



UNIVERSITÀ  
di **VERONA**

# Elaborato Assembly

## **Architettura degli Elaboratori**

*Timofte Robert Octavian - VR429581*

*Feola Emanuele - VR437205*

*Carra Mattia - VR429609*

## Indice

<b>Indice.....</b>	<b>2</b>
<b>Introduzione.....</b>	<b>3</b>
<b>Variabili.....</b>	<b>4-5</b>
<b>Funzioni.....</b>	<b>6-7</b>
<b>Diagramma di flusso.....</b>	<b>8</b>
<b>Scelte progettuali.....</b>	<b>9</b>

## Introduzione

L'elaborato consiste in una replica Assembly di un programma C.

Il funzionamento può essere riassunto nelle seguenti fasi:

- 1)** All'utente viene richiesto l'inserimento di 10 valori interi i quali vengono inseriti uno ad uno nel vettore che contiene i numeri
- 2)** Dopo aver inserito i valori, viene stampato un menù elencante le 10 operazioni effettuabili tramite il programma.  
All'utente viene chiesto di sceglierne una.

```
Inserimento dei 10 interi che compongono il vettore...
Inserire l'intero in posizione 1: 1
Inserire l'intero in posizione 2: 2
Inserire l'intero in posizione 3: 3
Inserire l'intero in posizione 4: 4
Inserire l'intero in posizione 5: 5
Inserire l'intero in posizione 6: 6
Inserire l'intero in posizione 7: 7
Inserire l'intero in posizione 8: 8
Inserire l'intero in posizione 9: 9
Inserire l'intero in posizione 10: 10

OPERAZIONI DISPONIBILI
-----
1) stampa a video del vettore inserito
2) stampa a video del vettore inserito in ordine inverso
3) stampa il numero di valori pari e dispari inseriti
4) stampa la posizione di un valore inserito dall'utente
5) stampa il massimo valore inserito
6) stampa la posizione del massimo valore inserito
7) stampa il minimo valore inserito
8) stampa la posizione del minimo valore inserito
9) stampa il valore inserito con maggior frequenza
10) stampa la media intera dei valori inseriti

Inserire valore operazione (0 uscita, -1 ristampa menu'): 1
```

- 3)** Se viene inserita un'opzione sconosciuta, un messaggio di errore comparirà ed il menù verrà ristampato.

In caso contrario l'utente può continuare a "giocare" con le opzioni disponibili fino all'inserimento del numero zero.

```
Inserire valore operazione (0 uscita, -1 ristampa menu'): 12
Opzione non supportata dall'applicazione!

Inserire valore operazione (0 uscita, -1 ristampa menu'): 0
Uscita dall'applicazione...
```

## Variabili

Procediamo ora elencando le variabili utilizzate dal programma Assembly e descrivendone brevemente lo scopo:

- **input\_num (long)** → serve per memorizzare un numero chiesto in input all'utente, principalmente viene usata durante l'inserimento dei valori nel vettore e nella scelta delle opzioni
- **length (long)** → serve per impostare la dimensione del vettore
- **max\_index (long)** → serve per impostare l'indice massimo del vettore
- **vector (long)** → vettore che serve per memorizzare i 10 numeri interi chiesti all'inizio del programma
- **dim\_error\_msg (string)** → utilizzata per stampare il messaggio che compare in caso di dimensione del vettore impostata in modo errato
- **start\_msg (string)** → utilizzata per stampare il messaggio iniziale
- **insert\_msg (string)** → utilizzata per stampare il messaggio che compare durante l'inserimento di ogni numero nel vettore
- **input\_format (string)** → utilizzata insieme ad *input\_num* per chiedere in input un valore intero
- **menu[1-12] (string)** → serie di stringhe usate per stampare il menù
- **option\_msg (string)** → utilizzata per stampare il messaggio che chiede la scelta di un'opzione da parte dell'utente
- **exit\_msg (string)** → utilizzata per stampare il messaggio di uscita dal programma
- **default\_msg (string)** → utilizzata per stampare il messaggio che compare quando viene inserita un'opzione sconosciuta
- **std\_msg (string)** → utilizzata per stampare il messaggio iniziale dell'opzione 1
- **std\_msg\_inv (string)** → utilizzata per stampare il messaggio iniziale dell'opzione 2
- **value\_msg (string)** → utilizzate durante la stampa dei valori presenti nel vettore quando vengono scelte le opzioni 1 e 2
- **even\_msg (string)** → utilizzata per stampare il numero di valori pari presenti nel vettore quando viene scelta l'opzione 3

- **odd\_msg (string)** → utilizzata per stampare il numero di valori dispari presenti nel vettore quando viene scelta l'opzione 3
- **search\_msg (string)** → utilizzata per stampare il messaggio che chiede all'utente un numero da cercare all'interno del vettore quando viene scelta l'opzione 4
- **found\_msg (string)** → utilizzata per stampare il messaggio che riporta la posizione del numero cercato tramite l'opzione 4 se esso viene trovato
- **not\_found\_msg (string)** → utilizzata per stampare il messaggio che compare quando non viene trovato il valore cercato tramite l'opzione 4
- **max\_value\_msg (string)** → utilizzata per stampare il messaggio che riporta il valore massimo presente nel vettore quando viene scelta l'opzione 5
- **max\_pos\_msg (string)** → utilizzata per stampare il messaggio che riporta la posizione del valore massimo all'interno del vettore quando viene scelta l'opzione 6
- **min\_value\_msg (string)** → utilizzata per stampare il messaggio che riporta il valore minimo presente nel vettore quando viene scelta l'opzione 7
- **min\_pos\_msg (string)** → utilizzata per stampare il messaggio che riporta la posizione del valore minimo all'interno del vettore quando viene scelta l'opzione 8
- **max\_freq\_msg (string)** → utilizzata per stampare il messaggio che riporta il valore che compare più frequentemente all'interno del vettore quando viene scelta l'opzione 9
- **mean\_msg (string)** → utilizzata per stampare il messaggio che riporta la media intera dei valori presenti all'interno del vettore quando viene scelta l'opzione 10

## Funzioni

Esaminiamo ora le funzioni utilizzate dal programma:

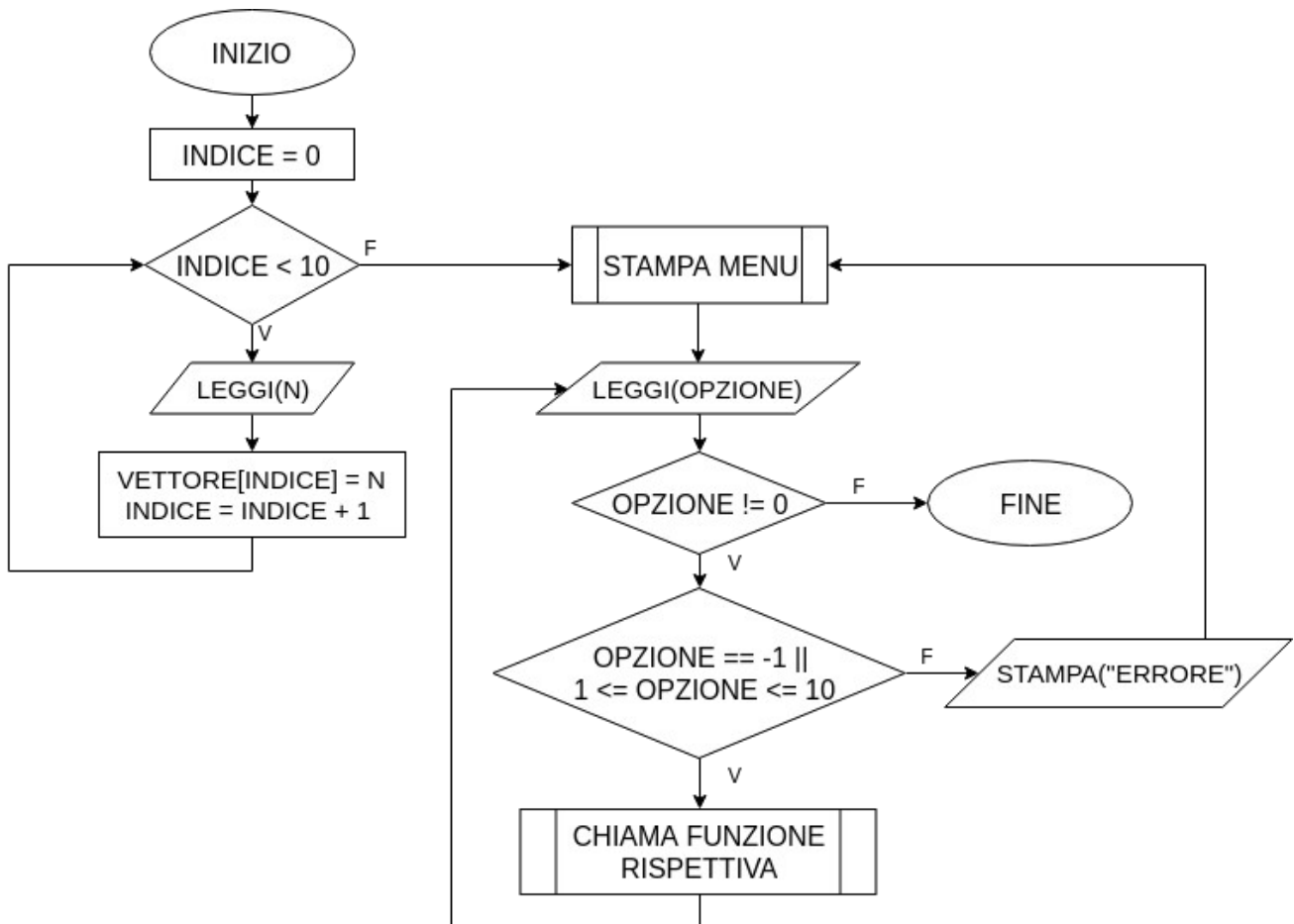
- **int getMaxIndex (int)** → riceve il parametro intero tramite la variabile *length* e restituisce nel registro ebx il massimo indice del vettore
- **void printOptions ()** → non riceve e non restituisce alcun parametro. Viene utilizzata per stampare il menù delle opzioni
- **void chooseOption ()** → non riceve e non restituisce alcun parametro. Continua a chiedere all'utente di scegliere un'opzione e la esegue (in caso di scelta sconosciuta stampa un messaggio di errore) fino a quando non viene inserito il numero 0
- **void printVector ()** → non riceve e non restituisce alcun parametro. Viene utilizzata per stampare i valori presenti nel vettore in ordine di inserimento
- **void printInverseVector ()** → non riceve e non restituisce alcun parametro. Viene utilizzata per stampare i valori presenti nel vettore in ordine di inserimento inverso
- **void printEvenOddNumber ()** → non riceve e non restituisce alcun parametro. Viene utilizzata per stampare il numero di valori pari e dispari presenti nel vettore
- **void findValue ()** → non riceve e non restituisce alcun parametro. Viene utilizzata per chiedere all'utente un valore da cercare e stampare un messaggio se quest'ultimo viene trovato o no
- **int getMaxValue ()** → non riceve alcun parametro ma restituisce un intero che corrisponde al valore massimo presente nel vettore. Quest'ultimo viene memorizzato nel registro eax
- **void printMaxValuePos (int)** → il parametro di tipo intero viene passato tramite il registro eax richiamando la funzione *getMaxValue*. Non restituisce nulla, serve per stampare la posizione del valore massimo all'interno del vettore
- **int getMinValue ()** → non riceve alcun parametro ma restituisce un intero che corrisponde al valore minimo presente nel vettore. Quest'ultimo viene memorizzato nel registro eax
- **void printMinValuePos (int)** → il parametro di tipo intero viene passato tramite il registro eax richiamando la funzione *getMinValue*. Non restituisce nulla, serve per stampare la posizione del valore minimo all'interno del vettore

- **void printMaxFreqValue ()** → non riceve e non restituisce alcun parametro. Viene utilizzata per stampare il valore che compare più frequentemente all'interno del vettore

- **void printMean ()** → non riceve e non restituisce alcun parametro. Viene utilizzata per stampare la media intera dei valori presenti nel vettore

## Diagramma di flusso

Il seguente diagramma di flusso rappresenta, in modo semplificato, il funzionamento del programma Assembly:





## Scelte progettuali

In questa sezione vengono elencate le varie scelte progettuali effettuate durante lo sviluppo del programma Assembly e le differenze tra quest'ultimo e la versione in C.

- 1)** Abbiamo deciso di sviluppare il programma usando un solo file per questioni di performance
- 2)** A differenza della versione in C, le funzioni implementate sono quasi tutte di tipo void, ovvero non ritornano alcun valore, questo perché abbiamo voluto evitare problemi con la gestione dei registri
- 3)** Come conseguenza del *punto 2* alcune funzioni risultano ripetitive, dato che all'interno di esse non vengono richiamate altre funzioni già sviluppate che potrebbero migliorare il funzionamento.  
Tuttavia dal punto di vista algoritmico, il programma Assembly rispecchia la versione in C
- 4)** Per quanto riguarda la dimensione del vettore abbiamo aggiunto la possibilità di poterla cambiare, tuttavia il valore *length* deve essere compreso tra 1 e 255 per evitare di generare errori
- 5)** Per poter gestire al meglio il programma abbiamo dovuto utilizzare una variabile per memorizzare l'indice massimo del vettore, il quale viene calcolato tramite una funzione che abbiamo aggiunto

Con ogni probabilità l'efficienza del programma Assembly può essere migliorata, tuttavia abbiamo deciso di consegnare una versione che secondo il nostro punto di vista rispecchia gli obiettivi imposti dalla consegna.