

# Пример программы для RISC-V

# Формулировка задания

Перестановка местами элементов массива слов с четными и нечетными индексами.

Массив данных и другие параметры располагаются в памяти по фиксированным адресам.

# Программа на языке C

```
#include <stddef.h>

static unsigned array[] = {
    0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
};

static const size_t array_length =
    sizeof( array ) / sizeof( array[ 0 ] );

int main( void ) {
    for( size_t i = 1; i < array_length; i += 2 ) {
        const unsigned t = array[ i - 1 ];
        array[ i - 1 ] = array[ i ];
        array[ i ] = t;
    }

    return 0;
}
```

# Программа на языке ассемблера RISC-V (1)

```
# static unsigned array[] = {  
#     0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10  
# };  
.data  
array:  
    .word 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10  
  
# static const size_t array_length =  
#     sizeof( array ) / sizeof( array[ 0 ] );  
.rodata  
array_length:  
    .word 11
```

# Программа на языке ассемблера RISC-V (2)

```
.text
start:
.globl start

    la x10, array           # pseudo => auipc + addi
    lw x11, array_length    # pseudo => auipc + lw

    # for( size_t i = 1; ...
    li x12, 1               # pseudo => addi
loop:
    ...
```

# Программа на языке ассемблера RISC-V (3)

```
loop:
# for( ...; i < array_length; ...) {
    bgeu x12, x11, loop_exit

...

# }
loop_exit:
...
```

# Программа на языке ассемблера RISC-V (4)

loop:

...

*# const unsigned t = array[ i - 1 ];*

*# array[ i - 1 ] = array[ i ];*

*# array[ i ] = t;*

slli x13, x12, 2     # x13 = x12 << 2 = x12 \* 4

add x13, x10, x15    # x13 = x10 + x15 = x10 + x12 \* 4

lw x6, -4(x13)       # t1 = array[i-1]

lw x5, 0(x13)        # t0 = array[i]

sw x6, 0(x13)        # array[i] = t1

sw x5, -4(x13)       # array[i-1] = t0

...

loop\_exit:

# Программа на языке ассемблера RISC-V (5)

```
loop:
```

```
...
```

```
# for( ...; i += 2 )
```

```
    addi x12, x12, 2
```

```
# }
```

```
    jal x0, loop
```

```
loop_exit:
```

```
...
```



# Программа на языке ассемблера RISC-V (6)

```
loop_exit:
```

```
# return 0;
```

```
# }
```

```
finish:
```

```
    li x10, 17
```

```
    li x11, 0
```

```
    ecall
```