

Esercizio 1 (punti 15) (scrivere la soluzione di seguito)

Scrivere un metodo in linguaggio C++ che prenda in input una matrice di puntatori a interi di dimensioni $n \times m$, un `double z` ed uno `short k` (si assuma $k < n$). Il metodo restituisca il numero di colonne che contengono almeno k elementi contigui la cui media aritmetica risulti minore di z .

In giallo sono evidenziati i parametri che la funzione si aspetta in input da parte dell'utente. È importante notare che k è già assunta essere minore di n quindi non sarà compito della funzione gestire i casi in cui k non è minore di n .

In verde è evidenziato ciò che la funzione darà in output.

Prima di scrivere il codice è buona norma fare un esempio con dei valori scelti da voi.

Ad esempio possiamo considerare:

- $n = 4$
- $m = 3$
- $k = 2$
- $z = 5.5$

Quindi si avrà una matrice 4×3 , 2 è minore di 4, quindi possiamo procedere.

Consideriamo la seguente matrice:

	0	1	2
0	10	5	8
1	11	7	2
2	2	4	8
3	3	1	10

Adesso bisogna scorrere la matrice lungo le colonne, per ogni colonna scorriamo lungo le righe e consideriamo gli elementi contigui di **almeno** k elementi.

Nella matrice di esempio gli elementi contigui di **almeno** k elementi sono:

PER LA PRIMA COLONNA (LA COLONNA 0):

10	11			11	2	
10	11	2		11	2	3
10	11	2	3	2	3	

PER LA SECONDA COLONNA (LA COLONNA 1):

5	7				7	4	
5	7	4			7	4	1
5	7	4	1		4	1	

PER LA TERZA COLONNA (LA COLONNA 2):

8	2				2	8	
8	2	8			2	8	10
8	2	8	10		8	10	

Adesso controlliamo la media degli elementi contigui e se almeno una media risulta minore di $z = 5.5$ allora aumenteremo un contatore, quest'ultimo lo indicheremo con c ed inizialmente $c = 0$.

$$\frac{10 + 11}{2} = 10,5 \stackrel{?}{<} 5,5 \Rightarrow \text{FALSO}$$

$$\frac{10 + 11 + 2}{3} = 7,7 \stackrel{?}{<} 5,5 \Rightarrow \text{FALSO}$$

$$\frac{10 + 11 + 2 + 3}{4} = 6,5 \stackrel{?}{<} 5,5 \Rightarrow \text{FALSO}$$

$$\frac{11 + 2}{2} = 6,5 \stackrel{?}{<} 5,5 \Rightarrow \text{FALSO}$$

$$\frac{11 + 2 + 3}{3} = 5,3 \stackrel{?}{<} 5,5 \Rightarrow \text{VERO}$$

Sono stati trovati almeno k elementi contigui la cui media è minore di 5.5, quindi possiamo incrementare il contatore e passare alla prossima colonna. Quindi $c = 1$.

Sono stati trovati almeno k elementi contigui la cui media è minore di 5.5, quindi possiamo incrementare il contatore e passare alla prossima colonna. Quindi $c = 2$.

$$\frac{5+7}{2} = 6 \stackrel{?}{<} 5,5 \Rightarrow \text{FALSO}$$

$$\frac{5+7+4}{3} = 5,3 \stackrel{?}{<} 5,5 \Rightarrow \text{VERO}$$

Sono stati trovati almeno k elementi contigui la cui media è minore di 5.5, quindi possiamo incrementare il contatore e concludere. Quindi $c = 3$.

$$\frac{8+2}{2} = 5 \stackrel{?}{<} 5,5 \Rightarrow \text{VERO}$$

Il valore restituito dalla funzione sarà 3 per questo specifico esempio.

A questo punto cerchiamo di trasformare in codice ciò che è stato fatto con la sola logica.

Abbiamo visto che bisognerà spostarsi per colonne quindi sarà necessario un primo *ciclo* che ci permetta di spostarci dalla colonna 0 alla colonna m . Inoltre, per ogni colonna bisognerà muoversi lungo le righe quindi sarà necessario un secondo *ciclo* annidato al primo. Infine, bisognerà selezionare i k elementi contigui, quindi sarà necessario un terzo *ciclo* annidato al secondo. In totale avremo 3 cicli annidati. Avremo bisogno, anche, di una variabile contatore che è inizializzata a zero.

```
int esercizio1(int*** M, int n, int m, short k, double z) {
    int c = 0; //Inizializzazione del contatore
    for(int j = 0; j < m; j++) { //ciclo che scorre le colonne
        for(int i = 0; i < n; i++) { //ciclo che scorre le righe
            double sum = 0.0; //mantiene la somma degli elementi contigui
            int seqlen = 0; //lunghezza della sequenza
            bool found = false;
            for(int l = i; l < n; l++) { //scorre gli elementi contigui
                if(M[l][j]){ //controlla se il puntatore non è nullptr
                    sum += M[l][j][0];
                    seqlen++;
                    if(seqlen >= k && sum/seqlen < z) { //controlla se la media degli elementi contigui di almeno k elementi è < z
                        found = true;
                        break;
                    }
                }
            }
            else {
                sum = 0.0;
                seqlen = 0;
            }
        }
        if(found) {
            c++;
            break;
        }
    }
    return c;
}
```

Di seguito é riportato l'esempio eseguito su un calcolatore usando la funzione sopra definita.

```
Inserisci un valore per la posizione [0][0]: 10
Inserisci un valore per la posizione [0][1]: 5
Inserisci un valore per la posizione [0][2]: 8
Inserisci un valore per la posizione [1][0]: 11
Inserisci un valore per la posizione [1][1]: 7
Inserisci un valore per la posizione [1][2]: 2
Inserisci un valore per la posizione [2][0]: 2
Inserisci un valore per la posizione [2][1]: 4
Inserisci un valore per la posizione [2][2]: 8
Inserisci un valore per la posizione [3][0]: 3
Inserisci un valore per la posizione [3][1]: 1
Inserisci un valore per la posizione [3][2]: 10
```

MATRICE

10	5	8
11	7	2
2	4	8
3	1	10

Il numero di colonne che contengono almeno 2 elementi contigui
la cui media aritmetica risulti minore di 5.5 e' 3

Esercizio 2 (punti 15) (scrivere la soluzione sul retro del foglio)

Scrivere un metodo in linguaggio C++ che prenda in input una matrice A di puntatori a stringhe ed un array B di caratteri. Il metodo operi sulle stringhe delle colonne di indice dispari di A depurandole delle vocali eventualmente presenti in B.

In questo caso la funzione prende in input una matrice e un array.

L'idea è la seguente:

1. Controlliamo quali sono le vocali presenti in B e le salviamo su una stringa *vocali*;
2. Se *vocali* è vuota allora abbiamo finito, altrimenti passiamo al punto 3;
3. Operiamo sulle stringhe delle colonne di indice dispari di A e per ogni stringa andiamo ad eliminare le vocali se queste sono presenti in *vocali*.

Vediamo un esempio:

LA MATRICE A È LA SEGUENTE:

	0	1	2
0	"CIAO"	"AIUOLA"	"TRENO"
1	"AB"	"A"	"CD"
2	"BC"	"BICI"	"FG"
3	"ZW"	"AEI"	"OA"

L'ARRAY B È IL SEGUENTE:

0	1	2
'A'	'B'	'U'

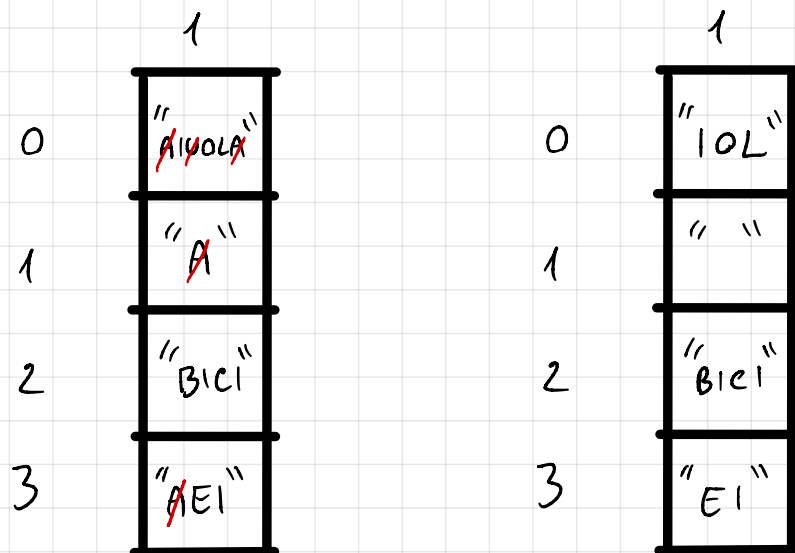
1. Da B consideriamo solo le vocali e le inseriamo in una stringa *vocali*, questo sarà utile quando implementeremo la funzione, per cui si ha:

vocali = "AU"

2. *vocali* non è vuota, quindi possiamo passare al punto 3.

3. Operiamo sulle stringhe delle colonne di indice dispari di A. In questo caso la colonna di indice 1.

Da ogni stringa dobbiamo eliminare le vocali che sono presenti all'interno della stringa *vocali*.



Qui si conclude l'esempio, adesso implementiamo la funzione.

```
void esercizio2(string***A, int n, int m, char* B, int k) {
    string vocali = "";
    for(int i = 0; i < k; i++) {
        //Controlla se il carattere B[i] è una vocale e la inserisce nella stringa vocali in maiuscolo
        switch(toupper(B[i])) {
            case 'A':
            case 'E':
            case 'I':
            case 'O':
            case 'U': {
                vocali += (char)(toupper(B[i]));
                break;
            }
        }
    }
    if(vocali.length() == 0) return; //se non ci sono vocali la funzione termina
    for(int j = 1; j < m; j+=2) { //scorre le colonne dispari
        for(int i = 0; i < n; i++) {
            if(A[i][j]) {
                string newString = "";
                string oldString = A[i][j][0];
                for(int l = 0; l < oldString.length(); l++) {
                    string check = "";
                    check += (char)(toupper(oldString[l]));
                    //string::npos è un valore che viene restituito dalla funzione find quando
                    //accade che la stringa check non è sotto stringa della stringa vocali.
                    if(vocali.find(check) == string::npos) {
                        newString += oldString[l];
                    }
                }
                A[i][j][0] = newString;
            }
        }
    }
}
```

Di seguito é riportato l'esempio eseguito su un calcolatore usando la funzione sopra definita.

```
Inserisci un carattere nel vettore B: a
Inserisci un carattere nel vettore B: b
Inserisci un carattere nel vettore B: u

Inserisci un valore per la posizione [0][0]: ciao
Inserisci un valore per la posizione [0][1]: aiuola
Inserisci un valore per la posizione [0][2]: treno
Inserisci un valore per la posizione [1][0]: ab
Inserisci un valore per la posizione [1][1]: a
Inserisci un valore per la posizione [1][2]: cd
Inserisci un valore per la posizione [2][0]: bc
Inserisci un valore per la posizione [2][1]: bici
Inserisci un valore per la posizione [2][2]: fg
Inserisci un valore per la posizione [3][0]: zw
Inserisci un valore per la posizione [3][1]: aei
Inserisci un valore per la posizione [3][2]: oa

-----
ARRAY
-----
a b u

Matrice prima:

-----
MATRICE
-----
ciao    aiuola  treno
ab      a      cd
bc      bici   fg
zw      aei    oa

Matrice dopo:

-----
MATRICE
-----
ciao    iol     treno
ab      cd
bc      bici   fg
zw      ei     oa
```