

**Esercizio 1.** Si scriva un programma C++, OPPORTUNAMENTE MODULARIZZATO IN FUNZIONI, che, lette da input tre parole A,B e C, verifichi se queste costituiscono un incastro di parole secondo uno schema prestabilito, anch'esso letto da input. Uno schema è una stringa lunga quanto la parola C e composta solamente da ripetizioni dei caratteri 'a' e 'b'. Si ha un incastro di parole se: 1) la lunghezza di C è uguale alla lunghezza di A più la lunghezza di B; 2) le lettere di cui C è composta provengono o dalla parola A o dalla parola B secondo lo schema (in altre parole: se nella posizione x dello schema è presente il carattere 'a' (risp. 'b') allora nella posizione x della parola C si deve trovare una lettera proveniente dalla parola A (risp. B)); 3) l'ordine in cui le lettere compaiono in A e B è rispettato in C.

Le parole A=**MEDI**, B=**RIONE** e C=**MERIDIONE**, formano un incastro di parole secondo lo schema **aabbaabbb**. Al contrario, le parole A=**CARE**, B=**SOLA** e C=**CASOLARE** non formano un incastro secondo lo schema **bbaaaabb** (in questo caso l'incastro avrebbe richiesto che C fosse **SOCARELA**).

**Esercizio 2.** Si scriva un programma C++ che, letti da input, nell'ordine, un numero naturale N strettamente maggiore di 0, una matrice di interi di dimensione N\*N, e un naturale K, con  $0 < K \leq N$ , stampi la sottomatrice quadrata di dimensione K\*K, la cui somma degli elementi in essa contenuti è massima. Nel caso in cui esistano più sottomatrici la cui somma degli elementi è massima, si stampi la prima che si incontra analizzando la matrice per righe, dall'alto verso il basso.

ATTENZIONE: perché il programma possa essere valutato correttamente da domjudge, gli elementi della sottomatrice individuata devono essere stampati tutti su un'unica linea, senza alcuna spaziatura.

Si scriva un programma C++ che, letti da input, nell'ordine, un numero naturale N strettamente maggiore di 0, una matrice di interi di dimensione N\*N, e un naturale K, con  $0 < K \leq N$ , stampi la sottomatrice quadrata di dimensione K\*K, la cui somma degli elementi in essa contenuti è massima. Nel caso in cui esistano più sottomatrici la cui somma degli elementi è massima, si stampi la prima che si incontra analizzando la matrice per righe, dall'alto verso il basso.

**ATTENZIONE:** perché il programma possa essere valutato correttamente da domjudge, gli elementi della sottomatrice individuata devono essere stampati tutti su un'unica linea, senza alcuna spaziatura.

#### ESEMPI

- Se N fosse pari a 4, la matrice di dimensione 4\*4 fosse la seguente,

1	-5	2	-2
-4	-3	0	2
0	-6	-3	6
5	4	0	-1

e K fosse pari a 3, si individuerebbero le seguenti 4 sottomatrici di dimensione 3\*3

1	-5	2
-4	-3	0
0	-6	-3

-5	2	-2
-3	0	2
-6	-3	6

-4	-3	0
0	-6	-3
5	4	0

-3	0	2
-6	-3	6
4	0	-1

Le cui somme, rispettivamente, sono -18, -9, -7, -1.

Il programma dovrebbe stampare l'ultima sottomatrice la cui somma è -1. Più in dettaglio, dovrebbe stampare -302-6-3640-1 (ossia -3 0 2 -6 -3 6 4 0 -1 senza spazi).

- Se N fosse pari a 3, la matrice 3\*3 fosse la seguente

-8	-9	1
-4	-1	1
7	-8	2

e K fosse pari a 2, si individuerebbero le seguenti 4 sottomatrici di dimensione 2\*2

-8	-9
-4	-1

-9	1
-1	1

-4	-1
7	-8

-1	1
-8	2

Le cui somme, rispettivamente, sono -22, -8, -6, -6.

Il programma dovrebbe stampare la terza sottomatrice la cui somma è -6. Più in dettaglio, dovrebbe stampare -4-17-8 (ossia -4 -1 7 -8 senza spazi).

