

# Informatica III / programmazione avanzata

Durata 2 h

08/04/22

Chiama i tuoi progetti con COGNOME\_NOME\_ESX con X il numero dell'esercizio. Per la consegna fai uno zip per progetto o uno zip unico come preferisci e caricalo su ilias sotto Consegnà. **Usa carica unico file, NON come file zip.**

## 1. Funzione in C

Scrivi una funzione in C che dato in ingresso due stringhe calcola la distanza tra di loro in questo modo.

Somma la differenza (in valore assoluto) tra i caratteri alla stessa posizione – se a quella posizione non c'è il carattere in una delle due stringhe, la distanza è 100. Ignora maiuscolo/minuscolo.

Ad esempio  $distanza("abc", "aa") = 0 + 1 + 100$  (0 per il primo carattere che è identico, 1 per il secondo essendo a e b nelle due stringhe, e 100 per il terzo che c'è solo nella prima stringa).

Scrivi tre versioni:

- Una iterativa non ricorsiva
- Una ricorsiva senza tail call
- Una ricorsiva con tail call.

Scrivi anche un main di esempio in cui chiami le funzioni con almeno le seguenti stringhe:

`distanza("abc", "aa") , distanza("", "AAAAAA"), distanza("abc", "ABC")`

Non usare alcuna variabile globale. Cerca di tenere il più semplice possibile la segnatura delle funzioni ricorsive, ma se non riesci fai una funzione con segnatura semplice che chiami quella ricorsiva.

## 2. Tipi opachi

Definisci il tipo opaco `LinkedIntStack` che rappresenta uno stack dinamico di interi (con i nodi creati quando serve aggiungere elementi per intenderci). L'inserimento e la rimozione dallo stack avvengono sempre in testa.

**Costruttore:** crea un `LinkedIntStack` vuoto.

**toString** che restituisce una stringa (array di caratteri) dello stack in formato `[a1, a2, a3 ...]`.

**push** aggiunge un numero intero allo stack

**pop** toglie l'ultimo numero aggiunto allo stack e lo restituisce. Se lo stack è vuoto scrive errore e non fa nulla.

**empty** restituisce true se e solo se lo stack è vuoto

**cancella** che distrugge il vettore e libera la memoria

Fai un esempio in cui costruisci uno stack, inserisci i numeri 10 e 5, stampi lo stack prima e dopo ogni inserimento, e poi togli un solo numero (sarà 5), lo stampi e poi cancelli lo stack. Fai in modo di non avere **nessun** memory leak.

## 3. initializer list in C++

A cosa serve la initializer list in C++? Fai un piccolo esempio e spiega il tutto tramite commenti ad un esempio di codice. Per cosa si deve usare che non si può fare altrimenti?

## 4. Smart pointers in C++ (solo AA 2122)

Fai un esempio in cui usi gli smart pointers e fai vedere la differenza rispetto all'uso dei raw pointers.

## 5. OrderedStack Java

Definisci una classe OrderedStack che è generica in E (ordinabile) e che ha le seguenti metodi:

- costruttore che costruisce un OrderedStack vuoto.
- push, pop, empty come esercizio 2
- max, min che restituisce il massimo elemento nello stack

Questo OrderedStack ha come base sottostante un array di Object (Object[]). Se quando faccio un push l'array non riesce più a contenere gli elementi, lo "allargo" (come?)

Nel main fai un paio di esempi con interi o stringhe. Se ho una classe Persona che è Comparable<Persona> e Studente che estende Persona, posso creare dei OrderedStack di Studente? Dimostra con il codice.

## 6. Programmazione funzionale (4 punti)

Haskell (solo AA 2122): Definire in Haskell una funzione

quadratimin :: Int → [Int]

che, dato un intero positivo, restituisce la lista di tutti i numeri quadrati perfetti più piccoli dell'input (ignora 1).

Esempio:

> quadratimin 10

[2, 4, 9]

Scala (prima di AA2122):

Scrivi in scala la funzione dell'esercizio 1 con Scala, però che salta le lettere maiuscole in una delle due stringhe.

Definisci diverse versioni (in ordine di difficoltà)

- distance\_for con un semplice ciclo for
- distance\_foreach usando il foreach (e una funzione + var locale)
- distance\_rec: usando la ricorsione
- distance\_tail usando la tail recursion
- distance\_filter usando anche filter
- distance\_map usando anche map (pensa a come poter usare la mappa in modo efficace)
- distance\_reduce usando anche foldLeft o foldRight

Scrivi alcuni esempi con le parole dell'esercizio 1 e chiama le funzioni sopra definite.

Se volessi generalizzare il metodo per filtrare non solo quelle maiuscole ma in generale quelli che soddisfano un certo criterio, come potrei usare le HOF? Ad esempio se volessi misurare la distanza delle vocali come farei con le HOF? Riscrivi un paio di metodi di cui sopra usando le higher order function e la funzione F opportuna. Usa il currying se riesci.