

# Status progetto ASM 16/05/2021

---

Prototipo 4 e 5 (`postfix_calculator4.py`, `postfix_calculator5.py`). Inizio scrittura funzione `postfix.s` (algoritmo principale in assembly)

## Fatto

- E' stata scritta la funzione per fare la divisione in assembly
- Sono stati caricati sul repository due file SVG con il diagramma di flusso delle funzioni che svolgono check di validita' dell'input (cartella assets)
- Sono stati scritti due nuovi prototipi (`postfix_calculator4.py`, `postfix_calculator5.py`).

Entrambi i prototipi adottano un approccio diverso nell'algoritmo per scorrere l'input (approccio simile ad una "macchina a stati")

Feature dei prototipi:

- tengono a mente quanti elementi sono stati messi nello stack durante i calcoli per poterli togliere in caso di errori di input
- la divisione per zero da come risultato "Invalid"
- restituisce "Invalid" quando un operando e' a fine espressione (l'ultimo elemento in una espressione ben formata e' un operatore)

`postfix_calculator5.py` in piu':

- riesce a gestire `\n` (se si va a capo nel file in input l'algoritmo funziona e riesce, se il resto della espressione e' corretto, a calcolare il risultato)
- non toglie elementi dallo stack se non sono stati aggiunti abbastanza elementi in precedenza (questo risolve casi che hanno input come `2 3 + 2`, notare gli spazi alla fine, dove il 2 finisce nello stack dopo il risultato di `2+3`: ci sono due elementi nello stack alla fine del calcolo ma puo' esistere solo un risultato dell'intera espressione)

## Da fare

- [OBIETTIVO PRINCIPALE] completare `postfix.s` (idealmente con l'algoritmo del prototipo 5)
- modificare la funzione per calcolare la divisione: attualmente i parametri dividendo e divisore sono adatti per l'uso in C ma non per poterla usare facilmente con lo stack

Occorre scambiare i due parametri

- scrivere test per testare l'intero programma (end to end tests)
- aggiungere altri unit test dove sara' necessario
- cercare di ottimizzare l'algoritmo di postfix dove possibile

difficilmente sara' possibile ottimizzare i check di validita'

- scrivere la documentazione