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

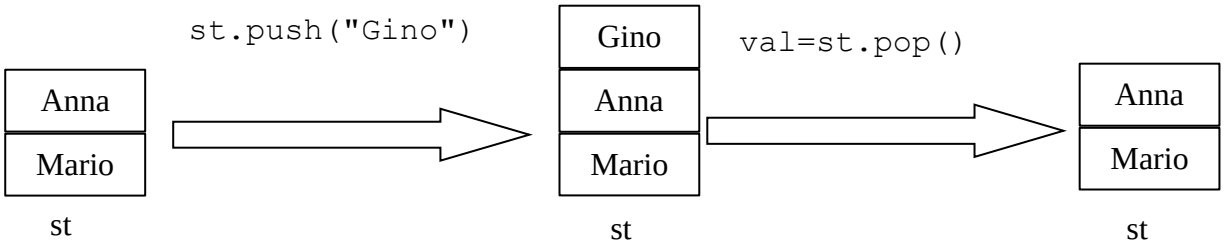
- Creazione di un programma visuale con le librerie Qt
- Questo progetto deve essere svolto solo dagli studenti dell'insegnamento di "Programmazione e Amministrazione di Sistema" iscritti a partire dall'AA 17/18.
- **Gli studenti di Programmazione e Amministrazione di Sistema degli anni precedenti al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

Progetto C++ del 21/09/2018

Data ultima di consegna: entro le 23.59 del 13/09/2018

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 che implementa uno **stack** di elementi di tipo **T**. Uno stack è una struttura dati dove gli elementi sono organizzati in una pila. Un elemento che viene inserito nello stack con il metodo **`push`** è posto "sopra" gli altri già nello stack. E' possibile prelevare dallo stack, mediante il metodo **`pop`**, l'elemento "in cima" alla pila. Una volta che un elemento è prelevato, deve essere rimosso dallo stack. La seguente figura mostra le due operazioni. Lo stack non ha una dimensione fissa.



Per questo progetto NON potete usare una struttura a lista.

A parte i metodi essenziali per la classe (tra cui conoscere il numero di elementi nello stack), devono essere implementate le seguenti funzionalità:

1. Un metodo per svuotare lo stack.

2. Un costruttore che prende una coppia di iteratori: uno che punta all'inizio di una sequenza di elementi e uno che punta alla fine della sequenza. Questo costruttore è usato per riempire lo stack.

3. Un metodo pubblico che ha la stessa funzionalità del costruttore al punto 2. Se lo stack contiene già dei dati, questi vengono rimossi.

4. Un iteratore di sola lettura. L'iteratore deve ritornare gli elementi a partire da quello in cima alla pila fino a quello in fondo alla pila.

5. Un metodo pubblico **`removeif`** che, dato un predicato generico **P** su un elemento di tipo **T** (liberamente definibile dall'utente), elimina dallo stack tutti gli elementi che soddisfano il predicato **P**.

Utilizzare dove opportuno la gestione delle eccezioni. Gestite con una logica opportuna i casi limite/di errore.

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++

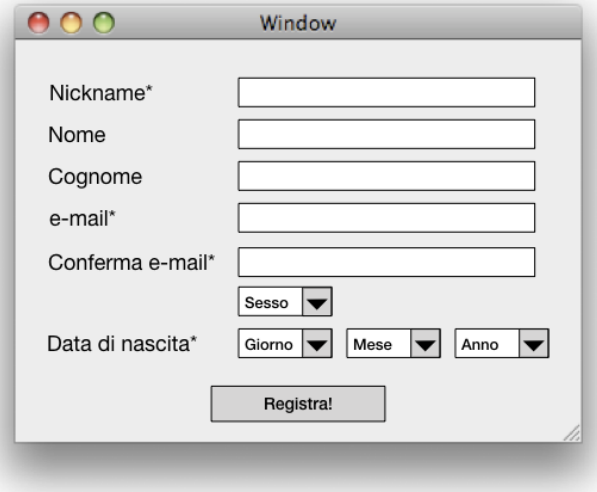
- Se in seguito a dei test effettuati dai docenti in fase di valutazione (es. chiamate a funzioni non testate da voi), il codice non compila, l'esame NON è superato.
- Implementazione di codice non richiesto non dà punti aggiuntivi ma se non corretto penalizza il voto finale.
- Gli errori riguardanti 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 21/09/2018

Data ultima di consegna: entro le 23.59 del 13/09/2018

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

Il progetto richiede la creazione di un form di registrazione simile a quello presentato nel mockup sottostante:



I campi recanti un asterisco (*) vanno riempiti obbligatoriamente affinché il pulsante "Registra!" sia abilitato e che quindi la registrazione possa essere completata. Una delle proprietà più importanti dell'interfaccia riguarda l'inserimento della data di nascita: poiché è possibile inserire indipendentemente giorno, mese e anno è necessario verificare che il giorno selezionato sia coerente con i giorni del mese e con l'anno (bisestile?).

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
```

```
|--dataset
   |--panda
   |--saxophone
   |--…
|--…
```

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.