

Premessa: qualsiasi consegna che non rispetti le regole di seguito indicate verrà considerata nulla al pari delle consegne al di fuori della deadline.

Il recapito di un compito invalido è considerato un filtro sufficiente per la NON ammissione all'esame scritto.

Altri requisiti necessari alla consegna sono i seguenti:

- dovrete inviare il compito dalla vostra casella di posta elettronica all'interno del dominio studenti.unitn.it o unitn.it
- in cc alla mail deve essere specificato l'indirizzo della Professoressa Paola Quaglia
- l'oggetto della mail deve essere LFC-LAB-PRJ id, dove id corrisponde al vostro numero di matricola es. matricola 123456 subject: LFC-LAB-PRJ 123456
- dovrete allegare alla mail il progetto zippato con nome "prj id"(prj123456) 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
- ogni file dovrà contenere come prima riga il vostro numero di matricola
- prendete come base di partenza l'esercizio 5.2; 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 con le operazioni di build del progetto e di clean dei file creati (il file eseguibile creato dovrà essere chiamato "interpreter")

Qualsiasi progetto non rispetti le seguenti condizioni sarà considerato invalido.

Il compito deve essere consegnato una e una sola volta, qualsiasi consegna extra implica l'annullamento del compito, il quale sarà quindi considerato invalido.

NOTE di servizio come requisito fondamentale:

- progetto zippato vuole dire un file in formato zip nominato come richiesto. eg: prj123456.zip, non una cartella compressa in qualsiasi altro formato (che sia tar, rar, 7zip etc etc)
- file eseguibile è considerato anche un file con i permessi di esecuzione (quindi +x), valutare bene che modalità di lettura/scrittura/esecuzione sono associate ai file quando vengono consegnati
- aggiungere l'opzione -std=c99 per la compilazione al vostro Makefile
- il makefile deve contenere il vostro numero di matricola
- non consegnate i file creati da lex e yacc vedete il punto 3: **solamente i file necessari alla build**

Aggiungere al linguaggio visto durante le lezioni di laboratorio lo statement:

repeat stmt until expr

Con la semantica di valutare stmt finche expr ha un valore booleano TRUE.

Un esempio di input, per aiutarvi a svolgere l'esercizio è il seguente:

```
main
{
    int i;
    i = 0;
    repeat{
        print (i);
        i = i+1;
    }until(i<5);
}
```

Tale esempio stamperà i numeri da 0 a 4.

Prendete di riferimento la cartella input per controllare che tutti i precedenti file di input mantengano la loro coerenza rispetto alle modifiche richieste.

NOTA BENE:

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.

Buon lavoro.