

ESAME DI PROGRAMMAZIONE E AMMINISTRAZIONE DI SISTEMA

L'esame deve essere svolto singolarmente e deve essere realizzato unicamente con gli strumenti utilizzati nel corso. Dato che i progetti verranno testati con essi, ogni altro strumento potrebbe far fallire, ad esempio, la compilazione e quindi l'esame. In caso di esito negativo dell'esame, lo studente dovrà presentarsi ad un successivo appello d'esame con il progetto previsto per quella sessione.

Controllate spesso il sito del corso per eventuali aggiornamenti!

Questo documento contiene DUE progetti (leggere le note evidenziate):

Progetto C++

- Creazione di un programma a riga di comando con g++, make e doxygen
- Questo progetto deve essere svolto da tutti gli studenti.

Progetto Qt

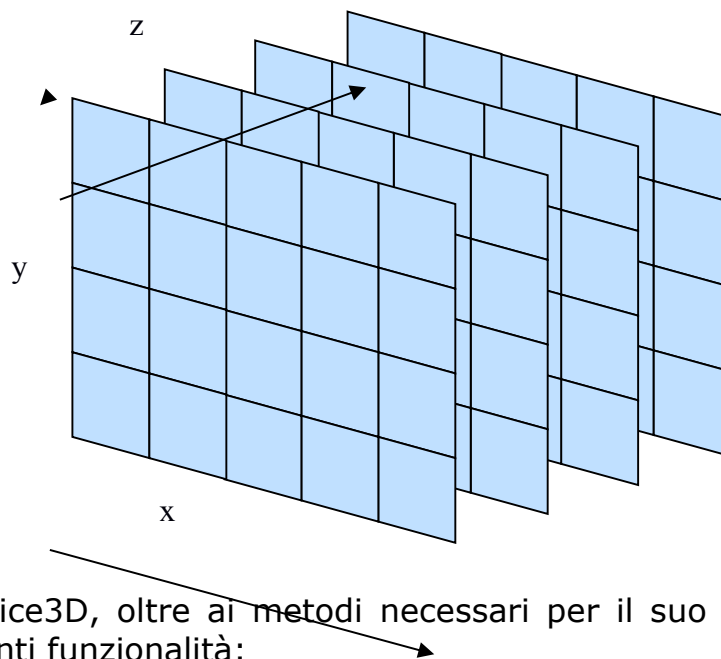
- Creazione di un programma visuale con le librerie Qt
- Questo progetto deve essere svolto solo dagli studenti dell'insegnamento di "Programmazione e Amministrazione di Sistema" iscritti a partire dall'AA 17/18.
- **Gli studenti di Programmazione e Amministrazione di Sistema degli anni precedenti al 17/18 devono svolgere invece un progetto .NET (CONTATTARE IL DOCENTE).**

Progetto C++ del 18/06/2018

**Data ultima di consegna: entro le 23.59 del
10/06/2018**

QUESTO PROGETTO E' VALEVOLE COME PROVA PARZIALE DEGLI INSEGNAMENTI DI Programmazione ad Oggetti C++ (6CFU), Programmazione C++ (4 CFU), Programmazione e Amministrazione di Sistema (8 CFU)

Il progetto richiede la progettazione e realizzazione di una classe generica **Matrice3D** che implementa una matrice a 3 dimensioni di celle. Ogni cella può essere acceduta dando le coordinate del piano (z), riga (y) e colonna (x).



La classe **Matrice3D**, oltre ai metodi necessari per il suo corretto uso, dovrà avere le seguenti funzionalità:

- Deve essere possibile convertire una **Matrice3D** definita su un tipo **U** a una **Matrice3D** definita su un tipo **T**. Ovviamente la conversione sarà possibile se un elemento di tipo **U** è convertibile/castabile a un elemento di tipo **T**.
- Dovrà essere possibile richiedere e scrivere il valore di una cella alla posizione (z,y,x) attraverso **operator()**. Es: $G(1,2,3) = G(2,2,3)$.
- Dovranno essere implementati gli opportuni iteratori di lettura e scrittura.

- Deve essere implementato un metodo, `slice` che, dato il valore del piano `z`, ritorna la **Matrice2D** contenente tutte le celle del piano richiesto. Questa struttura dati deve essere implementata e possedere tutti i metodi necessari per il suo corretto funzionamento, compresi gli iteratori.

Dovrà inoltre essere implementata una funzione generica `trasforma` che, data una Matrice3D **A** (su tipi `T`) e un funtore **F**, ritorna una nuova Matrice3D **B** (su tipi `Q`) i cui elementi sono ottenuti applicando il funtore agli elementi di **A**:

$$B(i,j,k) = F(A(i,j,k))$$

Testate la classe e la funzione globale sia su tipi primitivi che su tipi custom e con diversi predicati `F`.

Nota 1: Se non indicato diversamente, nella progettazione della classe, è vietato l'uso di librerie esterne e strutture dati container della `std library` come `std::vector`, `std::list` e simili. E' consentito il loro uso nel codice di test nel `main`.

Nota 2: Non potete utilizzare i costrutti C++11 e oltre.

Nota 3: Nella classe, è consentito l'uso della gerarchia di eccezioni standard, delle asserzioni e della gerarchia degli stream.

Nota 4: Per vostra sicurezza, tutti i metodi dell'interfaccia pubblica che implementate devono essere esplicitamente testati nel `main`.

Nota 5: Non dimenticate di usare Valgrind per testare problemi di memoria

Nota 6: Evitate di usare "test" come nome dell'eseguibile. Potrebbe dare dei problemi sotto `msys`.

Alcune note sulla valutazione del Progetto C++

- Se in seguito a dei test effettuati dai docenti in fase di valutazione (es. chiamate a funzioni non testate da voi), il codice non compila, l'esame NON è superato.
- Implementazione di codice non richiesto non dà punti aggiuntivi ma se non corretto penalizza il voto finale.
- Gli errori riguardanti la gestione della memoria sono considerati GRAVI.
- La valutazione del progetto non dipende dalla quantità del codice scritto.
- NON usate funzionalità C di basso livello come memcpy, printf, FILE ecc... Se c'e' una alternativa C++ DOVETE usare quella.
- NON chiedete ai docenti se una VOSTRA scelta implementativa va bene o meno. Fà parte della valutazione del progetto.
- PRIMA DI SCRIVERE CODICE LEGGETE ACCURATAMENTE TUTTO IL TESTO DEL PROGETTO.

Progetto Qt del 18/06/2018

**Data ultima di consegna: entro le 23.59 del
10/06/2018**

GLI STUDENTI ISCRITTI A PROGRAMMAZIONE E AMMINISTRARZIONE DI SISTEMA A PARTIRE DALL'AA 17/18 DEVONO SVOLGERE ANCHE QUESTO PROGETTO.

Il progetto richiede la creazione di un editor di testo minimale. Le funzionalità richieste sono le seguenti:

- Creazione di un nuovo file vuoto *Nuovo*;
- Apertura e visualizzazione di un file di testo con estensione “.rtf”, funzione *Apri*;
- Possibilità di modificare il testo in corrispondenza del cursore (cancellazione, inserimento);
- Modifica della dimensione del font (12, 16, 18) e dello stile (grassetto, corsivo e sottolineato) di parte del testo;
- Salvataggio del file, *Salva* per sovrascrivere la versione precedente del file o *Salva con nome...* per specificare il nome di un nuovo file.

Nota 1: Utilizzate la versione 5.6 della libreria Qt.

Alcune note sulla valutazione del Progetto Qt

- Se in seguito a dei test effettuati dai docenti in fase di valutazione (es. chiamate a funzioni non testate da voi), il codice non compila, l'esame NON è superato.
- Implementazione di codice non richiesto non dà punti aggiuntivi ma se non corretto penalizza il voto finale.
- NON verrà valutata l'efficienza dell'applicativo sviluppato.
- Gli errori riguardanti la gestione della memoria sono considerati GRAVI.
- La valutazione del progetto non dipende dalla quantità del codice scritto.
- NON chiedete ai docenti se una VOSTRA scelta implementativa o la configurazione dell'interfaccia grafica va bene o meno. Fà parte della valutazione del progetto.
- NON chiedete ai docenti come installare QtCreator e le librerie Qt
- **PRIMA DI SCRIVERE CODICE LEGGETE ACCURATAMENTE TUTTO IL TESTO DEL PROGETTO.**

Consegna

La consegna del/dei progetti è costituita da un archivio .tar.gz avente come nome la matricola dello studente. L'archivio deve contenere una e solo una cartella con lo stesso nome dell'archivio (senza estensione .tar.gz). Nella root della cartella devono essere presenti:

1. Un **makefile** (per poter compilare il progetto DA RIGA DI COMANDO) che deve compilare tutto il progetto chiamato "Makefile" (attenzione alle maiuscole). Se la compilazione fallisce, il progetto non viene considerato.
2. Tutti i **sorgenti** (commentati come avete visto ad esercitazione) del progetto e organizzati a vostro piacimento.
3. Il file di **configurazione di Doxygen** per la generazione della documentazione chiamato "Doxyfile" modificato per generare documentazione HTML. **GMail ha bloccato l'invio di file con estensione .js quindi non è più possibile allegare la documentazione in formato html.**
4. **Relazione in PDF** con descrizione del progetto contenente informazioni relative al design e/o analisi del progetto. La relazione serve per capire il perchè delle vostre scelte nell'implementazione o di design. Nella relazione mettere anche Nome, Cognome, Matricola ed E-Mail.
5. **Chi deve consegnare anche il "Progetto Qt", metta tutti i file sorgenti corrispondenti in una sotto-cartella "Qt".**
6. L'archivio NON deve contenere file di codice oggetto, eseguibili etc..

L'eseguibile che verrà prodotto non deve richiedere alcun intervento esterno (es. input da tastiera).

Per creare l'archivio è sufficiente lanciare il comando di msys:

- `tar -cvzf 123456.tar.gz 123456`

dove "123456" è la directory che contiene tutti i file da consegnare.

Ad esempio una struttura dell'archivio può essere questa:

```
123456
|--main.cpp
|--project.h
|--Doxyfile
|--...
|--Qt (SOLO PER PROGETTO Qt)
|  |--*.pro
|  |--MainWindow.cpp
|  |--Main.cpp
|  |--MainWindow.ui
```

|--...

Indirizzo Mail: **corsocpp.ciocca@gmail.com**

Mettere come tag all'oggetto della mail:

[c++-consegna] Per consegnare ufficialmente il progetto
Potete fare più consegne e solo l'ultima verrà ritenuta valida.

[c++-test] Per testare il meccanismo di consegna
Verrà eseguita una compilazione del **Progetto C++** (differita e con cadenza oraria 0.00, 1.00, 2.00,...) e restituita una mail con l'esito. Potete fare tutti i test che volete. E' vivamente consigliato effettuare almeno una prova di compilazione (i progetti che non compilano, non vengono considerati). Il Progetto Qt non verrà compilato.

NOTA: questa mail deve essere usata esclusivamente per la consegna dei progetti. Per ogni altra questione usare le mail dei docenti.