

Fondamenti di Informatica

Andrea Gussoni
andrea1.gussoni at polimi.it

Politecnico di Milano

May 11, 2021

Table of Contents

- 1 Ripasso Esercitazione Precedente

Section 1

Ripasso Esercitazione Precedente

Ripasso

Problemi con esercizi della precedente esercitazione?

Calcolo Esponenziale

- Implementare il calcolo dell'esponenziale e^x tramite l'uso della serie di Maclaurin
- $e^x = f(0)\frac{x^0}{0!} + f'(0)\frac{x^1}{1!} + f''(0)\frac{x^2}{2!} + \dots$
- $e^x = \frac{x^0}{0!} + \frac{x^1}{1!} + \frac{x^2}{2!} + \dots$
- Fare uso estensivo dei sottoprogrammi.

Vettore al Quadrato

- Implementare un sottoprogramma che, preso in input in vettore di interi, rimpiazzhi ogni intero originalmente conenuto con il suo quadrato.

Libreria Stringhe

- Implementare, utilizzando i puntatori, le seguenti funzioni di libreria per la manipolazione di stringhe:
- `strlen(char *stringa);`
- `strcpy(char *str_dst, char *str_src);`
- `strcat(char *str_dst, char *str_src);`
- `strcmp(char *stringa1, char *stringa2);`

Anagrammi

- Implementare, utilizzando i puntatori, un sottoprogramma C che controlla se due stringhe date in input sono anagrammi.

Parole Hertziane

- Due parole a e b sono *Hertziane* se entrambe sono leggibili anche oscillando durante la lettura dei caratteri tra di esse. Ad esempio, le parole *tigre* e *fiera*, possono entrambe essere lette partendo dal primo carattere di ciascuna, e oscillando poi sui caratteri dell'altra.
- Nella situazione descritta sopra, parliamo di parole direttamente *Hertziane*, ma possiamo anche avere parole inversamente *Hertziane*, dove la lettura comincia dalla prima lettera della parola opposta. *Fiera* e *Fresa* sono esempi di parole inversamente *Hertziane*.

Gara di Tuffi

- Nello sport dei tuffi individuali, ogni tuffo viene valutato da cinque giudici diversi.
- Ogni giudice esprime un punteggio individuale da 0 a 10, con la possibilità di esprimere anche mezzi punti. Il punteggio 0 corrisponde a un tuffo completamente sbagliato; il punteggio 10 a un tuffo perfetto.
- Per eliminare eventuali preferenze soggettive, il punteggio più alto e quello più basso vengono scartati in automatico. Ogni punteggio è espresso indipendentemente dal grado di difficoltà del tuffo (un valore numerico che va da 1.3 a 3.6) e il punteggio finale attribuito all'atleta è calcolato come prodotto della somma dei tre punteggi rimanenti per il grado di difficoltà del tuffo .

Gara di Tuffi

- Data la macro NUM_GIUDICI e la definizione di tipo punteggi_tuffo

```
#define NUM_GIUDICI 5
typedef float punteggi_tuffo[NUM_GIUDICI];
```

- scrivere in C la funzione:

```
float calcola_punteggio(punteggi_tuffo voti, floatdiff)
```

che calcola il punteggio finale di un tuffo a partire dai punteggi individuali espressi dai giudici (voti) e dal grado di difficoltà del tuffo (diff).

Gara di Tuffi

- A una gara di tuffi partecipa un certo numero di atleti, indicato con la costante NUM_ATLETI.
- Si considerino le seguenti definizioni:

```
#define NUM_ATLETI 6
#define MAX_STRING 30
typedef struct {
    char nome[MAX_STRING];
    char cognome[MAX_STRING];
    char sesso;
    char nazionalita[MAX_STRING];
    float punteggio_finale;
} atleta;
typedef atleta gara[NUM_ATLETI];
```

Gara di Tuffi

- Il tipo di dato atleta definisce le proprietà di ciascun partecipante alla gara. L'attributo sesso del tipo di dato atleta distingue uomini ('m') e donne ('f'), mentre l'attributo punteggio_finale è da intendersi come il punteggio complessivo ottenuto da un atleta durante tutta la manifestazione. Il tipo di dato gara definisce una array in cui le celle rappresentano gli atleti.
- Scrivere in C la funzione:

```
void stampa_migliori(gara G)
```

che stampa a schermo la migliore donna e il miglior uomo della competizione.