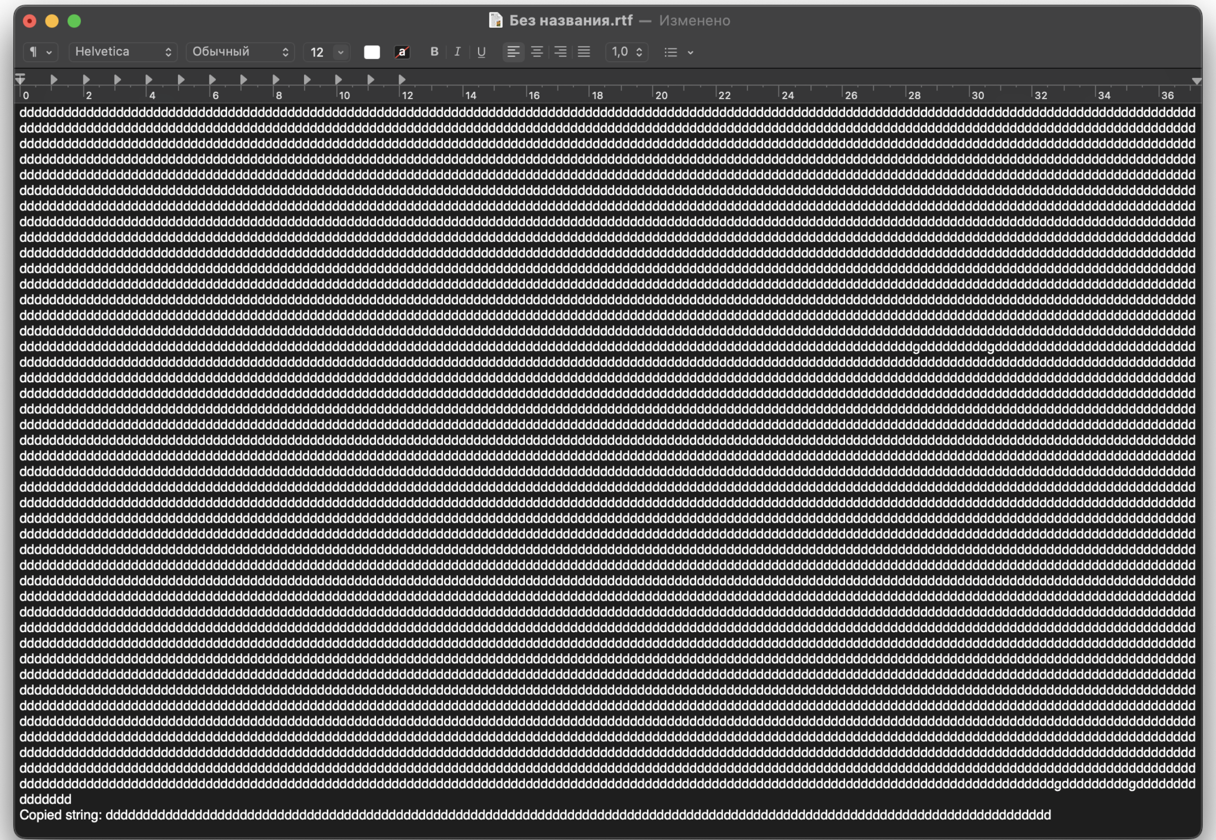
Дз 6 строки

Отчет по пользовательским тестам

```
Enter source string: **** user input : DəÎƒə∑´¨ˆˆ°Δ˜˜√÷æ…¬≤µ˜˜πøˆ¨¥
Copied string: DəÎƒə∑´¨ˆˆ°Δ˜˜√÷æ…¬≤µ˜˜πøˆ¨¥

Copied string: Hello, RISC-V!
Copied string: Assembly programming
Copied string: 1234567890

-- program is finished running (0) --
```

```
Enter source string: **** user input : 👌👌👌👌🤯🤯🤯🤯
Copied string: 👌👌👌👌🤯🤯🤯🤯

Copied string: Hello, RISC-V!
Copied string: Assembly programming
Copied string: 1234567890

-- program is finished running (0) --
```

Тест на строку длинной более 127 символов. Эмулятор тормозит и выводит ровно допустимое количество, обрезая строку.



может реализовать тестовый модуль?

in: test string

(Хотелось, но неуспелось)

Код

```
.include "strcopy.s"
.include "macros.s"
.include "data.s"

.text
.global main

main:
    # test string input
    print_string msg_src

    la a0, buffer_src
    li a1, 128
    read_string()

    la t0, buffer_dest
    la t1, buffer_src
    strcpy t0, t1
```

```
    print_string msg_res
    print_string buffer_dest
    print_string msg_newline

    j test1

    # test string
test1:
    la t0, buffer_dest
    la t1, test1_src
    strcpy t0, t1

    print_string msg_res
    print_string buffer_dest
    print_string msg_newline

    j test2

    # test another string
test2:
    la t0, buffer_dest
    la t1, test2_src
    strcpy t0, t1

    print_string msg_res
    print_string buffer_dest
    print_string msg_newline

    j test3

    # test numbers
test3:
    la t0, buffer_dest
    la t1, test3_src
    strcpy t0, t1

    print_string msg_res
    print_string buffer_dest
    print_string msg_newline

    j end_main

end_main:
    li a7, 10
    ecall
```

```
.data
msg_src:        .asciz "Enter source string: "
```

```
msg_res:        .asciz "Copied string: "
msg_newline:    .asciz "\n"
buffer_src:     .space 128      # Буфер для ввода строки
buffer_dest:    .space 128      # Буфер для хранения результата

test1_src:      .asciz "Hello, RISC-V!"
test2_src:      .asciz "Assembly programming"
test3_src:      .asciz "1234567890"
```

```
.macro strcpy %dest, %src
    mv a0, %dest  # Передаем адрес назначения
        mv a1, %src   # Передаем адрес источника
    jal strcopy   # Вызываем подпрограмму
.end_macro

# Макрос для вывода строки
.macro print_string %str
        la a0, %str
        li a7, 4
        ecall
.end_macro

# Макрос для чтения строки
.macro read_string()
        li a7, 8
        ecall
.end_macro
```

```
.text
.global strcopy

# strcopy: копирует строку из %src в %dest
# Входные параметры:
# a0: адрес назначения (dest)
# a1: адрес источника (src)
# Выход:
# Строка скопирована в dest, включая '\0'

strcopy:
        add t0, a0, zero  # Указатель на dest
        add t1, a1, zero  # Указатель на src

copy_loop:
        lb t2, 0(t1)      # Читаем символ из src
        sb t2, 0(t0)      # Пишем символ в dest
        beqz t2, end      # Если символ == '\0', завершить
        addi t0, t0, 1    # Увеличиваем указатель dest
        addi t1, t1, 1    # Увеличиваем указатель src
```

```
        j copy_loop

end:
        ret
```