

Ingegneria – Consorzio NETTUNO

Fondamenti di Informatica 1

14-2-09 – Sterbini

Risolvete almeno 3 esercizi a piacere tra i seguenti

Esercizio 1 (8 punti)

Si scriva la funzione C che riceve come argomenti: una stringa **parola** ed un vettore di stringhe **dizionario** e tutti gli altri argomenti che ritenete necessari, e che cerca nel dizionario la parola che più le si avvicina, ovvero la parola che la contiene e differisce per il minimo numero di caratteri aggiunti (nel caso migliore 0 se sono uguali).

La funzione deve tornare (in due variabili passate per riferimento) l'indice della parola trovata e il numero di caratteri differenti (0 se le parole sono uguali). Se il dizionario non contiene parole simili si torni -1 -1.

Consiglio: suddividete il programma in funzioni più semplici.

Esempio: la parola è “mamma” e la parola “mammella” sono simili (la seconda ha 3 caratteri aggiunti)

Esercizio 2 (8 punti)

Si scriva la funzione C che riceve come argomenti: una matrice RxC di interi **M** e tutti gli altri argomenti che ritenete necessari, e che calcola quante volte nella matrice sono presenti 3 interi consecutivi, disposti orizzontalmente, verticalmente o in diagonale.

La funzione deve tornare, in due variabili passate per riferimento, le coordinate X,Y del primo numero della terna più grande trovata.

Consiglio: suddividete il programma in funzioni più semplici.

Esempio di matrice con due terne di valori consecutivi, i valori da tornare sono 1, 1 (coordinate del valore 7 della terna 7,8,9).

1	2	3	5
2	7	9	10
2	7	8	11
2	7	9	9

Esercizio 3 (8 punti)

Si scriva la funzione C che riceve come argomenti: un vettore di stringhe **testo** i cui elementi sono stringhe (frasi formate da parole separate da spazi) e tutti gli altri argomenti che ritenete necessari, e che stampa tutte le frasi che sono palindrome **ignorando gli spazi**.

Consiglio: suddividete il programma in funzioni più semplici.

Esempio: la frase “amo Roma” è palindroma (se ignoro lo spazio la si può leggere uguale da SX e da DX)

Esercizio 4 (8 punti)

Si scriva la funzione C che riceve come argomenti: una matrice RxC di interi **M** e tutti gli altri argomenti che ritenete necessari, e che ordina gli elementi di ciascuna riga della matrice con valori crescenti da sinistra a destra, ed inoltre ordina tra loro le righe in ordine lessicografico crescente (ovvero in modo che una riga che ha il primo elemento minore del primo di un'altra si trovi prima, e se invece i due primi elementi sono uguali, venga prima la riga che ha il secondo elemento minore ... eccetera).

Consiglio: suddividete il programma in funzioni più semplici.

Esempio di matrice disordinata e ordinata:

4	2	1	3
2	10	9	5
8	7	9	2
9	7	10	2

1	2	3	4
2	5	9	10
2	7	8	9
2	7	9	10

Esercizio 5 (6 punti)

Si supponga di dover implementare l'algoritmo di merge-sort.

Si scriva la funzione RICORSIVA **merge**, che riceve come argomenti due vettori ordinati ed un vettore destinazione di dimensioni appropriate (e tutti gli altri argomenti che ritenete necessari), e che fonde i due vettori copiando nel vettore destinazione i valori in modo da produrre un unico vettore ordinato.

NOTA: non usate iterazioni (for, while, do) ma solo ricorsione.

Esercizio 6 (6 punti)

Si supponga di avere a disposizione la funzione **merge** dell'esercizio precedente.

Si scriva la funzione RICORSIVA **merge_sort**, che riceve come argomenti un vettore disordinato ed un vettore appoggio di uguali dimensioni (e tutti gli altri argomenti che ritenete necessari), e che usa la funzione merge per ordinare l'intero vettore seguendo la strategia:

- se il vettore è di un solo elemento è già ordinato
- altrimenti suddivide il vettore in due metà (che sono disordinate)
 - ordina ciascuna metà
 - fonde le due metà (che sono ordinate) con la funzione **merge**