

Zad 7

```
0000 0000 0001 0000 1000 0000 1011 0011
011 0011 -> opkod dla OP, format R
funct7 | rs2 | rs1 | funct3 | rd | opkod
funct7 -> 0000000
rs2     -> 00001 = 1
rs1     -> 00001 = 1
funct3  -> 000
rd      -> 00001 = 1
czyli funct3, funct7 -> instrukcja to add
add x1, x1, x1
```

Zad 9

Będziemy dla każdej cyfry w napisie brać jej wartość -'0', (48 w ASCII).
Aby nie liczyć długości napisu będziemy zwiększać sumę stopniowo, tj. w każdym kroku: $\text{suma} = \text{suma} \times 10 + (\text{wartość obecnej cyfry})$
np. $123 = (((10 \times 0 + 1) \times 10 + 2) \times 10 + 3)$
Początkowo $\text{suma}(x6)$ jest równa zero. Zapiszmy też wartość stałej $48(x7)$ do odejmowania, i $10(x28)$ do mnożenia.

```
addi x6, x0, 0      # suma
addi x7, x0, 48     # '0'
addi x28, x0, 10     # 10
loop: lb x5, 0(x10)  # wczytaj znak
      beq x5, x0, end # znak == \null -> koniec
      sub x5, x5, x7  # znak - '0'
      mult x6, x6, x28 # suma * 10
      add x6, x6, x5   # suma + wart(znak)
      addi x10, x10, 1 # kolejny znak
      j loop
end:  mv x10, x28      # na koniec a0 = suma
```