

Registri

Esercizio 1

Si realizzi un file register con 4 registri a 8 bit che abbia un unico input e un unico output. Il circuito deve avere un bit per *write enable*, uno per *read enable* e un indice per poter selezionare il registro che si vuole utilizzare.

Assembly

Esercizio 2

Si descriva, tramite linguaggio naturale, lo scopo del seguente frammento di codice assembly.

```
      MOV    R1, #10
      MOV    R2, #0
loop  CMP    R2, R1
      BEQ    end_loop
      ADD    R2, R2, #1
      ADD    R0, R0, R2
      B      loop
end_loop
```

Esercizio 3

Si descriva, tramite linguaggio naturale, lo scopo del seguente frammento di codice assembly.

```
      MOV    R1, #13
      MOV    R2, #4
      MOV    R3, #0
loop  CMP    R3, R2
      BEQ    end_loop
      ADD    R0, R0, R1
      ADD    R3, R3, #1
      B      loop
end_loop
```

Esercizio 4

Si traduca il seguente frammento di codice C nel linguaggio assembly. Il risultato deve essere salvato in *R0*.

```
int main(int argc, char** argv) {
    int firstNumber = 88;
    int secondNumber = 24;
```

```
int result = 0;
if (firstNumber >= secondNumber) {
    result = firstNumber + secondNumber;
} else {
    result = secondNumber - firstNumber;
}
}
```

Esercizio 5

Si traduca il seguente frammento di codice C nel linguaggio assembly. Il risultato deve essere salvato in *R0*.

```
int main(int argc, char** argv) {
    int myArray[] = {21, 5, 66, 14, 37};
    int i = 0;
    int result = 0;
    while (i < 5) {
        if (myArray[i] >= 30) {
            result = result + myArray[i];
        }
        i = i + 1;
    }
}
```

Esercizio 6

Dato un vettore di lunghezza 4, i cui elementi appartengono a \mathbb{Z} , si trovi il valore massimo e lo si memorizzi nel registro *R0*.

Esercizio 7

Dato un numero naturale n , si sommino i primi $n \in \mathbb{N} \setminus \{0\}$ numeri tramite la formula di Gauss, senza utilizzare le istruzioni di moltiplicazione e divisione. Il risultato deve essere salvato in *R0*.