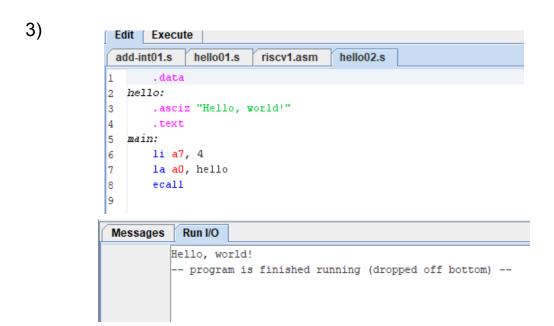
а) Скриншоты, демонстрирующие работу эмулятора с программами.

```
add-int01.s
1)
                      1i
                              а7 5 # Систепаний вызов N^05 — ввести десятичное число
                       ecall
                                        # Результат — в регистре а0
             2
             3
                       mv
                              t0 a0
                                        # Сохраняем результат в t0
                       ecall
                                        # Регистр а7 не менялся, тот же системный вызов
                       add
                              a0 t0 a0
                                        # Прибавляем ко второму число первое
                      li
                              a7 1
                                        # Системный вызов №1 — вывести десятичное число
             6
             7
                       ecall
             8
                       1i
                              a7 10
                                        # Системный вызов N^0\!\!=\!\!10 — останов программы
             9
                       ecall
             10
              Messages Run I/O
                        22
                        44
                        66
                        -- program is finished running (0) --
                Clear
2)
             add-int01.s
                             hello01.s
                 .text
             1
                           la aO, string
             2
                                                   # buffer
                                                   # syscall write (4)
                           li a7, 4
             3
                           ecall
             4
             5
                           li a0, 0
                                                   # exit code
             6
                           li a7, 10
                                                   # syscall exit
             7
                           ecall
                .data
             8
             9
                      string: .asciz "Hello! It works!!!\n"
            10
              Messages
                           Run I/O
                         Hello! It works!!!
                         -- program is finished running (0) --
```



```
4)
            add-int01.s hello01.s
                                 riscv1.asm
                                           hello02.s
                                                      hello03.s
               .text
            2
                      la aO, string # buffer
                      li a7, 4
            3
                                        # syscall write (4)
            4 .data
                  string: .asciz "Hello! It works!!!\n"
            5
            6
              .text
            7
                      ecall
            8
                      li aO, O
                                       # exit code
            9
                      li a7, 10
                                       # syscall exit
           10
                      ecall
           11
            Messages Run I/O
                     Hello! It works!!!
                     -- program is finished running (0) --
```

```
Edit Execute
5)
             add-int01.s hello01.s riscv1.asm hello02.s hello03.s hello-ru.s
                .text
                       la aO, string
                                         # buffer
                       li a7, 4
                                         # syscall write (4)
             3
                       ecall
                       li aO, O
                                         # exit code
                       li a7, 10
                                         # syscall exit
             7
                       ecall
             8
                .data
                   string: .asciz "Привет. Русский язык выглядит так!!!\n"
             9
             10
             Messages
                         Run I/O
                        Привет. Русский язык выглядит так!!!
                        -- program is finished running (0) --
```

```
Edit Execute
 add-int01.s hello01.s riscv1.asm hello02.s hello03.s hello-ru.s
                                                               add-int02.s
       arg02: .asciz "Input 2nd number: "
3
       result: .asciz "Result = "
      ln: .asciz "\n"
5
6
   .text
7
           la
                  aO, argOl # Подсказка для ввода первого числа
8
          li
                  a7, 4
                             # Системный вызов №4
          ecall
9
          1i
                  a7 5
                             # Системный вызов №5 — ввести десятичное число
10
11
           ecall
                             # Результат — в регистре а0
                  t0 a0
                             # Сохраняем результат в t0
12
          mv
13
                  aO, argO2 # Подсказка для ввода второго числа
14
          la
          1i
                             # Системный вызов №4
15
                  a7, 4
          ecall
16
17
          li
                  a7 5
                             # Системный вызов N^{o}5 — ввести десятичное число
          ecall
                             # Результат — в регистре а0
18
19
           mv
                  tl a0
                             # Сохраняем результат в t1
20
          la aO, result
                             # Подсказка для выводимого результата
21
                             # Системный вызов №4
          li a7, 4
22
23
           ecall
24
           add
                  aO tO tl # Складываем два числа
           1i
25
                  a7 1
                             # Системный вызов №1 — вывести десятичное число
26
           ecall
27
          la aO, ln
                             # Перевод строки
28
           li a7, 4
                             # Системный вызов №4
29
           ecall
30
31
32
           li
                 a7 10
                            # Систепаний вызов №10 — останов програнили
33
           ecall
34
```

```
Messages Run I/O

Input 1st number: 13
Input 2nd number: 34
Result = 47
-- program is finished running (0) --
```

- **b)** Описание, какие команды являются псевдокомандами,
- + анализ результатов компиляции одной из программ.

Псевдокоманды из программ:

• li	addi x17,x0,5	1:	li	a7 5	
• mv	add x5,x0,x10	3:	mv	t0 a0	
• la	auipc x10,64528	7:	la a0, hel	.10	
	addi x10,x10,-4				

Анализ результатов компиляции 1-ой программы:

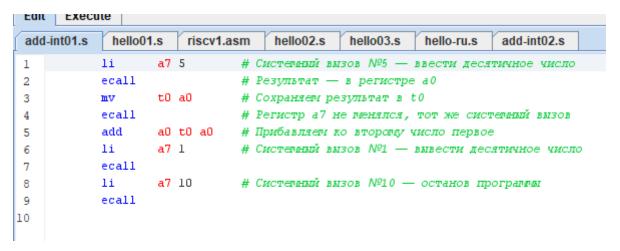
Basic		Source			
addi x17,x0,5	1:	li a7 5	# Системный вызов №5		
ecall	2:	ecall	# Результат — в реги		
add x5,x0,x10	3:	mv t0 a0	# Сохраняем результат		
ecall	4:	ecall	# Регистр а7 не менял		
add x10,x5,x10	5:	add a0 t0	а0 # Прибавляем ко второ		
addi x17,x0,1	6:	li a7 l	# Системный вызов №1		
ecall	7:	ecall			
addi x17,x0,10	8:	li a7 10	# Системный вызов №1		
ecall	9:	ecall			

- 1) для выполнения команды **li a7 5** вызывается команда **addi x17**, **x0**, **5**
- 2) для выполнения команды **mv t0 a0** вызывается команда **add x5, x0, x10**
- 3) для выполнения команды **li a7 1** вызывается команда **addi x17**, **x0**, **1**
- 4) для выполнения команды **li a7 10** вызывается команда **addi x17**, **x0**, **10**

команды li и mv в данной программе являются псевдокомандами, так как для их выполнения вызываются другие команды

с) Описание типов форматов команд для одной из представленных программ.

Первая программа



Basic		Source			
addi x17,x0,5	1:	li	a7 5	# Систев	
3 ecall	2:	ecall		# Резуль	
add x5,x0,x10	3:	mv	t0 a0	# Coxpan	
3 ecall	4:	ecall		# Региса	
add x10,x5,x10	5:	add	a0 t0 a0	# Прибав	
addi x17,x0,1	6:	li	a7 l	# Систем	
3 ecall	7:	ecall			
addi x17,x0,10	8:	li	a7 10	# Систем	
ecall	9:	ecall			

- 1) **addi x17,x0,5** тип I (immediate). Операнды регистр-регистр-непосредственное значение.
- 2) add x5,x0,x10 тип R (register). В роли операндов используются три регистра регистр назначения, первый аргумент и второй аргумент.
- 3) **add x10,x5,x10** тип R (register). В роли операндов используются три регистра регистр назначения, первый аргумент и второй аргумент.
- 4) **addi x17,x0,1** тип I (immediate). Операнды регистр-регистр-непосредственное значение.
- 5) **addi x17,x0,10** тип I (immediate). Операнды регистр-регистр-непосредственное значение.

d) Описание, какие системные вызовы используются в изученных программах.

- 1 вывод целого числа
- 4 вывод нуль терминированной строки
- 5 чтение целого числа
- 10 завершение программы с кодом 0