



Elaborato Assembly

Si consideri il sorgente `GestioneVettore.c` allegato al presente testo implementante una semplice interfaccia testuale per operare su un vettore di interi. Il programma, completamente funzionante, può essere compilato ed eseguito da terminale nel seguente modo:

```
$ gcc gestioneVettore.c -o gestioneVettore
$ ./gestioneVettore
```

Una volta eseguito il programma, il software prima chiede 10 numeri di tipo `int` all'utente per popolare un vettore e poi visualizza un menù testuale di scelta attraverso il quale l'utente può selezionare diverse operazioni tra cui ricerca di un valore, calcolo del valore minimo/massimo inserito, ecc.

Per consegnare l'elaborato, lo studente deve implementare una versione equivalente del programma fornito in codice assembly. L'implementazione prodotta può essere interamente basata su codice assembly o può fare uso di funzioni C (tipo `printf` o `scanf`), come spiegato nell'ultima lezione del corso.

Le nozioni di assembly necessarie per completare l'elaborato e che non sono state trattate a lezione possono essere prese in autonomia facendo riferimento al manuale di assembly caricato sull'e-learning durante la prima lezione del corso. In particolare, la sezione del testo che spiega come utilizzare i vettori è la 3.18 a pagina 54.

È importante notare che l'implementazione C fornita deve essere usata come traccia, ma che durante l'implementazione in assembly i vari algoritmi potranno essere scritti in modo diverso, purché il risultato finale delle operazioni sia lo stesso. Eventuali differenze tra l'implementazione C e quella assembly prodotta od eventuali assunzioni prese durante l'implementazione dovranno essere evidenziate e descritte nella relazione finale consegnata assieme al programma.

In fase di correzione, saranno valutati meglio progetti che ottimizzeranno meglio il codice, ovvero programmi che dimostreranno una migliore leggibilità del codice od una minore complessità in tempo (minore tempo di esecuzione a parità di hardware). Oltre che il codice prodotto, anche la relazione sarà valutata sia in termini di correttezza di lingua italiana, sia in termini di chiarezza espositiva.

Durante l'esame orale è possibile che venga richiesto di operare delle modifiche al codice sul momento.

Il progetto è pensato per essere svolto come attività di gruppo. I gruppi potranno essere costituiti da un minimo di uno studente ad un massimo di tre. Anche se è certamente possibile produrre l'elaborato da soli, essere in grado di lavorare in team è una caratteristica importante da sviluppare e molto apprezzata in ambito lavorativo. Gli studenti sono quindi **caldamente incoraggiati** a formare gruppi da almeno due persone.



Laboratorio di Architettura degli Elaboratori

A.A. 2018/19

Modalità di consegna

Tutto il materiale va consegnato elettronicamente tramite procedura guidata sul sito Moodle del corso. Indicativamente 15 giorni prima della data di consegna (ultima settimana di giugno/ prima di luglio) sarà attivata una apposita sezione sul sito con una chiara indicazione della procedura da seguire per la sottomissione dei lavori. La consegna del materiale comporta automaticamente l'iscrizione all'appello orale.

Il calendario degli orali sarà pubblicato sul sito Moodle secondo la procedura descritta all'apertura della sezione per la sottomissione. Indicativamente gli orali saranno pianificati nella settimana che parte con lunedì 15 luglio od in quella successiva.

Materiale da consegnare

Si richiede ad ogni gruppo di caricare un singolo archivio **.tgz** denominato `asm_cognome1_nome1_cognome2_nome2_cognome3_nome3.tgz`¹ contenente tutti i file di seguito elencati senza sottocartelle:

1. Il file `GestioneVettore.s` contenente l'implementazione assembly del programma
2. Tutti gli altri (eventuali) files sorgenti necessari al progetto (ad esempio funzioni assembly)
3. Makefile per la compilazione (non fornito, ogni gruppo deve scrivere il proprio)
4. Un file `Relazione.pdf` con una relazione del progetto che affronti nel dettaglio almeno i seguenti punti:
 - introduzione su come funziona in termini generali il programma;
 - le variabili utilizzate e il loro scopo;
 - le modalità di passaggio/restituzione dei valori delle funzioni create;
 - il diagramma di flusso o lo pseudo-codice ad alto livello del codice prodotto;
 - la descrizione delle scelte progettuali effettuate

Si ricorda che è possibile effettuare più sottomissioni, ma ogni nuova sottomissione cancella quella precedente. Ogni gruppo deve consegnare una sola volta il materiale, ovvero un solo membro del gruppo deve effettuare la sottomissione!

¹ Per generare correttamente un file `filename.tgz` in linux seguire la seguente procedura:

- posizionare tutti i file da comprimere in una cartella
- rinominare la cartella "filename"
- uscire dalla cartella e lanciare il comando `tar cvfz filename.tgz filename`