

Scrivere una procedura RISC-V **isupr** che riceva l'indirizzo di un byte in memoria (nel registro **a0**) e restituisca 1 se il byte rappresenta un carattere maiuscolo nella tabella ASCII, e 0 in caso contrario. Il valore di ritorno deve essere lasciato nel registro **a0**.

I caratteri maiuscoli A-Z hanno come valore nella tabella ASCII da 65 a 90 (in decimale, estremi inclusi).

- Incollare nel campo di sotto soltanto il contenuto della funzione partendo dall'etichetta **isupr**:
- Attenzione alle convenzioni di chiamata!
- Usare il seguente codice "main" per sviluppo e debugging nel simulatore RARS

```
.globl _start
.data
    ch: .string "A"
.text
_start:
    # call isupr
    la    a0, ch
    jal   ra, isupr

    # completare la funzione nel campo di sotto
```

Scrivere una funzione RISC-V **uppercase** che riceva l'indirizzo di una stringa in memoria (nel registro **a0**) e ritorni 1 se la stringa è scritta completamente in maiuscolo, 0 in caso contrario. Il valore di ritorno deve essere lasciato nel registro **a0**.

Le stringhe di caratteri in C sono memorizzate come un array di byte in memoria, dove il byte **0x00** rappresenta la fine della stringa.

**uppercase** deve utilizzare la funzione **isupr** sviluppata nell'esercizio precedente.

- **NON INCOLLARE IL CODICE DI isupr NELLA RISPOSTA**
- Incollare nel campo di sotto soltanto il contenuto della funzione **uppercase**
- Attenzione alle convenzioni di chiamata!
- Usare il seguente codice "main" per sviluppo e debugging nel simulatore RARS

```
.globl _start
.data
    src: .string "COMPUTER"
.text

# NON INCOLLARE IL CODICE DI isupr NELLA RISPOSTA.
#isupr:
    # soluzione dell'esercizio #1

_start:
    # call uppercase
    la    a0, src
    jal   ra, uppercase

    # completare la funzione nel campo di sotto
```

Scrivere una procedura RISC-V *inrange* che riceva l'indirizzo di una parola in memoria (nel registro **a0**) e ritorni 1 rappresentato da questa parola (in complemento a 2) può essere memorizzato in un singolo byte (sempre in complemento a 2).

Il valore di ritorno deve essere lasciato nel registro **a0**.

- Incollare nel campo di sotto soltanto il contenuto della funzione partendo dall'etichetta *inrange*:
- Attenzione alle convenzioni di chiamata!
- Usare il seguente codice "main" per sviluppo e debugging nel simulatore RARS

```
.globl _start
.data
    nbr: .word -128
.text
_start:
    # call inrange
    la  a0, nbr
    jal ra, inrange

    # completare la funzione nel campo di sotto
```

Scrivere una funzione RISC-V *bytearray* che riceva nel registro **a0** l'indirizzo di un array di parole in memoria (numeri interi in complemento a 2) e nel registro **a1** la lunghezza di questo array. La funzione deve restituire il numero di elementi nell'array che possano essere rappresentati con un singolo byte (in complemento a 2).

Il valore di ritorno deve essere lasciato nel registro **a0**.

***bytearray* deve utilizzare la funzione *inrange* sviluppata nell'esercizio precedente.**

- **NON INCOLLARE IL CODICE DI *inrange* NELLA RISPOSTA**
- Incollare nel campo di sotto soltanto il contenuto della funzione *bytearray*
- Attenzione alle convenzioni di chiamata!
- Usare il seguente codice "main" per sviluppo e debugging nel simulatore RARS

```
.globl _start
.data
    array: .word 127,6,12,-3,-128
    size: .word 5
.text

# NON INCOLLARE IL CODICE DI divt NELLA RISPOSTA.
#divt:
    # soluzione dell'esercizio #1

_start:
    # call bytearray
    la  a0, array
    la  t0, size
    lw  a1, 0(t0)
    jal ra, bytearray
```