

Modificare il linguaggio visto a lezione in modo da separare la dichiarazione di variabili dalla loro assegnazione. Le variabili saranno dichiarate con il costrutto

```
var x;  
il quale assegnerà un valore 0 (default) alla variabile.  
L'assegnazione avviene come di consueto  
x = 89;
```

Se si tenta di assegnare un valore ad una variabile che non esiste il programma dovrà dare un messaggio di errore.

Le variabili dovranno essere dunque dichiarate in un blocco di programma ben definito, secondo la seguente sintassi

```
/**  
blocco dichiarazioni  
**/  
  
main {  
  /*  
    programma  
  */  
}
```

Esempio:

```
var x;  
var y;  
  
main{  
  x = 2;  
  y = 1;  
  print x;  
  print y;  
  z = 89; ---> errore  
}
```

Risultato atteso:

2

1

Messaggio di errore

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).