
FONDAMENTI DI PROGRAMMAZIONE A

Tempo a disposizione: 1 ora 40 minuti

Nome Cognome Matricola

Esercizio 1 (5pt). Scrivere una funzione di nome **revert** con tipo di ritorno **void** che, preso come parametro una stringa C-style **str**, inverte l'ordine dei caratteri della stringa.

Esempio:

```
char s[5] = "hello";
revert(s);
cout << s; // stampa "olleh"
```

Non è consentito usare la libreria **cstring**.

Esercizio 2 (10pt). Scrivere una funzione **main** che legge una sequenza di interi **positivi** da tastiera e li memorizza in un **array dinamico** di interi **a** (dimensione massima 1000 elementi). La lettura si interrompe quando l'utente inserisce da tastiera un intero negativo. La funzione **main** deve:

- ▶ stampare a video l'array **a**
- ▶ creare un nuovo **array dinamico** **b** in cui vengono copiati tutti gli elementi dell'array **a** che compaiono solo una volta in **a**
- ▶ stampare a video l'array **b**
- ▶ deallocare opportunamente la memoria dinamica utilizzata

Per esempio, se l'array **a** è [3 6 2 7 6 2 8], l'array **b** sarà [3 7 8];

Esercizio 3 (15pt). Scrivere una funzione di nome **concat** con tipo di ritorno **void** che, presi come suoi parametri due liste concatenate semplici **l1** ed **l2** i cui elementi hanno campo informazione di tipo **char**, concateni la lista **l2** alla lista **l1**. Ad esempio, se **l1** = {'a', 'd'} e **l2** = {'g', 'f', 'l'}, dopo la chiamata alla funzione **concat** passando come parametri **l1** ed **l2**, la lista **l1** diventa {'a', 'd', 'g', 'f', 'l'}. Gestire in modo opportuno i casi in cui **l1** e/o **l2** siano liste vuote. (+2pt se la funzione è ricorsiva)