CLA	A = 0
017	480A	ADD (00A)	↔ (80A) --> (00A) -> A += ((00A)), (00A)++;
018	A01A	BMI 01A	if (A < 0) CK = 01A
019	2045	JSR 045	045 = (CK), CK = 045 + 1 В 045 будет храниться 01A - следующая команда для выполнения при выходе.
01A	0014	ISZ 014	(01A)++, if ((01A) >= 0) CK++;
01B	C016	BR 016	CK = 016
01C	F000	HLT	End
01D	0000	-	-
045	0000	-	Адрес для выхода из подпрограммы
046	F200	CLA	A = 0
047	F800	INC	A++
048	4015	ADD 015	A += (015)
049	3015	MOV 015	(015) = A
04A	C845	BR (045)	CK = (045)

Таблица трассировки:

Команда		Содержимое регистров после выполнения программы						Измененная ячейка	
Адрес	Код	СК	РА	РК	РД	А	С	Адрес	Новый код
016	F200	017	016	F200	F200	0000	0		
017	480A	018	010	480A	8080	8080	0	00A	0011
018	A01A	01A	018	A01A	A01A	8080	0		
01A	0014	01B	014	0014	FFFD	8080	0	014	FFFD
01B	C016	016	01B	C016	C016	8080	0		
016	F200	017	016	F200	F200	0000	0		
017	480A	018	011	480A	ABDA	ABDA	0	00A	0012
018	A01A	01A	018	A01A	A01A	ABDA	0		
01A	0014	01B	014	0014	FFFE	ABDA	0	014	FFFE
01B	C016	016	01B	C016	C016	ABDA	0		
016	F200	017	016	F200	F200	0000	0		
017	480A	018	012	480A	630D	630D	0	00A	0013
018	A01A	019	018	A01A	A01A	630D	0		
019	2045	046	045	2046	001A	630D	0	045	001A
046	F200	047	046	F200	F200	0000	0		
047	F800	048	047	F800	F800	0001	0		
048	4015	049	015	4015	0000	0001	0		
049	3015	04A	015	3015	0001	0001	0	015	0001
04A	C845	01A	045	C845	001A	0001	0		
01A	0014	01B	014	0014	FFFF	0001	0	014	FFFF
01B	C016	016	01B	C016	C016	0001	0		
016	F200	017	016	F200	F200	0000	0		
017	480A	018	013	480A	71BD	71BD	0	00A	0014
018	A01A	019	018	A01A	A01A	71BD	0		
019	2045	046	045	2046	001A	71BD	0	045	001A
046	F200	047	046	F200	F200	0000	0		
047	F800	048	047	F800	F800	0001	0		
048	4015	049	015	4015	0001	0002	0		
049	3015	04A	015	3015	0002	0002	0	015	0002
04A	C845	01A	045	C845	001A	0002	0		
01A	0014	01C	014	0014	0000	0002	0	014	0000
01C	F000	01D	01C	F000	F000	0002	0		

Описание программы:

- При выполнении программы происходит подсчет количества неотрицательных чисел.
- Количество всех чисел задано в ячейке 014. Если наше число удовлетворяет, то идем в подпрограмму, запоминая адрес для возврата, и увеличиваем значение в ячейке 015, иначе не идем в подпрограмму. В любом случае мы окажемся в ячейке 01A. Там мы увеличиваем значение в 014 \Leftrightarrow уменьшаем счетчик цикла и переходим к другому числу. Если счетчик стал ≥ 0 , то завершаем программу.
- Области данных и результатов:
 - Результат: 015
 - Данные: 010 – 013
- Расположение в памяти ЭВМ исходных данных и результатов:
 - 00A: косвенная адресация на новое число
 - 010 – 013: сами числа
 - 014: счетчик цикла
 - 015: ответ
 - 016 – 019: взятие нового числа, запуск подпрограммы, если такой необходим и переход к следующей итерации цикла.
 - 01A – 01D: увеличение счетчика на 1, и проверка на конечность
 - 045: адрес для возврата
 - 046 – 049: подпрограмма, которая увеличивает значение в 015 на 1
 - 04A: Адресация по значению в ячейке 045 \Leftrightarrow (045) = адрес для возврата.
- Адреса первой и последней исполняемых команд:
 - Первой: 016
 - Последней: 01C

Код программы:

```
CMD CK 016
00A 0010 // link to my numbers

00B 0000
00C 0000
00D 0000
00E 0000
00F 0000

/* numbers */
010 8080
011 ABDA
012 630D
013 71BD

014 FFFC // loop counter

015 0000 // answer

/* new number */
016 F200 // clear
017 480A // 80A --> 00A -> A += (00A), (00A)++
018 A01A // if (A < 0) -> 01A, 019

019 2045 // to subprogram

/* increment or end*/
01A 0014 // ((01A))++, if ( ((01A)) >= 0) -> CK++
01B C016 // to 016
01C F000 // end

01D 0000

/* subprogram */
045 0000 // nothing
046 F200 // clear
047 F800 // A++
048 4015 // A += (015)
049 3015 // (015) = A
04A C845 // to (045) <-> 01A
```