Lab 4

Operandi Allocati in Memoria e array

Esercizio 1 - Max in memoria

Si scriva un programma in linguaggio RISC-V che calcoli il massimo tra tre numeri interi presenti nella memoria in word contigue. Il valore massimo va salvato in un’ulteriore posizione della memoria contigua a quelle usate per il calcolo.

Esercizio 2 - Somma Array

Data una word in memoria che contiene l'intero X, seguita da X interi in **word** contigue (array di **word**), scrivere un programma RISC-V che stampi a video la somma degli elementi dell'array. Il valore ottenuto va anche salvato in una variabile identificata con l’etichetta "result".

Esercizio 3 - Max Array

Data una word in memoria che contiene l'intero X, seguita da X interi in **word** contigue (array di **word**), scrivere un programma RISC-V che stampi a video il massimo tra gli interi nell'array.

Esercizio 4 - Numero di Dispari

Data una word in memoria che contiene l'intero X, seguita da X interi in **byte** contigue (**array di byte**), scrivere un programma RISC-V che stampi a video il numero totale di dispari tra gli X interi. Il valore ottenuto va anche salvato in una variabile identificata con l’etichetta "result".

Esercizio 5 - sorted

Scrivere il codice RISC-V che verifichi se un **array** di "**size**" elementi in memoria (**double word** contigue) contiene una sequenza ordinata di numeri interi (crescente).

Il risultato va lasciato nella variabile result.

```
result = 1;
for (int i = 0; i < size-1; i++) {
    if (array[i] > array[i+1]) {
            result = 0;
                    break;
    }
}
```

Esercizio 6 - Invert

Scrivere il codice per invertire un **array di double word** in memoria (**size** elementi).

```
for (int i = 0; i < size/2; i++)  {
    long temp = array[i];
    array[i] = array[size-i-1];
    array[size-i-1] = temp;
}
```