

Федеральное государственное автономное образовательное учреждение
высшего образования «Национальный исследовательский университет ИТМО»

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа по основам профессиональной деятельности №4

Вариант 15601

Выполнил: Снагин Станислав Максимович
Проверил: Блохина Елена Николаевна
Группа: Р3115

Санкт Петербург 2025 г.

Содержание

Задания и цели работы.....	3
Ход работы.....	4
Текст исходной программы.....	4
Назначение программы и реализуемая ею функция.....	5
Область представления.....	5
Область допустимых значений.....	5
Расположение данных в памяти ЭВМ программы, исходных данных и результатов:..	6
Программа.....	6
Исходные данные.....	6
Результат.....	6
Адреса первой и последней выполняемой программы.....	6
Таблица трассировки.....	6
Вывод.....	8

Задания и цели работы

Необходимо по варианту 1560

39E: + 0200		3AC: 0800		3BA: YYYY		729: EC01
39F: EE1C		3AD: 0740		3BB: XXXX		72A: 0A00
3A0: AE19		3AE: 6E0D		3BC: FF3A		72B: 0F6C
3A1: 0700		3AF: EE0C		-----		72C: 00C1
3A2: 0C00		3B0: AE0A		71F: AC01		
3A3: D71F		3B1: 0700		720: F204		
3A4: 0800		3B2: 0C00		721: F003		
3A5: 0700		3B3: D71F		722: 7E08		
3A6: 6E15		3B4: 0800		723: F004		
3A7: EE14		3B5: 0740		724: F803		
3A8: AE10		3B6: 4E05		725: 4C01		
3A9: 0740		3B7: EE04		726: 6E05		
3AA: 0C00		3B8: 0100		727: CE01		
3AB: D71F		3B9: ZZZZ		728: AE02		

Ход работы

Текст исходной программы

Адрес	Код команды	Мнемоника	Комментарии
39E+	0200	CLA	Начало программы
39F	EE1C	ST IP + 28	(AC=0) AC → D, отчистка результата
3A0	AE19	LD IP + 25	Y → AC
3A1	0700	INC	AC = Y + 1
3A2	0C00	PUSH	Передача Y + 1 (аргумента) в стек (0x7FF)
3A3	D71F	CALL 0x71F	Вызов подпрограммы F(Y + 1) (0x7FE)
3A4	0800	POP	Получаем результат подпрограммы в AC
3A5	0700	INC	AC = AC + 1 (AC = F(Y + 1) + 1)
3A6	6E15	SUB IP + 21	AC = AC – D вычитаем из результата ноль AC = F(Y + 1) + 1 – 0
3A7	EE14	ST IP + 20	AC → D Сохраняем новый результат (D = F(Y + 1) + 1)
3A8	AE10	LD IP + 16	Z → AC
3A9	0740	DEC	AC = AC - 1 (AC = Z - 1)
3AA	0C00	PUSH	Передача Z – 1 (аргумента) в стек (0x7FF)
3AB	D71F	CALL 0x71F	Вызов подпрограммы F(Z – 1)

			(0x7FE)
3AC	0800	POP	Результат из SP → AC
3AD	0740	DEC	$AC = F(Z - 1) - 1$
3AE	6E0D	SUB IP + 13	$AC = F(Z - 1) - D$ $(AC = F(Z - 1) - F(Y + 1) - 1)$
3AF	EE0C	ST IP + 12	$AC \rightarrow D$ $(D = F(Z - 1) - F(Y + 1) - 1)$
3B0	AE0A	LD IP + 10	$X \rightarrow AC$
3B1	0700	INC	$AC = AC + 1$ $(AC = X + 1)$
3B2	0C00	PUSH	Передача аргумента $X + 1$ в стек (0x7FF)
3B3	D71F	CALL 0x71F	Вызов подпрограммы $F(X + 1)$ (0x7FE)
3B4	0800	POP	Результат из SP → AC
3B5	0740	DEC	$AC = AC - 1$ $(AC = F(X + 1) - 1)$
3B6	4E05	ADD IP + 5	$AC = AC + D$ $(AC = F(X + 1) - 1 + F(Z - 1) - F(Y + 1) - 1)$
3B7	EE04	ST IP + 4	$AC \rightarrow D$, итоговый результат программы $(D = F(X + 1) - 1 + F(Z - 1) - F(Y + 1) - 1)$
3B8	0100	HLT	Остановка программы

Текст подпрограммы

Адрес	Код команды	Мнемоника	Комментарии
71F	AC01	LD SP + 1	Загрузка аргумента (0x7FF) + флаги
720	F204	BMI IP + 4	Если АС < 0 , переход на 0x725
721	F003	BEQ IP + 3	Если АС = 0 , переход на 0x725
722	7E08	CMP IP + 8	Флаги по результату АС - А
723	F004	BEQ IP + 4	Если АС - А = 0 , переход на 0x728
724	F803	BLT IP + 3	Если АС - А < 0 , переход на 0x728
725	4C01	ADD SP + 1	Умножить аргумент на 2
726	6E05	SUB IP + 5	АС = АС - В + флаги по результату. т. е. Если изначальный аргумент был отрицательный, то АС = 2X - В
727	CE01	JUMP IP + 1	Переход на 0x729
728	AE02	LD IP + 2	А → АС -- Загрузка константы т. е. Если значение АС ≤ А , загружаем А в АС
729	EC01	ST (SP + 1)	АС → М Сохранение результата в ячейку аргумента программы в стеке
72A	0A00	RET	Возврат

Данные подпрограммы

72B	0F6C	А	Константа
-----	------	----------	------------------

			F6C (16) = 3948 (10)
72C	00C1	B	Константа, вычитаемое в 2X – B C1 (16) = 193 (10)

Данные

3B9	ZZZZ	Z	Значение Z
3BA	YYYY	Y	Значение Y
3BB	XXXX	X	Значение X
3BC	00C1	D	Результат

Назначение программы и реализуемая ею функция

Программа представляет собой сумму трех значений

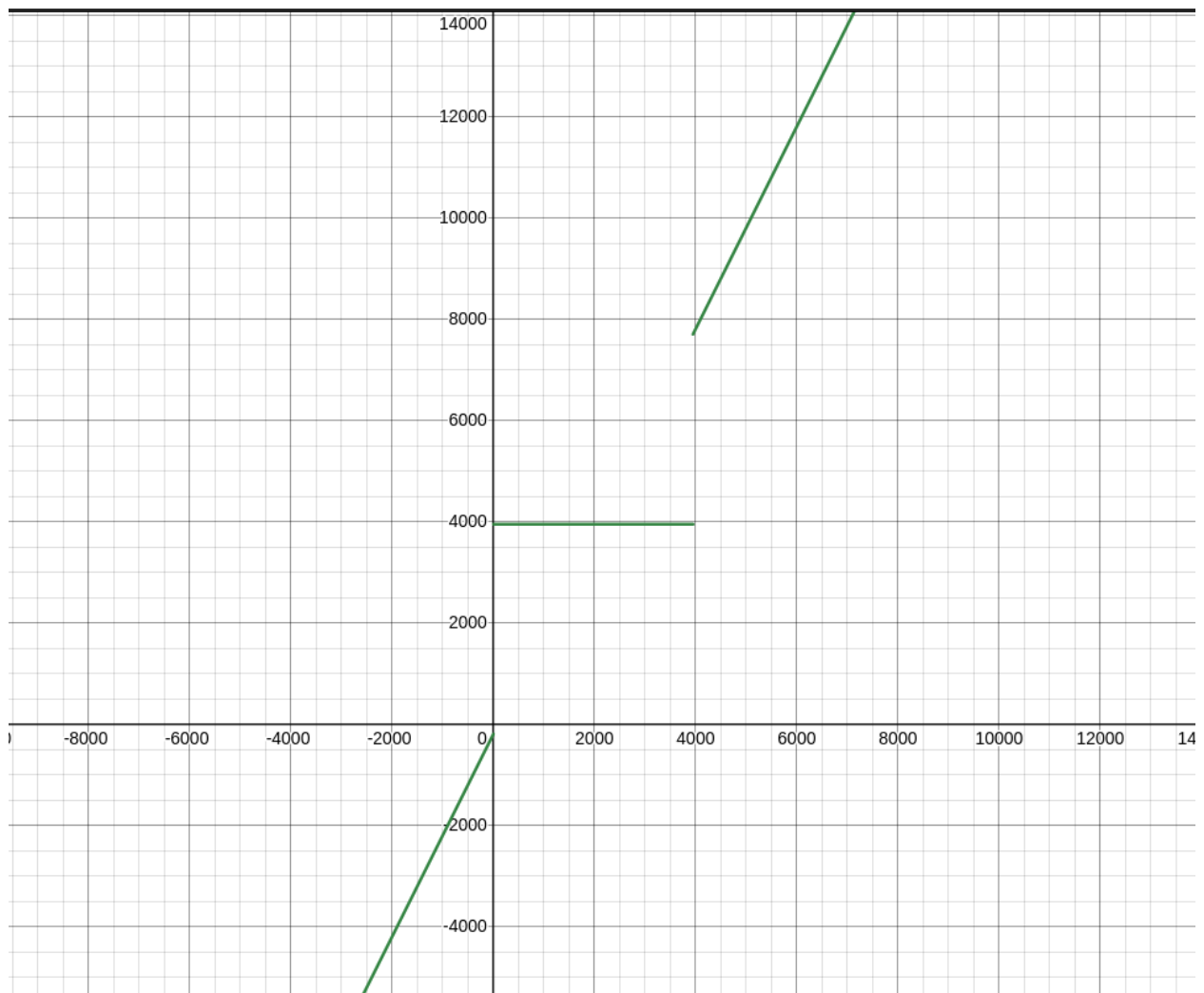
$$D = F(X + 1) - 1 + F(Z - 1) - F(Y + 1) - 1$$

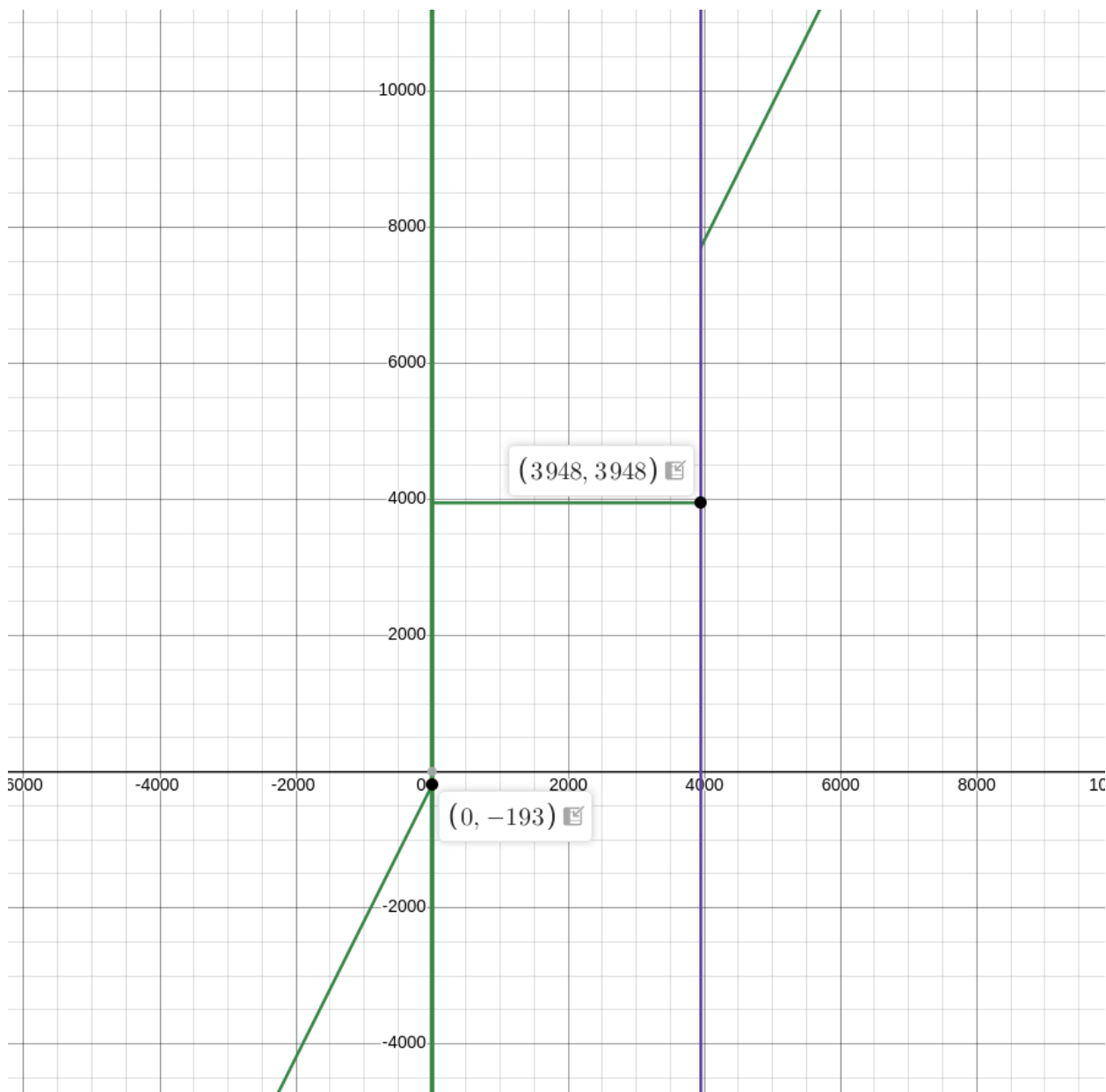
$$D = F(X + 1) - F(Y + 1) - F(Z - 1) - 2$$

от кусочно заданной функции

$$F(X) = \{ 2X - 193, \text{ если } (x \leq 0 \text{ или } x > 3948) ; 3948, \text{ если } 0 < x \leq 3948 \}$$

от переменных X, Y и Z. На вход в качестве аргументов через стек подается X + 1, Y + 1 и Z - 1 соответственно, а результат подпрограммы записывается в ту же ячейку данных, через которую в подпрограмму и передавались аргументы (в нашем случае, это 0x7FF)





Область представления

Некоторые значения X , Y , Z — 16-ти разрядные знаковые числа в допкоде.

Константы A , B -- 16-ти разрядные знаковые числа в допкоде.

Результат D -- 16-ти разрядное знаковое число в допкоде.

Область допустимых значений

$$A = 0F6C_{(16)} = 3948_{(10)}$$

$$B = C1(16) = 193(10)$$

(1) Рассмотрим случай, где $0 < a \leq 3948$

Функция $F(a)$ вернёт 3948, если значение аргумента a будет находиться в диапазоне $(0, 3948]$. при использовании любых значений a , переполнения не возникнет

(2) Рассмотрим случай, где $a \leq 0$.

Максимальное значение достигается в $F(0) = -193$, минимальное число в $F(-2^{15})$. Переполнения не возникнет, но это будет минимальное возможное значение функции

$$F(-2^{15}) = -65729$$

(3) Рассмотрим случай, где $a > 3948$: так как функция монотонно возрастает на этом промежутке, то и максимальное значение достигается в $2^{15} - 1$:

$$F(2^{15} - 1) = 65341$$

$$D = F(X + 1) - F(Y + 1) - F(Z - 1) - 2$$

Минимально для D мы можем получить:

$$D = -65729 - (65341) - (65341) - 2 = -196413 < -2^{15}, \text{ переполнение}$$

Максимальное для D :

$$D = 65341 - (-65729) - (-65729) - 2 = 196797 > 2^{15} - 1, \text{ переполнение}$$

Исходя из этого, минимальное и максимальное значение -2^{15} и $2^{15} - 1$

$$X \text{ и } Y = [-2^{15}; 2^{15} - 2]$$

$$Z = [-2^{15} + 1, 2^{15}]$$

$$D = [-2^{15}, 2^{15} - 1]$$

Расположение данных в памяти ЭВМ программы, исходных данных и результатов:

Программа

0x39E — 0x3B8 - команды

0x3B9 — 0x3BB - данные

0x3BC — данные (результат)

Подпрограмма

0x71F — 0x72A - команды

0x72B и 0x72C — данные (константы)

Адреса первой и последней выполняемой программы

Адрес первой выполняемой программы: 0x39E

Адрес последней выполняемой программы: 0x72A

Вывод

В ходе лабораторной работы были подробно изучены команды, связанные с работой со стеком: POP, RET, CALL, а так же прямая адресация относительно SP.