

Esercizi su programmazione Risc-V 32 I: 13/11/2024

Partendo dal template indicato di seguito definire un programma Risc-V che data una sequenza di numeri terminata da -1 già allocata in memoria (vedi esempi: arr, arr1, arr2...) elimina tutti le occorrenze di 0 spostando i numeri rimanenti a sinistra senza cambiarne l'ordine e rimanendo nello stessa area di memoria

Esempi

arr: 0 0 2 0 0 -1 → arr: 2 -1

arr1: 4 0 0 2 0 0 0 0 -1 → arr: 4 2 -1

...

Template Risc-V

```
.data
```

```
arr: .word 0 -1
```

```
arr1: .word 1 -1
```

```
arr2: .word 0 0 2 0 0 -1
```

```
arr3: .word 0 3 0 0 3 0 3 0 -1
```

```
arr4: .word 4 4 0 0 0 0 0 -1
```

```
arr5: .word 5 5 5 0 0 0 5 5 5 -1
```

```
msg: .string " "
```

```
.macro print_str (%x)
```

```
    li a7, 4
```

```
    la a0, %x
```

```
    ecall
```

```
.end_macro
```

```
.macro print_int (%x)
```

```
    li a7, 1
```

```
    mv a0, %x
```

```
    ecall
```

```
.end_macro
```

```
.text
```

```
MAIN:
```

```
    # TEST
```

```
    la s4, arr
```

```
    #la s4, arr1
```

```
    #la s4, arr2
```

#la s4,arr3

#la s4,arr4

#la s4,arr4

#...

VERY_END: