

Modificare il linguaggio visto a lezione:

- estendere la definizione di variabile in modo che possano essere definite variabili con lettere e numeri (devono cominciare con una lettera).
- Aggiungete al linguaggio visto a lezione i puntatori.

La dichiarazione di puntatore è a se stante rispetto alle variabili (si veda l'esempio di seguito) ed è identificata dal nome della variabile racchiusa tra < e > (*dereferenziazione*); tale annotazione viene usata anche per accedere al valore della zona di memoria puntata da tale puntatore. Se ad un puntatore non è stato assegnato nessun valore e si tenta di dereferenziare tale puntatore (es. Print <p>) allora dovrà essere segnalato un errore all'utente e dovrà terminare l'esecuzione.

Non è prevista l'assegnazione tra puntatori (es. x = y ove x e y sono puntatori).

L'accesso alla zona di memoria di un'altra variabile avviene tramite @ (si veda l'esempio di seguito).

Esempio

```
x = 0;
aaa = 4;
bbb = 7;
<p> ;

if(x){
    print x;
}
p=@x;
x = 1;
if (<p>) {
    print x;
}

print aaa + <p>;
```

Risultato atteso

1
5

NOTA:

- l'esecuzione del programma deve avvenire nel seguente modo ./esercizio fileTest e non ./esercizio < fileTest
- L'esempio di guida proposto in precedenza è a titolo esemplificativo. Ottenere il risultato desiderato non implica l'esito positivo della prova di sbarramento, ma solo il fatto che il lavoro si sta svolgendo nella giusta direzione.

Modalità di consegna dell'elaborato

Consegna tramite email con mittente, destinatari e oggetto come segue:

- **MITTENTE:** vostro indirizzo di posta elettronica del dominio studenti.unitn.it o unitn.it
- **DESTINATARIO:** lorenzom<dot>gramola<at>gmail<dot>com
- **CC:** paola<dot>quaglia<at>unitn<dot>it
- **OGGETTO:** LFC-LAB-PRJ**id**, dove **id** è il vostro numero di matricola
es: se la vostra matricola è 123456 allora l'oggetto è LFC-LAB-PRJ123456

Inoltre

- dovrete allegare alla mail il progetto zippato con nome "prj**id**"(stesso esempio di cui sopra: prj123456). Il formato del progetto zippato deve esser **.zip** (non tar, rar, 7zip etc etc)
- una volta scompattato il file compresso dovrà creare la sua cartella, il cui nome sarà il vostro numero di matricola.
- il file zippato non dovrà contenere nessun eseguibile e nessun file oggetto (nessuno, compresi gli eventuali permessi di esecuzione assegnati ai file)
- **ogni file** dovrà contenere come prima riga il vostro numero di matricola
- prendete come base di partenza l'esercizio sull'interprete; il file zippato dovrà contenere solamente i file necessari alla build (lexer, parser, eventuali header e file di implementazione, in nessun caso è ammesso il logger che è a vostro unico utilizzo).
- oltre ai file necessari alla build dovrà essere fornito un Makefile (**cc = GCC**)con le operazioni di build del progetto e di clean dei file creati (il file eseguibile creato dovrà essere chiamato "interpreter"), con opzione **-std=c99** per la compilazione.

Il rispetto dei requisiti suddetti è considerato condizione necessaria alla validazione del progetto.

Controllate puntualmente ciascun requisito, non sono ammesse consegne multiple per lo stesso appello (il che implica l'invio di una sola email).