МИНОБРНАУКИ РОССИИ САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА) Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №3

по дисциплине «Организация ЭВМ и систем»

Тема: Представление и обработка целых чисел. Организация ветвящихся процессов.

Студент гр. 1383	Сапожников А.Э.
Преподаватель	Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Получение знаний о ветвлениях и работе с целочисленными значениями на языке Ассемблера.

Задание.

Разработать на языке Ассемблера программу, которая по заданным целочисленным значениям параметров a, b, i, k вычисляет: a) значения функций

i1 = f1(a,b,i) и i2 = f2(a,b,i); b) значения результирующей функции res = f3(i1,i2,k), где вид функций f1 и f2 определяется из табл. 2, а функции f3 - из

табл.3 по цифрам шифра индивидуального задания (n1,n2,n3), приведенным в

табл.4. Значения a, b, i, k являются исходными данными, которые должны выбираться студентом самостоятельно и задаваться в процессе исполнения программы в режиме отладки. При этом следует рассмотреть всевозможные

комбинации параметров a, b и k, позволяющие проверить различные маршруты

выполнения программы, а также различные знаки параметров а и b.

Выполнение работы.

Вариант 3.7.5

$$i1=f1(a,b,i) := a>b ? : 7 - 4*i : 8 - 6*i$$

$$i2=f2(a,b,i) := a>b ? -(4*i-5) : 10 - 3*i$$

$$res=f3(i1,i2,k) := min(|i1|, 6) ? k=0 : |i1|+|i2|$$

1. В целях оптимизации сразу обрабатываем обе функции:

(f1 и f3 для ситуаций a > b или a <= b)

2. формулы:

7 -
$$4*i = 7 + (-i << 2)$$

$$8 - 6*i = 8 - (i << 2 + i + i)$$

$$-(4*i-5) = 5 - (i << 2)$$

10 - 3*i = 10 + (-i << 2) + i

для упрощения рассчетов считается 4*і

Таблица 1- тесты программы

Номер	Входные данные	Результат
1	a = 0 b = 0 i = 0 k = 0	i1 = 8-0 i2 = 10 res = 6
2	a = 0 b = -1 i = 1 k = 0	i1 = 3 i2 = 1 res = 1
3	a = 0 b = -1 i = -1 k = 0	i1 = 11 i2 = 9 res = 6
4	a = 1 $b = 1$ $i = 1$ $k = 1$	i1 = 2 i2 = 7 res = 9
5	a = 2 b = 3 i = 4 k = -1	i1 = -16 i2 = -2 res = 18
6	a = 3 b = 2 i = -2 k = 1	i1 = 15 i2 = 13 res = 28
7	a = -3 b = 2 i = 2 k = 0	i1 = -4 i2 = 4 res = 4

Выводы.

Мною были изучены работа с целыми числами на языке Ассемблера и ветвления.

Приложение А

Исходный код программы

Название файла: lr3.asm

	oft (R) 22 20:26:1	Macro Assembler Version 5.10
Page	1-1	
	2 2 1	a EQU -3 b EQU 2 i EQU 2 k EQU 1 AStack SEGMENT STACK DW 12 DUP(?)
0018		AStack ENDS
0000		DATA SEGMENT
0000 0002	0000	res DW 0 DATA ENDS
0000		CODE SEGMENT ASSUME CS:CODE, DS:DATA,
SS:AStack		ASSOME CS.CODE, DS.DATA,
0000 0000 0001 0003	1E 2B C0 50	Main PROC FAR push ds sub ax, ax push ax
		mov ax, i shl ax, 1 shl ax, 1 mov bx, ax

```
000D B9 FFFD
                           mov cx, a
 0010 83 F9 02
                            cmp cx, b
 0013 7E 0D
                        jle fless
 0015
     F7 D8
                        neg ax
 0017 05 0007
                            add ax, 7
 001A 50
                  push ax ;i1
     05 FFFE
 001B
                            add ax, -2
      50
 001E
                   push ax ;i2
 001F EB 16 90
                            jmp f3
 0022
                fless:
 0022 05 0002
                            add ax, i
 0025 05 0002
                            add ax, i
 0028 F7 D8
                        neg ax
 002A 05 0008
                            add ax, 8
 002D 50
                   push ax
 002E 05 FFF8
                            add ax, -8
 0031 D1 F8
                        sar ax, 1
 0033 05 000A
                            add ax, 10
 0036 50
                   push ax
            f3:
0037
0037 5A
                  pop dx ;i2
Microsoft (R) Macro Assembler Version 5.10
11/13/22 20:26:1
Page 1-2
0038 59
                   pop cx ;i1
 0039 83 F9 00
                            cmp cx, 0
     79 02
                         jns i1abs ; |i2| = |i2|
 003C
 003E
     F7 D9
                        neg cx
 0040
                ilabs:
 0040 B8 0001
                            mov ax, k
 0043 3D 0000
                            cmp ax, 0
 0046 75 0B
                        jne f3v2
 0048 83 F9 06
                            cmp cx, 6
 004B 7C 0F
                        jl finish
 004D B9 0006
                            mov cx, 6
 0050 EB 0A 90
                            jmp finish
```

i1	0053 0053 0056			00	f3	v2:	•		jn		cmp i2a			i1	=
+ +	0058	F7	DA		<u>.</u> 0	م ام			nec	j (xb				
	005A 005A	03	CA		12	abs	:		ado	d (CX,	dx			
	005C 005C 005D 005E 005F 005F	59 CB			Ma CC EN	in DE D N	pus pop ret EN EN	sh ; iDE iDS	CX C						
	Micros 11/13/				Mac	ro	A	SS	sem	bl	er	Ver	rsio	n 5.	10
	Symbol	s-1													
;	Segmen	ts a	and	Grou	ıps:										
Combi	ine Cl	ass		N a	m e						Lenç	ŋth		Ali	gn
STACE	ASTACK	•				•		•	•	•	•	001	. 8	P <i>F</i>	ARA
	CODE .	•				•	•	•		•	•	005	F	P <i>P</i>	ARA
	DATA .	•				•	•	•	•	•	•	000	2	P <i>P</i>	ARA
	Symbol	s:													
Attr				N a	m e						Тур	Э		Val	ue
-0003	A	•				•	•	•	•	•	•			NUME	BER
0002	В	•				•	•	•	•	•				NUME	BER

		CODE	•	•	•	•	•	•	•	•	•	•	•	•	L	NEAR
	F3V	2	•	•	•	•	•	•	•	•	•	•	•	•	L	NEAR
	FIN	CODE ISH .						•						•	L	NEAR
		CODE SS .						•	•					•	L	NEAR
0022	2	CODE														
0002			•	•	•	•	•	•	•	•		•				NUMBER
	I1A	BS .		•	•	•		•	•	•	•	•		•	L	NEAR
		CODE BS .						_							L	NEAR
		CODE	•	•	•	•	•	•	•	•	•	•	•	•		TVIII (
0001			•	•	•	•	•	•	•	•	•	•	•	•		NUMBER
		Ν						•	•					•	F	PROC
0000)	CODE		I	ıer.	ıgt	h	=	00)5E	7					
		 DATA	•	•			•	•	•	•			•	•	L	WORD
	@CP	U						•	•					•	TEXT	0101h
		LENAME RSION			•	•	•	•	•	•	•	•	•	•	TEXT TEXT	
	G V E.	TOTON	•	•	•	•	•	•	•	•	•	•	•	•	тпут	$\supset \perp \cup$

78 Source Lines

78 Total Lines

20 Symbols

47978 + 461329 Bytes symbol space free

0 Warning Errors

O Severe Errors

ПРИЛОЖЕНИЕ В ФАЙЛЫ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: LR3.lst

Microsoft (R) 11/13/22 20:2	Macro Assembler Version 5.10 6:1
Page 1-1	
=-0003 = 0002 = 0002 = 0001 0000 0000 000C[????	a EQU -3 b EQU 2 i EQU 2 k EQU 1 AStack SEGMENT STACK DW 12 DUP(?)
0018	AStack ENDS
0000	DATA SEGMENT
0000 0000 0002	res DW 0 DATA ENDS
0000	CODE SEGMENT ASSUME CS:CODE, DS:DATA, SS:AStack
0000 0000 1E 0001 2B C0 0003 50	Main PROC FAR push ds sub ax, ax push ax
0004 B8 0002 0007 D1 E0 0009 D1 E0 000B 8B D8	mov ax, i shl ax, 1 shl ax, 1 mov bx, ax
000D B9 FFFD	mov cx, a

```
cmp cx, b
jle fless
0010 83 F9 02
 0013 7E 0D
0015 F7 D8
                   neg ax
0017 05 0007
                      add ax, 7
001A 50
                 push ax ;i1
001B 05 FFFE
                          add ax, -2
                push ax ;i2
001E 50
001F EB 16 90
                          jmp f3
0022
             fless:
0022 05 0002
                          add ax, i
0025 05 0002
                          add ax, i
0028 F7 D8
                      neg ax
002A 05 0008
                          add ax, 8
002D 50
                 push ax
002E 05 FFF8
                          add ax, -8
0031 D1 F8
                   sar ax, 1
0033 05 000A
0036 50 push ax
                          add ax, 10
0037 5A
               pop dx ;i2
Microsoft (R) Macro Assembler Version 5.10
   11/13/22 20:26:1
   Page 1-2
                 pop cx ;i1
0038 59
0039 83 F9 00
                         cmp cx, 0
 003C 79 02
                      jns ilabs ; |i2| = |i2|
003E F7 D9
                      neg cx
0040
              ilabs:
 0040 B8 0001
                          mov ax, k
0043 3D 0000
                          cmp ax, 0
                       jne f3v2
0046 75 OB
0048 83 F9 06
                         cmp cx, 6
                      jl finish
004B 7C 0F
004D B9 0006
                         mov cx, 6
0050 EB 0A 90
                          jmp finish
               f3v2:
0053
```

0053 8 0056 7 0058 F 005A 005A 0	'7 DA	i2abs:	neg	cmp dx, i2abs; dx cx, dx		: i1
005C 005C 5 005D 5 005E C 005F 005F	9	finish: push pop ret Main END CODE END END Main	CX P S			
	ft (R) 13/22 20:20	Macro		embler	Versio	on 5.10
Symb	ools-1					
Segments	and Group	os:				
Combine	Nam	е		Length		Align
ASTACK .				. 00	18	PARA
STACK CODE				. 00)5F	PARA
NONE DATA				. 00	002	PARA
NONE						
Symbols:						
Attr	N a m	. е		Туре		Value
А				. NUM	IBER	-0003
в				. NUM	IBER	0002
F3				. L	NEAR	0037
CODE F3V2 CODE				. L	NEAR	0053

FINISH CODE	•	•	•	•	•	•			•			•	•	L :	NEAR		005C
FLESS CODE	•	•	•	•	•	•	•	•	•	•	•	•		L :	NEAR		0022
I														NUME	3ER	0002	
I1ABS	•	•	•	•	•	•	•	•	•	•	•		•		NEAR	0002	0040
CODE																	
I2ABS		•											•	L	NEAR		005A
CODE																	
T.7														N T T T N # T		0001	
K	•	•	•	•	•	•	•	•	•	•	•	•	•	NUME	3ER	0001	
MAIN .																	
	_	_	_	_	_	_	_	_	_	_	_	_	_	F	PROC		0.000
	• I	Ler	ngt	th	· =	00)5E	•	•	•	•	•	•	F	PROC		0000
CODE	·	Ler	ngt	th	• =	00	•)5E	•	•	•	•	•	•	F	PROC		0000
		er	ngt	th	• =)5E	•				•	•		PROC WORD		0000
CODE	I	· Ler	ngt	· ch	• =		• 95E •	•					•				
CODE RES . DATA	· I	· Ler	· ngt	ch	• =)5E	•						L	WORD	1 1-	
CODE RES DATA GCPU .			· ngt ·	· h ·	• = •		• 95E •	•						L TEXT	WORD	1h	
CODE RES . DATA	AME		ngt	• th	· = ·		• 95E •	•						L	WORD 010: LR3	1h	

- 78 Source Lines
- 78 Total Lines
- 20 Symbols

47978 + 461329 Bytes symbol space free

- 0 Warning Errors
- O Severe Errors