

ESAME DI PROGRAMMAZIONE C++ (8 CFU)

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 anche 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 CONTATTARE IL DOCENTE.**

Progetto C++ del 17/02/2021

**Data ultima di consegna: entro le 23.59 del
07/02/2021**

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), "Programmazione C++" (8CFU)

Il progetto richiede la progettazione e realizzazione di una classe che implementa un **MultiSet** di elementi generici **T**. Un MultiSet è come un insieme di dati che può contenere duplicati: es. $S=\{1,4,6,4,9,4,7,10,12\}$. Implementare il MultiSet in modo tale da minimizzare l'uso della memoria cioè non dovete memorizzare i duplicati di un elemento. Facendo riferimento all'esempio precedente, il '4' va memorizzato una sola volta sapendo però che ci sono tre occorrenze di '4' in S.

A parte i metodi essenziali per la classe (tra cui conoscere il numero totale di elementi, aggiunta/rimozione elementi, conteggio occorrenze di un elemento, ecc...), devono essere implementate le seguenti funzionalità:

1. la costruzione di un **MultiSet** anche a partire da una sequenza di dati generici **Q** identificata da una coppia di iteratori generici. Questo costruttore prende in input: l'iteratore di inizio sequenza, l'iteratore di fine sequenza. Lasciate al compilatore la gestione della conversione di dati tra **Q** e **T**.
2. Un iteratore di sola lettura (scegliere la categoria). L'iteratore deve ritornare tutti gli elementi del MultiSet. Cioè, i duplicati vanno ritornati in numero corretto. Nel caso d'esempio di S potrebbe essere una sequenza di valori tipo: 1 4 4 4 6 9 7 10 12
3. Implementare l'operatore di confronto **operator==** tra due MultiSet che ritorna true sse i due MultiSet (dello stesso tipo) contengono gli stessi elementi con lo stesso numero di occorrenze dei duplicati.
4. Implementate un metodo **contains** che, dato un elemento di tipo T, ritorna true se l'elemento esiste nel MultiSet.
5. Implementare la funzione globale **operator<<** per inviare su `std::ostream` il contenuto del MultiSet nella forma: `{<X1, occorrenzeX1>, <X2, occorrenzeX2>, ..., <XN, occorrenzeXN>}`

Tenete anche conto che:

- L'ordine degli elementi nel MultiSet non è rilevante.
- La rimozione di un elemento X avviene quando il numero di occorrenze di tale elemento diventa zero.
- Gli elementi del MultiSet sono immutabili. Una volta inseriti, non cambiano valore.

Utilizzare dove opportuno la gestione delle eccezioni.

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: A parte `nullptr`, `std::swap` e altre istruzioni C++11 viste a lezione, non potete utilizzare altri costrutti C++11 e oltre.

Nota 3: Nella classe, è consentito l'uso della gerarchia di eccezioni standard, delle **asserzioni**, la gerarchia degli stream e la funzione `std::swap`.

Nota 4: Per vostra sicurezza, **tutti i metodi dell'interfaccia pubblica che implementate devono essere esplicitamente testati nel main anche su tipi custom.**

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'è una alternativa C++ DOVETE usare quella.
- NON chiedete ai docenti se una VOSTRA scelta implementativa va bene o meno. Fa parte della valutazione del progetto.
- PRIMA DI SCRIVERE CODICE LEGGETE ACCURATAMENTE TUTTO IL TESTO DEL PROGETTO.

Progetto Qt del 17/02/2021

Data ultima di consegna: entro le 23.59 del 07/02/2021

QUESTO PROGETTO E' VALEVOLE COME PROVA PARZIALE DELL'INSEGNAMENTO DI "PROGRAMMAZIONE C++" (8CFU) GLI STUDENTI ISCRITTI A PROGRAMMAZIONE E AMMINISTRAZIONE DI SISTEMA A PARTIRE DALL'AA 17/18 DEVONO SVOLGERE ANCHE QUESTO PROGETTO.

Il progetto richiede la creazione di un form di iscrizione e login. Segue un mockup dell'applicazione:

The mockup shows a web application window with a blue header and a white body. The header contains two input fields: "E-mail o telefono" and "Password", followed by a blue "Accedi" button. Below the header, there is a link "Non ricordi più come accedere all'account?". The main form area contains several input fields: "Nome", "Cognome", "Numero di cellulare o e-mail", and "Nuova password". Below these fields are three dropdown menus for date selection (day, month, year) and two radio buttons for gender selection ("Donna" and "Uomo"). At the bottom of the form is a large green "Iscriviti" button.

Le funzionalità attese sono le seguenti:

- **Iscrizione con obbligo di inserire: nome, cognome, numero di cellulare o e-mail, password, data di nascita, e genere.** Prima di confermare l'iscrizione deve essere verificato che l'utente **non sia già registrato** (e-mail e/o numero di telefono non presenti) e **sia maggiorenne**. L'iscrizione viene confermata con l'apertura di una dialog che dia il benvenuto all'utente registrato;
- **Login** con verifica che l'utente e la password corrispondano. In caso di corrispondenza viene aperta una **dialog che conferma l'accesso riuscito**;
- **Recupero password** in cui viene aperta una dialog di conferma "e-mail con procedura di ripristino inviata" solo se si è verificata l'esistenza dell'utente (e-mail e/o password);
- **Login utente admin** (email: admin@pas.com, password: admin) che possa:
 - Accedere ad un **elenco delle persone iscritte**;
 - Visualizzare **due grafici a torta**: nel primo mostrare la percentuale di uomini e donne iscritte al portale, nel secondo la percentuale di persone rispetto all'età (dividendo le età 5 in gruppi, ad esempio 18-26, 27-35, 36-44, 45-53, e 54+)

Nota 1: Utilizzate la versione 5.12 della libreria Qt.

Nota 2: Sui dati immessi in fase di iscrizione effettuare un controllo di coerenza di formato: email <testo>@<dominio>.it, 31/02/1988, ecc.

Nota 3: I dati degli utenti devono essere salvati in un file (*.csv) per poter garantire la persistenza tra sessioni diverse e altresì consentire il controllo di quali utenti siano già iscritti.

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. Fa 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.