



Esercizi per la preparazione all'esame pratico

Alessandra Giordani

agiordani@disi.unitn.it

Lunedì 13 maggio 2013

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



Date Appelli

■ 18 giugno 2013

- Pratica: 9.00-13.00 Aula pc A201 + A202
- Teoria 13.30-15.30 Aula A101

■ 18 luglio 2013

- Pratica: 9.00-13.00 Aula pc A201 + A202
- Teoria 13.30-15.30 Aula A101

■ 6 settembre 2013

- Pratica: 14.00-14.00 Aula pc A201 + A202
- Teoria 10.30-12.30 Aula A106



Nuove Regole per gli Esami

- Teoria e Pratica si possono sostenere in appelli diversi (nell'arco di 12 mesi)
 - Non è necessario passare prima la Pratica
- Chi consegna la teoria a giugno e non supera l'esame teorico non può presentarsi alla teoria nell'appello di luglio, ma all'esame pratico si. E Viceversa.
- Iscrivetevi tramite Esse3
 - Se non ci riuscite mandate una mail
 - Potreste non venir accettati se non siete iscritti



Esempio 1

- Pluto si presenta all'esame di giugno, ma solo alla teoria (perché ad esempio la mattina ha un altro esame)
 - Se consegna e non supera l'esame teorico può riprovarlo a settembre (o appelli dopo)
 - Se consegna e supera l'esame otterrà un punteggio $T1+T2+T3$ di almeno 18, valido per 12 mesi
- In entrambi i casi a luglio potrà presentarsi all'esame pratico



Esempio 2

- Pippo si presenta all'esame di giugno, consegna la pratica (fa correggere il programma) e la supera con voto P.
- Il pomeriggio si presenta alla teoria
 - Se non consegna può riprovare la teoria all'appello di luglio
 - Se consegna e non supera l'esame teorico può riprovarlo a settembre
 - Se consegna e supera l'esame, il voto finale sarà $(P+T1+T2+T3)/4*3$ trentesimi



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 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...



Altra Simulazione d'esame

Implementare la funzione

```
void differenza(char* a, char* b, char *c)
```

che estrae e ritorna la lista di caratteri c dalla stringa a tale che i caratteri di c non appartengono alla stringa b .



Dettagli:

- La funzione accetta solo caratteri alfabetici e non fa distinzione tra maiuscole e minuscole.
- Casi particolari: Se la stringa a e' vuota, anche c sara' la stringa vuota (indipendentemente da b).
- Se la stringa b e' vuota, c sara' identica alla stringa a.



Esempi

- Esempi di stringhe immesse

`"militare", "mare" ⇒ "ilit"`

`"mare", "militare" ⇒ ""`

`"Amore", "mare" ⇒ "o"`

`"marea", "mare" ⇒ ""`

`"al mare", "mare" ⇒ Stampa messaggio di errore`

`"mari", "Orme" ⇒ "ai"`



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