

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.
-

Progetto C++ del 17/09/2024

**Data ultima di consegna: entro le 23.59 del
09/09/2024**

Il progetto richiede la progettazione e realizzazione di una classe che implementa una matrice sparsa (**SparseMatrix**) di elementi generici **T**. Una matrice sparsa è una matrice nella quale solo gli elementi esplicitamente inseriti sono fisicamente memorizzati. Dal punto di vista logico la matrice sparsa si deve comportare come una normale matrice: alla richiesta dell'elemento in posizione (i,j) deve essere ritornato l'elemento inserito oppure un valore di default (D). Questo valore di default (di tipo **T**) è liberamente definito dall'utente e deve essere passato in fase di costruzione della matrice.

A parte i metodi essenziali per la classe (tra cui conoscere il numero di righe/colonne, numero degli elementi inseriti, il valore dell'elemento D, ecc...), devono essere implementate le seguenti funzionalità:

1. La matrice deve essere realizzata in modo tale che siano esplicitamente memorizzati solo gli elementi inseriti dall'utente (utilizzo di memoria minimale). In pratica se la dimensione logica della matrice è NxM, non bisogna memorizzare tutti gli NxM elementi ma solo quelli effettivamente inseriti dall'utente.
2. L'inserimento di un valore alle coordinate (i,j) deve avvenire con un metodo `set`. Se alle coordinate (i,j) esiste già un valore, questo viene sovrascritto a meno che il valore inserito non sia D. In questo ultimo caso il valore in posizione (i,j) è RIMOSSO dalla struttura dati. Se l'utente inserisce esplicitamente il valore di default D in una posizione (i,j) vuota, non succede nulla. In pratica la struttura contiene esplicitamente solo valori diversi da D.
3. La lettura di un valore alle coordinate (i,j) deve avvenire tramite `operator()`. Se alle coordinate (i,j) non c'è un valore inserito dall'utente deve essere ritornato il valore di default D.
4. La classe deve includere il supporto agli iteratori (in sola lettura) di tipo forward per l'accesso ai solli elementi inseriti nella matrice diversi da D. L'iteratore deve ritornare un oggetto `element` che contiene le coordinate (i,j) e il valore inserito in quella posizione. I dati in `element` non devono essere modificabili una volta ritornati dall'iteratore. Non è importante l'ordine con cui sono ritornati gli `element`.

5. Scrivete una funzione globale generica `evaluate` che data una matrice sparsa `M` e un predicato `P`, ritorna quanti valori in `M` (compresi i default) soddisfano `P`. Testate diverse versioni della funzione su diversi tipi di dati custom.

Utilizzare dove opportuno la gestione degli errori tramite asserzioni o 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`, nella classe, non potete utilizzare altri costrutti C++11 e oltre se non indicato diversamente.

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. Evitate di fare dei test interattivi. Fatto solo test automatici.

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. Usate sempre il nome **main.exe**.

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 17/09/2024

Data ultima di consegna: entro le 23.59 del 09/09/2024

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



The mockup shows a Qt application window with a dark blue header and a light blue body. The header contains two input fields: "E-mail o telefono" and "Password", followed by an "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 (23, mar, 1994) 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+)

E' richiesto il controllo della correttezza degli input dell'utente nella compilazione dei campi testuali.

Nota 1: Per lo sviluppo, utilizzare almeno la **versione 5.12.11 della libreria Qt (la stessa installata sulla VM).**

Nota 2: Si renda il contenuto dell'applicazione adattivo rispetto alla dimensione della finestra.

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
- **Nella valutazione viene anche considerata l'usabilità dell'interfaccia sviluppata.**

PRIMA DI SCRIVERE CODICE LEGGETE ACCURATAMENTE TUTTO IL TESTO DEL PROGETTO.

Consegna

La consegna del/dei progetti avviene tramite la piattaforma di eLearning ed 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.
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 o console Linux):

- `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
|  |--...
```