

Prova d'Esame del 06/07/2011
Tempo a Disposizione: 3 ore

Esercizio 1. Si scriva un programma C++, **OPPORTUNAMENTE MODULARIZZATO IN FUNZIONI**, che legga da standard input una frase in cui le parole sono separate da spazi e, dopo aver opportunamente individuato e memorizzato tutte le parole di cui la frase è composta, esegua le seguenti operazioni: 1) controlli se le parole nella frase sono disposte in ordine alfabetico (ossia, ogni parola nella frase deve precedere la successiva secondo l'ordine alfabetico) e stampi a video "Frase Crescente" se la condizione è verificata, "Frase non Crescente" altrimenti; 2) stampi a video la parola più corta e la relativa lunghezza (nel caso di più parole di lunghezza uguale stampi la prima incontrata); 3) calcoli la media delle lunghezze di tutte le parole, troncata all'intero più piccolo, e stampi per ogni parola lo scarto della sua lunghezza dalla media. **Esempio:** se la frase in input fosse "*ai bagnanti il mare mosso non piace*" su standard output si dovrebbe stampare: 1) Frase Crescente, 2) Parola più corta: *ai* con lunghezza: 2 (perché le parole più corte sono *ai* ed *il* entrambe di lunghezza 2, ma il programma deve stampare solo la prima trovata) e 3) Media delle lunghezze: 4, Scarti: *ai* 2, *bagnanti* -4, *il* 2, *mare* 0, *mosso* -1, *non* 1, *piace* -1 (perché, ad esempio, la lunghezza di *bagnanti* è 8, la media è 4 e $4 - 8 = -4$).

Esercizio 2. Si scriva in C++ una funzione RICORSIVA che, ricevuti come parametri (almeno) un array di numeri naturali, la sua dimensione, ed un numero naturale X, restituisca un altro numero naturale N calcolato come segue: si cerchino tutte le occorrenze dell'intero X, e nelle relative posizioni si sostituisca X con 0; sia inoltre il numero di queste occorrenze C; si ripeta il procedimento appena illustrato cercando le occorrenze di C e così via. Il processo si ferma quando capita che il numero di cui ricercare le occorrenze è 0. Il numero naturale N da restituire è dato dalla somma di tutte le occorrenze dei numeri cercati.

ESEMPIO: si consideri l'array $\{2\ 3\ 1\ 3\ 4\ 2\ 4\ 9\ 2\}$, e si supponga che la funzione sia invocata con numero da cercare $X = 4$. Allora i passi che la funzione deve eseguire sono i seguenti:

$2\ 3\ 1\ 3\ 4\ 2\ 4\ 9\ 2 \leftarrow$ il 4 è presente 2 volte; tutte le occorrenze di 4 passeranno a 0; il nuovo numero da cercare è 2.

$2\ 3\ 1\ 3\ 0\ 2\ 0\ 9\ 2 \leftarrow$ il 2 è presente 3 volte; tutte le occorrenze di 2 passeranno a 0; il nuovo numero da cercare è 3.

$0\ 3\ 1\ 3\ 0\ 0\ 0\ 9\ 0 \leftarrow$ il 3 è presente 2 volte; tutte le occorrenze di 3 passeranno a 0; il nuovo numero da cercare è 2.

$0\ 0\ 1\ 0\ 0\ 0\ 0\ 9\ 0 \leftarrow$ il 2 è presente 0 volte; il processo termina restituendo 7 (cioè $2+3+2$).

Esercizio 3.

Si consideri il famoso gioco enigmistico "CRUCIPUZZLE" in cui è data una griglia di lettere all'interno della quale sono celate alcune parole. L'obiettivo del gioco è individuare queste parole. Le parole sono individuabili, a partire da una lettera, in orizzontale e verticale nelle 4 direzioni dei punti cardinali (si noti che nel gioco originale le parole possono essere ricercate anche in diagonale, ma per semplicità qui trascuriamo questa possibilità). Si scriva un programma C++ che supporti un giocatore pigro. Il programma riceve in input la griglia delle parole e la sua dimensione e una parola; se la parola è presente nella griglia il programma stampa in output le coordinate della cella da cui la parola inizia (la posizione della prima lettera) e la direzione verso cui si estende ('N', 'S', 'E', 'O'); altrimenti, stampa la stringa "Parola non presente nella griglia".

ESEMPIO: si consideri la griglia riportata in figura, e la ricerca della parola "zie". Il programma, in tal caso, deve stampare la posizione della lettera 'z' da cui parte la parola, ossia (2,3) e la direzione 'N'.



c	a	n	e
a	t	r	i
s	u	e	z
a	t	t	o

BONUS 1: si modifichi il programma in modo che riceva, invece di una sola parola, un elenco di parole ed esegua la ricerca su descritta per ciascuna delle parole nell'elenco.

BONUS 2: si modifichi ancora il programma (a partire da quanto realizzato nel Bonus 1) in modo da stampare in output la sequenza di tutte le lettere della griglia di partenza non comprese in nessuna delle parole in elenco.

Esempio: Si supponga che l'elenco in input comprenda le parole "casa", "tuta", "cane", "irta", "zie", per semplicità evidenziate in figura. Il programma deve stampare la sequenza di lettere 'e', 't', 'o'.

c	a	n	e
a	t	r	i
s	u	e	z
a	t	t	o