

№7 -

Nasm

Invalid Date

-
-
-

1:

```
(base) hhl@LAPTOP-058L8DAA:~$ cd ~/work/study/2025-2026/архитектура\ компьютера/arch-pc/labs/lab07
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ touch lab7-1.asm
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$
```

Figure 1:

: ~/work/arch-pc/lab07 ,
.

2:

```
(base) hhl@LAPTOP-058L8DAA:~$ cd ~/work/study/2025-2026/архитектура\ компьютера/arch-pc/labs/lab07
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ touch lab7-1.asm
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$
```

Figure 2:

```

:          touch          lab7-1.asm,
.

```

3:

```

GNU nano 7.2 lab7-1.asm *
#include 'in_out.asm'

SECTION .data
msg1: DB 'Сообщение "% 1",0
msg2: DB 'Сообщение "% 2",0
msg3: DB 'Сообщение "% 3",0

SECTION .text
GLOBAL _start
_start:
    jmp _label2

_label1:
    mov eax, msg1
    call sprintf

_label2:
    mov eax, msg2
    call sprintf

_label3:
    mov eax, msg3
    call sprintf

_end:
    call quit

```

Figure 3:

```

:          jmp          lab7-1.asm,
.

```

4:

```

(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ nano lab7-1.asm
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ nasm -f elf lab7-1.asm
lab7-1.asm:1: error: unable to open include file 'in_out.asm': No such file or directory
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ nasm -f elf lab7-1.asm
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ ld -m elf_i386 lab7-1.o -o lab7-1
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ ./lab7-1
Сообщение "% 2
Сообщение "% 3

```

Figure 4:

```

:          ,          _label2,          _label1,
jmp.

```

5:

```
GNU nano 7.2 lab7-1.asm *
%include 'in_out.asm'

SECTION .data
msg1: DB 'Сообщение %s 1',0
msg2: DB 'Сообщение %s 2',0
msg3: DB 'Сообщение %s 3',0

SECTION .text
GLOBAL _start
_start:
    jmp _label3

_label1:
    mov eax, msg1
    call sprintf
    jmp _end

_label2:
    mov eax, msg2
    call sprintf
    jmp _label1

_label3:
    mov eax, msg3
    call sprintf
    jmp _label2

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  ^U Undo      ^A Set Mark
^X Exit      ^R Read File ^_ Replace   ^U Paste     ^J Justify   ^/ Go To Line ^E Redo      ^G Copy
```



```
GNU nano 7.2 lab7-1.asm *
msg2: DB 'Сообщение %s 2',0
msg3: DB 'Сообщение %s 3',0

SECTION .text
GLOBAL _start
_start:
    jmp _label3

_label1:
    mov eax, msg1
    call sprintf
    jmp _end

_label2:
    mov eax, msg2
    call sprintf
    jmp _label1

_label3:
    mov eax, msg3
    call sprintf
    jmp _label2

_end:
    call quit

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  ^U Undo      ^A Set Mark
^X Exit      ^R Read File ^_ Replace   ^U Paste     ^J Justify   ^/ Go To Line ^E Redo      ^G Copy
```

: , jmp ,

6:

```
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ nano lab7-1.asm
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ nasm -f elf lab7-1.asm
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ ld -m elf_i386 lab7-1.o -o
lab7-1
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ ./lab7-1
Сообщение % 3
Сообщение % 2
Сообщение % 1
```

Figure 5:

2 → : , (3 →
1), .

7:

lab7-2.asm

```
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ touch lab7-2.asm
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ nano lab7-2.asm
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ nasm -f elf lab7-2.asm
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ ld -m elf_i386 lab7-2.o -o
lab7-2
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ ./lab7-2
Введите B: 2
Наибольшее число: 2
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ ./lab7-2
Введите B: 1
Наибольшее число: 2
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ ./lab7-2
Введите B: 3
Наибольшее число: 3
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ ./lab7-2
Введите B: 4
Наибольшее число: 4
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ ./lab7-2
Введите B: 213
Наибольшее число: 213
```

Figure 6: lab7-2

: lab7-2.asm, .

8:

The image shows two screenshots of a GNU nano 7.2 editor window. The top screenshot shows the beginning of the assembly code, including the .data section with messages and initial values for A, B, and C, and the .text section starting with the _start label. The bottom screenshot shows the continuation of the code, including the check_B label, the fin label, and the final calls to sprint and iprintlnf. The code is as follows:

```
GNU nano 7.2 lab7-2.asm *
#include 'in_out.asm'

SECTION .data
msg1: DB 'Введите B: ',0
msg2: DB "Наибольшее число: ",0
A: DD 20
C: DD 50

SECTION .bss
max: RESB 10
B: RESB 10

SECTION .text
GLOBAL _start
_start:
    mov eax, msg1
    call sprint

    mov ecx, B
    mov edx, 10
    call sread

    mov eax, B
    call atoi
    mov [B], eax

    mov [B], eax

    mov ecx, [A]
    mov [max], ecx

    cmp ecx, [C]
    jg check_B
    mov ecx, [C]
    mov [max], ecx

check_B:
    mov eax, max
    call atoi
    mov [max], eax

    mov ecx, [max]
    cmp ecx, [B]
    jg fin
    mov ecx, [B]
    mov [max], ecx

fin:
    mov eax, msg2
    call sprint
    mov eax, [max]
    call iprintlnf
    call quit
```

: , A, B, C ,
cmp .

9: lab7-2.asm

```
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ touch lab7-2.asm
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ nano lab7-2.asm
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ nasm -f elf lab7-2.asm
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ ld -m elf_i386 lab7-2.o -o
lab7-2
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ ./lab7-2
Введите B: 2
Наибольшее число: 2
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ ./lab7-2
Введите B: 1
Наибольшее число: 2
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ ./lab7-2
Введите B: 3
Наибольшее число: 3
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ ./lab7-2
Введите B: 4
Наибольшее число: 4
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ ./lab7-2
Введите B: 213
Наибольшее число: 213
```

Figure 7: lab7-2

: , B, A C,

10:

```

hh1@LAPTOP-058L8DAA: ~/w  ×  +  ▾
(base) hh1@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
(base) hh1@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ cat lab7-2.lst
1                                     %include 'in_out.asm'
2                                     <1> ;----- slen -----
3                                     <1> ; Функция вычисления длины сообщения
4                                     <1> slen:
5 00000000 53                         <1> push    ebx
6 00000001 89C3                       <1> mov     ebx, eax
7                                     <1>
8 00000003 803800                     <1> nextchar:
9 00000006 7403                       <1> cmp     byte [eax], 0
10 00000008 40                       <1> jz      finished
11 00000009 EBF8                      <1> inc     eax
12                                     <1> jmp     nextchar
13                                     <1> finished:
14 0000000B 29D8                      <1> sub     eax, ebx
15 0000000D 5B                       <1> pop     ebx
16 0000000E C3                       <1> ret
17                                     <1>
18                                     <1>
19                                     <1> ;----- sprint -----
20                                     <1> ; Функция печати сообщения
21                                     <1> ; входные данные: mov eax,<message>
22                                     <1> sprint:
23 0000000F 52                         <1> push    edx
24 00000010 51                         <1> push    ecx
25 00000011 53                         <1> push    ebx
26 00000012 50                         <1> push    eax
27 00000013 E8E8FFFFFF                <1> call    slen
28                                     <1>
29 00000018 89C2                      <1> mov     edx, eax
30 0000001A 58                         <1> pop     eax
31                                     <1>
32 0000001B 89C1                      <1> mov     ecx, eax
33 0000001D B801000000                <1> mov     ebx, 1
34 00000022 B804000000                <1> mov     eax, 4
35 00000027 CD80                      <1> int     80h
36                                     <1>
37 00000029 5B                         <1> pop     ebx
38 0000002A 59                         <1> pop     ecx
39 0000002B 5A                         <1> pop     edx
40 0000002C C3                       <1> ret
41                                     <1>
42                                     <1>
43                                     <1> ;----- sprintf -----
44                                     <1> ; Функция печати сообщения с переводом строки
45                                     <1> ; входные данные: mov eax,<message>
46                                     <1> sprintf:
47 0000002D E8DDFFFFFF                <1> call    sprint
48                                     <1>
49 00000032 50                         <1> push    eax
50 00000033 B80A000000                <1> mov     eax, 0Ah
51 00000038 50                         <1> push    eax

```

Figure 8:

: nasm -f elf -l lab7-2.lst lab7-2.asm ,

11:

```
hhl@LAPTOP-058L8DAA: ~/w... X + v
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/Lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/Lab07$ cat lab7-2.lst
1      %include 'in_out.asm'
1      <1> ;----- slen -----
2      <1> ; Функция вычисления длины сообщения
3      <1> slen:
4      00000000 53      <1>      push    ebx
5      00000001 89C3    <1>      mov     ebx, eax
6      <1>
7      <1> nextchar:
8      00000003 803800    <1>      cmp     byte [eax], 0
9      00000006 7403    <1>      jz      finished
10     00000008 40      <1>      inc     eax
11     00000009 EBF8    <1>      jmp     nextchar
12     <1>
13     <1> finished:
14     0000000B 29D8    <1>      sub     eax, ebx
15     0000000D 5B      <1>      pop     ebx
16     0000000E C3      <1>      ret
17     <1>
18     <1>
19     <1> ;----- sprint -----
20     <1> ; Функция печати сообщения
21     <1> ; входные данные: mov eax,<message>
22     <1> sprint:
23     0000000F 52      <1>      push    edx
24     00000010 51      <1>      push    ecx
25     00000011 53      <1>      push    ebx
26     00000012 50      <1>      push    eax
27     00000013 E8E8FFFFFF <1>      call    slen
28     <1>
29     00000018 89C2    <1>      mov     edx, eax
30     0000001A 58      <1>      pop     eax
31     <1>
32     0000001B 89C1    <1>      mov     ecx, eax
33     0000001D B801000000 <1>      mov     ebx, 1
34     00000022 B804000000 <1>      mov     eax, 4
35     00000027 CD80    <1>      int     80h
36     <1>
37     00000029 5B      <1>      pop     ebx
38     0000002A 59      <1>      pop     ecx
39     0000002B 5A      <1>      pop     edx
40     0000002C C3      <1>      ret
41     <1>
42     <1>
43     <1> ;----- sprintLF -----
44     <1> ; Функция печати сообщения с переводом строки
45     <1> ; входные данные: mov eax,<message>
46     <1> sprintLF:
47     0000002D E8DDFFFFFF <1>      call    sprint
48     <1>
49     00000032 50      <1>      push    eax
50     00000033 B80A000000 <1>      mov     eax, 0Ah
51     00000038 50      <1>      push    eax
52     <1>
53     <1>
54     00000040 58      <1>      pop     eax
55     00000041 58      <1>      pop     eax
56     00000042 C3      <1>      ret
57     <1>
58     <1> ;----- sread -----
59     <1> ; Функция считывания сообщения
60     <1> ; входные данные: mov eax,<buffer>, mov ebx,<N>
61     <1> sread:
62     00000043 53      <1>      push    ebx
63     00000044 50      <1>      push    eax
64     <1>
65     00000045 BB00000000 <1>      mov     ebx, 0
66     0000004A B803000000 <1>      mov     eax, 3
67     0000004F CD80    <1>      int     80h
68     <1>
69     00000051 5B      <1>      pop     ebx
70     00000052 59      <1>      pop     ecx
71     00000053 C3      <1>      ret
72     <1>
73     <1> ;----- iprint -----
74     <1> ; Функция вывода на экран чисел в формате ASCII
75     <1> ; входные данные: mov eax,<int>
76     <1> iprint:
77     00000054 50      <1>      push    eax
78     00000055 51      <1>      push    ecx
79     00000056 52      <1>      push    edx
80     00000057 56      <1>      push    esi
81     00000058 B900000000 <1>      mov     ecx, 0
82     <1>
83     <1> divideLoop:
```

```

hhl@LAPTOP-058L8DAA: ~/w × + v
108                                     <1> ;----- iprintLF -----
109                                     <1> ; Функция вывода на экран чисел в формате ASCII
110                                     <1> ; входные данные: mov eax,<int>
111                                     <1> iprintLF:
112 00000086 E8C9FFFFFF                 <1>     call    iprint
113                                     <1>
114 0000008B 50                         <1>     push    eax
115 0000008C B80A000000                 <1>     mov     eax, 0Ah
116 00000091 50                         <1>     push    eax
117 00000092 89E0                     <1>     mov     eax, esp
118 00000094 E876FFFFFF                 <1>     call    sprint
119 00000099 58                         <1>     pop     eax
120 0000009A 58                         <1>     pop     eax
121 0000009B C3                         <1>     ret
122                                     <1>
123                                     <1> ;----- atoi -----
124                                     <1> ; Функция преобразования ascii-код символа в целое число
125                                     <1> ; входные данные: mov eax,<int>
126                                     <1> atoi:
127 0000009C 53                         <1>     push    ebx
128 0000009D 51                         <1>     push    ecx
129 0000009E 52                         <1>     push    edx
130 0000009F 56                         <1>     push    esi
131 000000A0 89C6                     <1>     mov     esi, eax
132 000000A2 B800000000                 <1>     mov     eax, 0
133 000000A7 B900000000                 <1>     mov     ecx, 0
134                                     <1>
135                                     <1> .multiplyLoop:
136 000000AC 31DB                     <1>     xor     ebx, ebx
137 000000AE 8A1C0E                 <1>     mov     bl, [esi+ecx]
138 000000B1 80FB30                 <1>     cmp     bl, 48
139 000000B4 7C14                     <1>     jl      .finished
140 000000B6 80FB39                 <1>     cmp     bl, 57
141 000000B9 7F0F                     <1>     jg      .finished
142                                     <1>
143 000000BB 80EB30                 <1>     sub     bl, 48
144 000000BE 01D8                     <1>     add     eax, ebx
145 000000C0 BB0A000000                 <1>     mov     ebx, 10
146 000000C5 F7E3                     <1>     mul     ebx
147 000000C7 41                         <1>     inc     ecx
148 000000C8 EBE2                     <1>     jmp     .multiplyLoop
149                                     <1>
150                                     <1> .finished:
151 000000CA 83F900                 <1>     cmp     ecx, 0
152 000000CD 7407                     <1>     je      .restore
153 000000CF BB0A000000                 <1>     mov     ebx, 10
154 000000D4 F7F3                     <1>     div     ebx
155                                     <1>
156                                     <1> .restore:
157 000000D6 5E                         <1>     pop     esi
158 000000D7 5A                         <1>     pop     edx
159 000000D8 59                         <1>     pop     ecx
160 000000D9 5B                         <1>     pop     ebx
161 000000DA C3                         <1>     ret

```

```

hh@LAPTOP-058L8DAA: ~/w  ×  +  ∨
168 000000E0 B801000000 <1> mov     eax, 1
169 000000E5 CD80 <1> int     80h
170 000000E7 C3 <1> ret
2
3
SECTION .data
4 00000000 D092D0B2D0B5D0B4D0- msg1: DB 'Введите B: ',0
4 00000009 B8D182D0B520423A20-
4 00000012 00
5 00000013 D09DD0B0D0B8D0B1D0- msg2: DB "Наибольшее число: ",0
5 0000001C BED0BBD18CD188D0B5-
5 00000025 D0B520D187D0B8D181-
5 0000002E D0BBD0BE3A2000
6 00000035 14000000 A: DD 20
7 00000039 32000000 C: DD 50
8
9
SECTION .bss
10 00000000 <res Ah> max: RESB 10
11 0000000A <res Ah> B: RESB 10
12
13
SECTION .text
14 GLOBAL _start
15 _start:
16 000000E8 B8[00000000] mov     eax, msg1
17 000000ED E81DFFFFFF call    sprint
18
19 000000F2 B9[0A000000] mov     ecx, B
20 000000F7 BA0A000000 mov     edx, 10
21 000000FC E842FFFFFF call    sread
22
23 00000101 B8[0A000000] mov     eax, B
24 00000106 E891FFFFFF call    atoi
25 0000010B A3[0A000000] mov     [B], eax
26
27 00000110 8B0D[35000000] mov     ecx, [A]
28 00000116 890D[00000000] mov     [max], ecx
29
30 0000011C 3B0D[39000000] cmp     ecx, [C]
31 00000122 7F0C jg      check_B
32 00000124 8B0D[39000000] mov     ecx, [C]
33 0000012A 890D[00000000] mov     [max], ecx
34
check_B:
35 00000130 B8[00000000] mov     eax, max
36 00000135 E862FFFFFF call    atoi
37 0000013A A3[00000000] mov     [max], eax
38
39 0000013F 8B0D[00000000] mov     ecx, [max]
40 00000145 3B0D[0A000000] cmp     ecx, [B]
41 0000014B 7F0C jg      fin
42 0000014D 8B0D[0A000000] mov     ecx, [B]
43 00000153 890D[00000000] mov     [max], ecx
44
fin:
45 00000159 B8[13000000] mov     eax, msg2
46 0000015E E8ACFFFFFF call    sprint
47 00000163 A1[00000000] mov     eax, [max]

```

: , , ,
, .

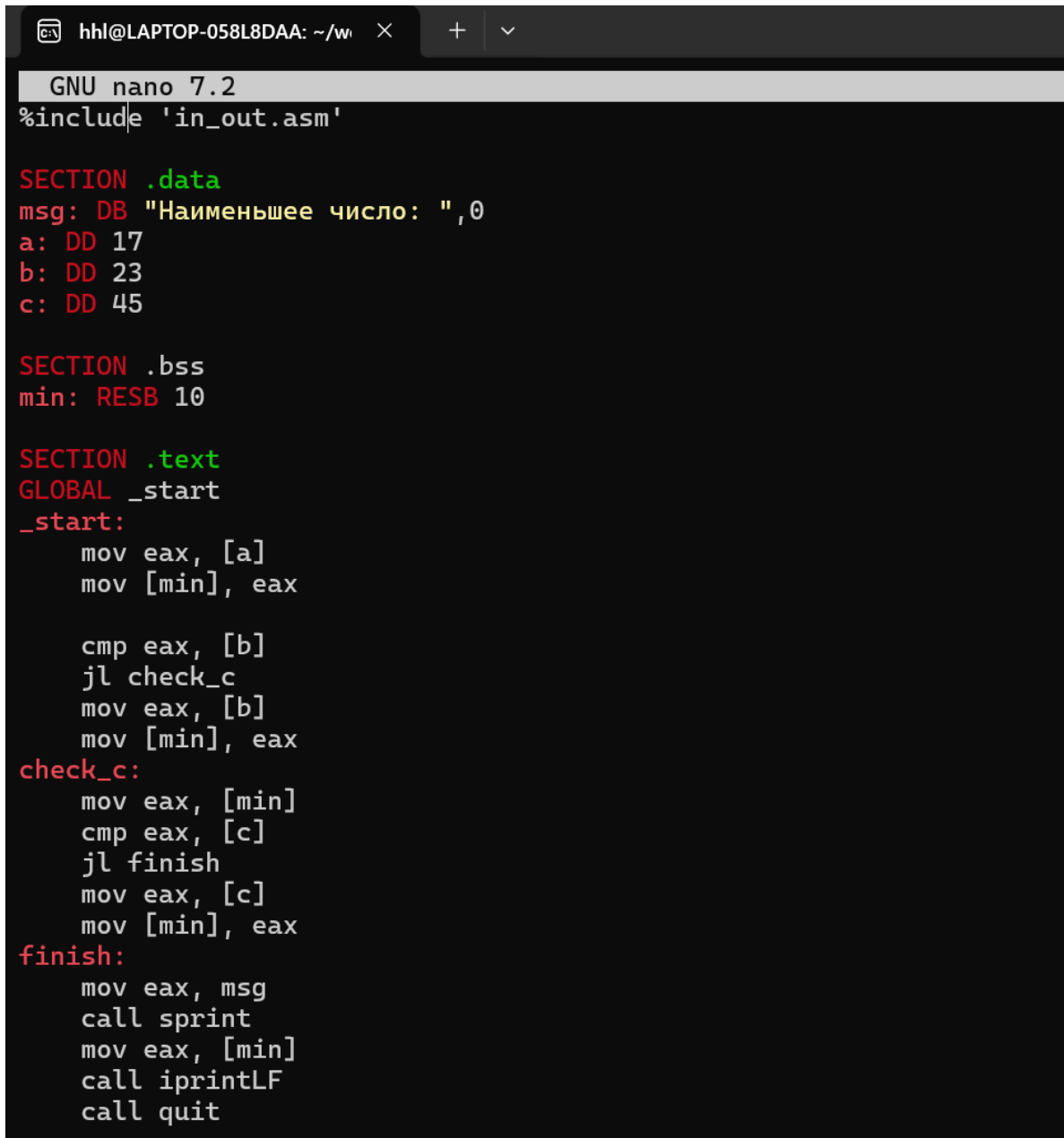
12:

```
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ nano lab7-2.asm  
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm  
lab7-2.asm:19: error: invalid combination of opcode and operands
```

Figure 9:

: , ,
.

1:



```
GNU nano 7.2
%include 'in_out.asm'

SECTION .data
msg: DB "Наименьшее число: ",0
a: DD 17
b: DD 23
c: DD 45

SECTION .bss
min: RESB 10

SECTION .text
GLOBAL _start
_start:
    mov eax, [a]
    mov [min], eax

    cmp eax, [b]
    jnl check_c
    mov eax, [b]
    mov [min], eax
check_c:
    mov eax, [min]
    cmp eax, [c]
    jnl finish
    mov eax, [c]
    mov [min], eax
finish:
    mov eax, msg
    call sprint
    mov eax, [min]
    call iprintLF
    call quit
```

Figure 10:

: ,
a=17, b=23, c=45.

```
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ nasm -f elf min.asm  
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ ld -m elf_i386 min.o -o min  
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ ./min  
Наименьшее число: 17
```

Figure 11:

: 17,
.

2: f(x)

```
GNU nano 7.2
#include 'in_out.asm'

SECTION .data
msg_x: DB "Введите x: ",0
msg_a: DB "Введите a: ",0
msg_result: DB "Результат: ",0

SECTION .bss
x: RESB 10
a: RESB 10
result: RESB 10

SECTION .text
GLOBAL _start
_start:
    mov eax, msg_x
    call sprint
    mov ecx, x
    mov edx, 10
    call sread

    mov eax, msg_a
    call sprint
    mov ecx, a
    mov edx, 10
    call sread

    mov eax, x
    call atoi
    mov [x], eax

    mov eax, a
    call atoi
    mov [a], eax

    mov ebx, [x]
    mov ecx, [a]

    cmp ebx, ecx
    jge else_case

    ; if x < a: result = 2a - x
    mov eax, ecx
    imul eax, 2
    sub eax, ebx
    jmp end_if

else_case:
    ; else: result = 8
    mov eax, 8
```

```
GNU nano 7.2

SECTION .text
GLOBAL _start
_start:
    mov eax, msg_x
    call sprint
    mov ecx, x
    mov edx, 10
    call sread

    mov eax, msg_a
    call sprint
    mov ecx, a
    mov edx, 10
    call sread

    mov eax, x
    call atoi
    mov [x], eax

    mov eax, a
    call atoi
    mov [a], eax

    mov ebx, [x]
    mov ecx, [a]

    cmp ebx, ecx
    jge else_case

    ; if x < a: result = 2a - x
    mov eax, ecx
    imul eax, 2
    sub eax, ebx
    jmp end_if

else_case:
    ; else: result = 8
    mov eax, 8

end_if:
    mov [result], eax

    mov eax, msg_result
    call sprint
    mov eax, [result]
    call iprintLF

    call quit
```


: , $f(x)$ x a,

1: (x=1, a=2)

```
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ touch function.asm
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ nano function.asm

(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ nasm -f elf function.asm
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ ld -m elf_i386 function.o -
o function
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ ./ function
-bash: ./: Is a directory
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ ./function
Введите x: 1
Введите a: 2
Результат: 3
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ ./function
Введите x: 2
Введите a: 1
Результат: 8
```

Figure 12: 1

: 1: x=1, a=2, $x < a$, $f(x) = 2 \times 2 - 1 = 3$,
3.

2: (x=2, a=1)

```
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ touch function.asm
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ nano function.asm

(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ nasm -f elf function.asm
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ ld -m elf_i386 function.o -
o function
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ ./ function
-bash: ./: Is a directory
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ ./function
Введите x: 1
Введите a: 2
Результат: 3
(base) hhl@LAPTOP-058L8DAA:~/work/study/2025-2026/архитектура компьютера/arch-pc/labs/lab07$ ./function
Введите x: 2
Введите a: 1
Результат: 8
```

Figure 13: 2

: 2: x=2, a=1, $x \geq a$, $f(x) = 8$,
else.

1. ? .

```

-      jmp,          .      : jmp
_label1      _label1      .

```

2. ?

(OF) . , (ZF), (CF), (SF),

3. cmp?

cmp , op2 - op1, ,
.

4. ?

5. : je label

label,	ZF=1,	0.
--------	-------	----

6. ?

```
: nasm -f elf -l filename.lst filename.asm
```

7. ?

8. ?

ZF (), CF (), SF (), OF ().

