

# 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**

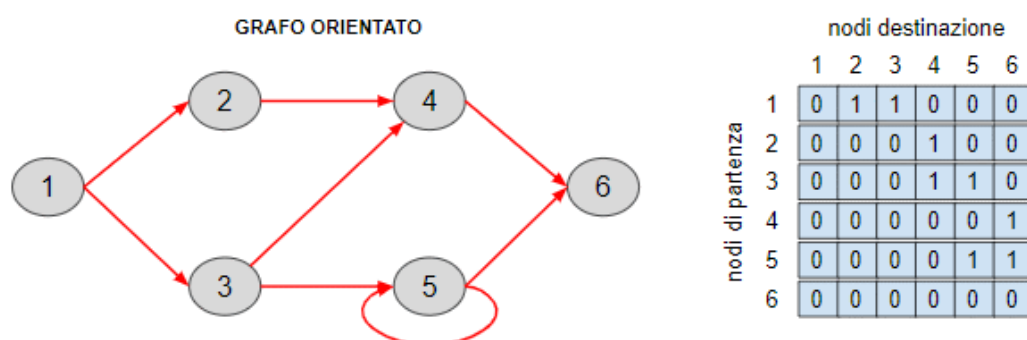
- Creazione di un programma visuale con le librerie Qt
- Questo progetto deve essere svolto anche dagli studenti dell'insegnamento di "Programmazione e Amministrazione di Sistema" iscritti a partire dall'AA 17/18.
- **Gli studenti di Programmazione e Amministrazione di Sistema degli anni precedenti al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

# Progetto C++ del 19/04/2021

**Data ultima di consegna: entro le 23.59 del 10/04/2021**

**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 generica che implementa un **grafo orientato**. Un grafo è costituito da un insieme di nodi e archi. I nodi sono rappresentati da un generico identificativo (es. un numero, una stringa, un oggetto, ecc...). Gli archi mettono in relazione due nodi creando un collegamento tra loro. Un esempio di grafo i cui nodi sono identificati da numeri, è mostrato in figura:



Requisito essenziale del progetto è che il grafo deve essere implementato mediante matrici di adiacenza come in figura. Non possono essere usate liste.

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

1. La classe deve includere il supporto al solo `const_iterator` di tipo `forward` sui nodi. L'iteratore itera sull'insieme degli identificativi dei nodi contenuti nel grafo e l'ordine con cui vengono ritornati non è rilevante.
2. Deve essere possibile aggiungere e rimuovere nodi e archi. Gestire i casi in cui i nodi o gli archi già esistano, si tenti di aggiungere un arco su nodi non esistenti, o si tenti di rimuovere un nodo o un arco non esistente.
3. Deve essere possibile interrogare il grafo per sapere se esiste un nodo tramite un metodo `exists` e se una coppia di nodi è connessa da un arco tramite un metodo `hasEdge`. Nell'esempio in figura, `exists('9')`=false, `hasEdge('1','2')`=true, `hasEdge('4','3')`=false.

Possono essere trascurate considerazioni di efficienza di accesso ai dati e di occupazione di memoria.

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`, 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 19/04/2021

**Data ultima di consegna: entro le 23.59 del  
10/04/2021**

**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 sistema per la risoluzione automatica di un Sudoku. Il Sudoku è costituito da una griglia di numeri 9 x 9, e l'intera griglia è anche divisa in caselle 3 x 3.

9		6		7		4		3
			4			2		
	7			2	3		1	
5						1		
	4		2		8		6	
		3						5
	3		7				5	
		7			5			
4		5		1		7		8

Le regole per risolvere il Sudoku sono:

1. Usare cifre da 1 a 9.
2. Una cifra non può essere ripetuta in una riga, una colonna o in una casella 3 x 3.

Utilizzando un algoritmo di backtracking, risolvere il problema del Sudoku:

- Ricorsivamente cerca quali celle della griglia sono vuote;
- Quando trova una cella vuota, la riempie con una cifra nel range consentito (punto 1 delle regole) e controlla se è valida o meno (se soddisfa il punto 2 delle regole);
- Se non è valida, verifica la presenza di altri numeri.
- Se tutte le cifre nel range consentito sono state verificate e non è stata trovata alcuna cifra valida da posizionare, si torna all'opzione precedente.

Devono essere implementate quindi le seguenti funzionalità:

1. Visualizzare una griglia 9 x 9 vuota.
2. Consentire all'utente di modificare il contenuto delle celle per poter inizializzare le cifre nella griglia del Sudoku.
3. Avere un pulsante "Risolvi" per avviare l'algoritmo di risoluzione.

4. Visualizzare la griglia del Sudoku risolta o un messaggio d'errore nel caso in cui non esiste soluzione (il contenuto della griglia a questo punto non deve essere modificabile).
5. Consentire all'utente di ripercorrere *step-by-step* la risoluzione del Sudoku.
6. Avere un pulsante per resettare la griglia del Sudoku e procedere con un nuovo inserimento.

### **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.
- Il contenuto dell'interfaccia deve essere sempre totalmente visibile, il fatto che non tutti i componenti dell'interfaccia siano visualizzati ad esempio quando la finestra viene rimpicciolita è motivo di penalizzazione.
- 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.