

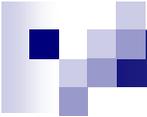
Esercizi per la preparazione all'esame pratico

Alessandra Giordani

agiordani@disi.unitn.it

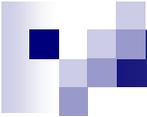
Lunedì 28 maggio 2012

<http://disi.unitn.it/~agiordani/>



Implementare la funzione

- `int anagramma (char *a, char* b);`
 - `a` e `b` sono frasi, cioè sono ammessi i caratteri di punteggiatura (range [A-Za-z ,.!])
 - tuttavia `anagramma` deve considerare solo i caratteri alfabetici e senza ripetizioni
 - ritorna 0 se `a` e `b` non contengono gli stessi caratteri (in ordine anche diverso, con ripetizioni)
 - altrimenti ritorna il numero di caratteri (senza ripetizione) comuni alle due frasi
 - `anagramma` deve essere case insensitive



Esempi

A. Anagramma non valido

□ `a="Ciao"`

□ `b="Cibo"`

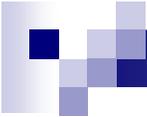
- `anagramma(a, b)` ritorna 0
- Infatti togliendo non sono composte degli stessi caratteri (pur avendo al stessa lunghezza!)

B. Anagramma non valido

□ `a="Sonno profondo"`

□ `b="sono profondo, oh"`

- `anagramma(a, b)` ritorna 0
- Infatti togliendo i caratteri doppi e non alfabetici da entrambe le stringhe la lunghezza di $a < b$



Esempi

C. Anagramma valido

□ a="mi dai tonno?"

□ b="dammi 1 tono"

■ `anagramma(a, b)` ritorna 7

■ Infatti togliendo i caratteri doppi e non alfabetici:
`midaton`

D. Anagramma valido

□ a="Ho sonno profondo"

□ b="sono profondo, oh"

■ `anagramma(a, b)` ritorna 8

■ Infatti togliendo i caratteri doppi e non alfabetici:
`hosnprfd`



Implementare un main per

- Leggere interattivamente `a` e `b`
- Scrivere a video quanto letto
- Richiamare `anagramma`
- Utilizzare il valore ritornato dalla funzione per stampare un messaggio del tipo:
 - *Non sono anagrammi.*
 - *Sono anagrammi di 7 lettere.*
 - *Sono anagrammi di 8 lettere.*
 - *...*



Note

- Non si possono richiamare funzioni di libreria `string.h` come `strlen`, `strstr`, `strcmp`
- Potete usare le funzioni di `ctype.h`:
 - `char toupper (char c)`
 - Restituisce il maiuscolo del carattere `c`
 - `int isalpha (char c)`
 - Restituisce 0 se `c` non è un carattere alfabetico



Valutazione

- Deve compilare (senza errori)
 - Deve accettare due stringhe (spazi!!!)
 - Deve funzionare correttamente con almeno 2 esempi dei punti A,B,C,D
- per essere sufficiente (VOTO C)
- Se funziona con 3 esempi → VOTO B
 - Se funziona con i 4 esempi → VOTO A
- ... in piu se si finisce prima dei 45 minuti o il codice è particolarmente pulito, geniale, ... → VOTO A+

Suggerimenti: indici e cicli

- Quando sono coinvolte 2 stringhe il più delle volte servono 2 indici distinti per scorrerle quindi avrete 2 cicli for annidati

```
for (i=0;str[i]!='\0';i++)  
    for (j=0;str2[j]!='\0';j++)  
        //fai qualcosa con str[i] e str2[j]
```

```
for (i=0;i<n;i++)  
    for (j=0;j<m;j++)  
        //fai qualcosa con str[i] e str2[j]
```

```
for (i=0;str[i]!='\0';i++)  
    for (j=i+1;str[j]!='\0';j++)  
        //fai qualcosa con str[i] e str[j] (j>i)
```

- Anche quando abbiamo solo 1 stringa e dobbiamo considerare coppie di caratteri a posizioni diverse!

Suggerimenti: occhio ai simboli

- Tante volte si sbagliano delle piccolezze!
- Il compilatore non segnala errori di errata digitazione di simboli.. Ad esempio:
 - `=!` Non è un errore ma non è come scrivere `!=`
 - `if (a=b)` non è come `if (a==b)`
 - `for (...;...;...);` ha un significato diverso da `for (...;...;...)` e non è detto che sia sbagliato!
 - `"a"` Non è un errore ma non è come scrivere `'a'`
 - `"a"` in particolare è formato da `'a'` e da `'\0'`



Suggerimenti: getchar()

- La `getchar()` serve per eliminare il carattere `\n` dal buffer di lettura
- La `scanf` infatti legge finchè non trova un terminatore: può essere l'acapo, ma anche uno spazio, nel caso in cui non mettiamo lo spazio nel range dei caratteri accettati
- Può accadere che debba servire un ciclo di `getchar()` per pulire i caratteri non alfabetici presenti nel buffer...



Altri esercizi per l'esame...

- <http://disi.unitn.it/~agiordani/teaching10.htm>
- <http://disi.unitn.it/~agiordani/teaching11.htm>
 - Prove pratiche e scritte con soluzioni allegate
- <http://danielepighin.net/cms/files/esercizi.txt>
 - Prove pratiche senza soluzioni disponibili online

Per ogni problema o dubbio riguardante la compilazione e l'implementazione di tali programmi scrivete a agiordani@disi.unitn.it