

Elaborato Assembly

ARCHITETTURA DEGLI ELABORATORI

CORSO DI LAUREA IN INFORMATICA, A.A. 2018/2019

Filippo Zeni (VR437690)

Giuseppe Brusco (VR429704)

Consegna

Si consideri il sorgente `GestioneVettore.c` allegato al presente testo implementante una semplice interfaccia testuale per operare su un vettore di interi.

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.

L'implementazione prodotta può essere interamente basata su codice assembly o può fare uso di funzioni C (tipo `printf` o `scanf`).

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.

Descrizione del progetto

Il programma si apre con una funzione main che richiede l'inserimento dei 10 valori nel vettore (è possibile aumentare o diminuire la grandezza del vettore: si può adattare in base alle esigenze). Per implementarlo è stato utilizzato un ciclo per inserire i vari elementi.

Successivamente viene chiamata una funzione che stampa a video il menù.

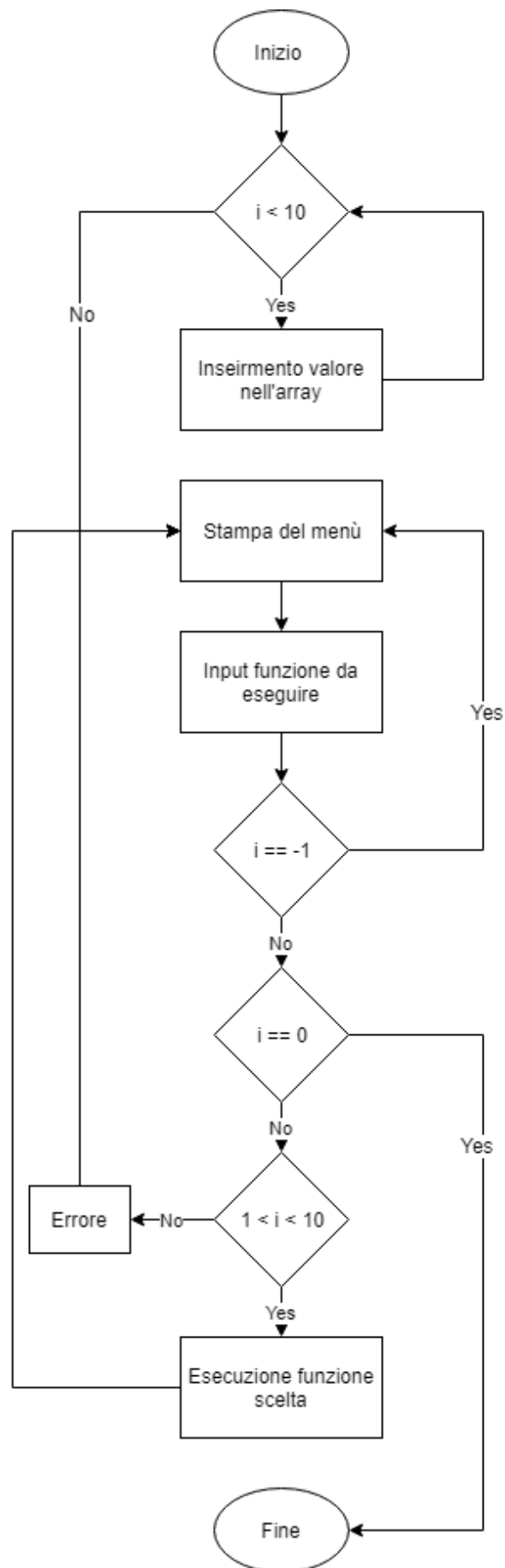
Dopo aver ricevuto in input dall'utente la scelta dell'operazione da eseguire, è stato implementato uno "switch" in Assembly che, in base al numero inserito, verrà eseguita una determinata funzione:

| Numero | Funzione |
|-----------------------|--|
| 0 | Chiusura programma |
| -1 | Vengono ristampate le opzioni e viene chiesto il reinserimento del numero |
| 1 | Stampa del vettore nell'ordine corretto e nell'ordine inverso |
| 2 | |
| 3 | Stampa la quantità di numeri pari e dispari |
| 4 | Ritorna la posizione del valore inserito |
| 5 | Stampa il valore massimo del vettore |
| 6 | Ritorna la posizione del valore massimo |
| 7 | Stampa il valore minimo del vettore |
| 8 | Ritorna la posizione del valore minimo |
| 9 | Stampa il valore con maggiore frequenza. In caso ce ne fosse più di uno, viene stampato il primo. |
| 10 | Calcola la media intera |
| Qualsiasi altro input | Viene stampato un messaggio di errore, vengono ristampate le opzioni e chiesto il reinserimento del numero |

Specifiche delle funzioni

- Nei punti 1 e 2, viene utilizzato il registro EDX come variabile booleana per determinare se effettuare la stampa in ordine corretto o inverso: nel primo caso EDX viene settato a 0, nel secondo viene settato a 1. Per eseguire queste due stampe, è stata scritta una sola funzione per ottimizzare il codice;
- Nel punto 3, l'implementazione calcola i numeri pari e, successivamente, questi vengono sottratti al numero totale per calcolare i numeri dispari;
- Nei punti 6 e 8 sono state riutilizzate rispettivamente le funzioni 4-5 e 4-7. Anche in questo caso viene utilizzato il registro EDX come variabile booleana: nel primo caso EDX viene settato a 0 quando vengono eseguite normalmente le funzioni 5 e 7. Nel secondo caso (cioè nei punti 6 e 8) viene settato a 1 in modo tale da utilizzare il valore di ritorno di 5 e 7 come input per 6 e 8.

Diagramma di flusso



Scelte progettuali

Il programma è stato opportunamente commentato su ogni riga per semplificarne lettura e comprensione. I commenti contengono i punti essenziali di questa relazione.

Abbiamo verificato che il programma funziona con numeri da 0 a 127.