

Laboratorio di Architettura degli Elaboratori II

Corso di laurea triennale in Informatica
Università degli Studi di Milano, A.A. 2018-2019

Marco Tarini, Nicola Basilico

Appello del 20 Settembre 2019

- L'esame ha una durata di 3 ore.
- È possibile consultare il libro di testo, appunti e la documentazione di MARS o Spim.
- È proibito l'accesso ad Internet con qualsiasi mezzo.
- Verranno corretti solo gli esercizi che non generano errori in compilazione ed esecuzione.
- I sorgenti vanno uploadati su <https://upload.di.unimi.it/>

Ciascun esercizio richiede la stesura di una procedura, che deve essere invocabile dall'esterno. Le quattro procedure devono essere chiamate `procedura1`, `procedura2`, `procedura3`, e `procedura4`.

1 Paragona e scambia

Fornire una procedura `procedura1`, che dato in input un array di numeri e due indici, scambi fra di loro i due elementi dell'array se il primo è maggiore del secondo. Nel caso in cui uno dei due indici non sia valido, la procedura stampa un messaggio di errore e restituisce -1.

Si noti che, se i due elementi sono uguali, non ha ovviamente alcuna importanza se vengono scambiati oppure no.

2 Sì e no

Fornire una procedura `procedura2` che, preso in input l'indirizzo in memoria di una stringa (terminata da 0), trasformi ogni occorrenza della parola "no" (minuscola) in un "si". Si considera una parola "no" qualsiasi occorrenza delle due lettere 'n' e 'o' in successione.

Info: il codice ASCII del carattere 'a' è (in decimale) 97, la I è la nona lettera dell'alfabeto, la N è la quattordicesima, la O la quindicesima, e la S è la diciannovesima.

Per esempio, la stringa “no ai nobili!” deve diventare “si ai sibili!”.

3 Il numero più bello

É definita una procedura `nicer`, fornita dall'esterno, che identifica quale, fra due numeri dati, sia più “bello”, secondo un criterio estetico che non conosciamo.

Questa procedura, che ci verrà fornita in un secondo momento, seguirà le seguenti specifiche (oltre alle convenzioni del MIPS):

- input 1: il numero intero a
- input 2: il numero intero b
- output: -1 se a è più bello di b , $+1$ se b è più bello di a , e 0 in caso di parità.

Produrre una procedura `procedura3` che, dato un array A di n numeri, restituisca

- $+1$ se A è in ordine *strettamente crescente* di bellezza;
- -1 se è in ordine *strettamente decrescente* di bellezza;
- 0 altrimenti

La procedura prodotta prende i seguenti input:

- input 1: l'indirizzo dell'origine dell'array A (base address)
- input 2: il numero di elementi n , che è sempre maggiore di 1

Suggerimento: al fine di testare la propria soluzione, produci un esempio della funzione `nicer` (ricordati però di rimuoverlo alla fine!).

4 Quanti divisori

Sia A l'insieme $\{2, 3, 4, 5, 7\}$. Implementare una procedura `procedura4` che, dato un numero intero n diverso da 0, restituisce il numero degli elementi di A che dividono n .

Per esempio, se viene passato il numero 360, la procedura restituisce il valore 4, se viene passato il numero 17 la procedura restituisce il valore 0.