

## Durata della prova

2 ore e 30 minuti.

## Istruzioni per lo svolgimento e la consegna

**Fase 1 (carta e penna).** Svolgere i due esercizi con carta e penna **commentando adeguatamente le proprie idee**. Al termine della prova, creare un file pdf (denominato `cognome_nome.pdf`) contenente la scansione di tutti i fogli (**al massimo 4 facciate**) su cui sono stati svolti gli esercizi e inviarlo come allegato all'indirizzo di consegna.

**Fase 2 (implementazione).** Implementare le soluzioni dei due esercizi svolti durante la prima fase in due file C separati e denominarli `cognome_nome1.c` e `cognome_nome2.c`. **Commentare adeguatamente le proprie idee e indicare esplicitamente ogni differenza con la soluzione cartacea**. Inviare i due file all'indirizzo di consegna entro le ore 21:00 del ... febbraio 2021.

<b>Programmazione I (Classe 1) – Appello III - ...</b>	
<b>TEMPO PARTE PRATICA:</b> 150 minuti	
<b>NUMERO di fogli per la consegna:</b> massimo 4 facciate A4 (no brutta, no traccia)	
<b>ORA di consegna dei file:</b> 21:00	
<b>Indirizzo di consegna:</b> proglresto02@gmail.com	
<b>Cognome:</b>	<b>Nome:</b>
<b>Matricola:</b>	<b>Firma:</b>

**Attenzione:** Commentare adeguatamente la soluzione proposta sia nella versione cartacea sia nella corrispondente implementazione.

**Esercizio 1** Due parole si dicono legate da uno scarto quando l'una si può ottenere dall'altra togliendo una lettera, come per esempio accade con *porzione* e *pozione*, oppure con *appello* e *cappello*.

Si scriva una funzione `char *scarto (char *s, char *t)` che prenda in input due stringhe *s* e *t* e restituisca:

- il puntatore ad *s* se *s* si ottiene togliendo una lettera da *t*;
- il puntatore a *t* se *t* si ottiene togliendo una lettera da *s*;
- `NULL` altrimenti.

La funzione dovrà essere usata da un programma `scarto` che prenda le due stringhe dalla riga di comando e, nel caso le due parole siano legate da uno scarto, stampi la stringa che si ottiene dall'altra togliendo una lettera; altrimenti non stampi nulla.

Ecco alcuni esempi di esecuzione:

```
> ./scarto porzione pozione
porzione

> ./scarto appello cappello
appello

> ./scarto oscura oscurata

> ./scarto oscura oscura

> ./scarto oscura scura
scura
```

**Attenzione:** Il programma può usare solo le librerie `<stdio.h>` e `<stdlib.h>`.

. ~ .

**Esercizio 2** Scrivere un programma *classifica* che permetta di esaminare la classifica finale di un campionato di calcio della serie A, memorizzata in un file *classifica.txt*.

Il file è organizzato come segue. Ogni linea corrisponde ad una squadra del campionato e presenta nell'ordine: il nome della squadra seguito da una virgola, il numero di punti, il numero di partite giocate, il numero di vittorie, il numero di pareggi, il numero di sconfitte, il numero di gol fatti e il numero di gol subiti. Si noti che il numero di spazi fra gli elementi è indefinito.

Esempio (campionato 1986-1987):

```
Napoli,      42   30  15  12  3  41  21
Juventus,    39   30  14  11  5  42  27
Inter,       38   30  15   8  7  32  17
Verona,      36   30  12  12  6  36  25
Milan,       35   30  13   9  8  31  21
...
```

Il programma può essere usato in tre modi.

1) Passando il nome di una squadra. Esaminando il file, il programma stampa, su un'unica linea il nome della squadra seguito dal suo numero di punti.

Esempio:

```
> ./classifica Avellino
Avellino 30
```

2) Passando due numeri *a* e *b*. Esaminando il file, il programma stampa l'elenco delle squadre con un numero di punti compreso tra *a* e *b* (estremi inclusi); ciascuna squadra sarà seguita dal suo numero di punti.

Esempio:

```
> ./classifica 30 35
Milan 35
Sampdoria 35
Roma 33
Avellino 30
```

3) Passando *-a* oppure *-d* seguito da un numero *n*. Esaminando il file, il programma si comporta come segue.

- Caso *-a*: fra tutte le squadre con almeno *n* punti, il programma stampa quella con il miglior attacco (maggior numero di gol fatti);
- Caso *-d*: fra tutte le squadre con almeno *n* punti, il programma stampa quella con la miglior difesa (minor numero di gol subiti).

Esempi:

```
> ./classifica -a 30
Juventus, gol fatti: 42

./classifica -a 40
Napoli, gol fatti: 41

./classifica -d 39
Napoli, gol subiti: 21

./classifica -d 35
Inter, gol subiti: 17
```

Ogni situazione non specificata nei due punti precedenti (inclusi vari casi di input non validi) può essere ignorata o gestita liberamente.

**Attenzione:** Per la stampa dell'output, ci si attenga alle indicazioni date sopra evitando di introdurre frasi non richieste, come: "la squadra ... ha vinto ..., ecc. **Soluzioni che ignorino tali indicazioni, rischiano di essere penalizzate o di non essere prese in considerazione!**

Esempio con input errato.

Uso: ./classifica <squadra>

Uso: ./classifica <numero> <numero>

Uso: ./classifica -a/-d <numero>

. ~ .