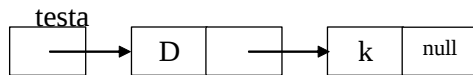


Esercizio 1

Sia data la struttura seguente `struct elem {char info; elem* pun;};`

Scrivere una funzione che, data una lista ordinata di elementi di tipo `elem`, ed un file di caratteri, legge i caratteri dal file e per ogni carattere letto che rappresenta una lettera maiuscola o minuscola dell'alfabeto inserisce un elemento nella lista con campo `info` uguale al carattere letto, mantenendo la lista ordinata.

Per esempio, se la funzione viene chiamata con la lista seguente,



Se nel file sono presenti i seguenti caratteri: *W r B eof*
la lista viene modificata come segue:



Esercizio 2

Scrivere una funzione che data una matrice di interi composta di r righe e c colonne, ed un intero k , restituisce una matrice di r righe e c colonne con gli elementi calcolati come segue. L'elemento in posizione (i, j) e' uguale ad $A * B$, dove A e' la somma dei k elementi che precedono e dei k elementi che seguono (i, j) sulla riga i , e B e' la somma dei k elementi che precedono e dei k elementi che seguono (i, j) sulla colonna j . Assumere $r \geq 2$, $c \geq 2$ e $k \geq 1$.

Se la funzione viene chiamata con la matrice M seguente di dimensione 3×5 e con k uguale a 2, la funzione restituisce la matrice M' :

$$M = \begin{bmatrix} 2 & 1 & 0 & 1 & 0 \\ 3 & 0 & 0 & 4 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad M' = \begin{bmatrix} 3 & 0 & 0 & 5 & 1 \\ 0 & 7 & 0 & 2 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Esercizio 3

Scrivere una funzione ricorsiva che prende come argomento un naturale dispari N maggiore di 1 e stampa a video una 'X' di asterischi con numero di asterischi sulle diagonali uguale a N .

Esempio con $N=5$:

```
* *
* *
*
* *
* *
```

SOLUZIONI

Esercizio 1

```
void inserisci(elem* & testa, const char* s) {
    ifstream in;
    in.open(s);
    char c;
    while(in >> c) {
        if( (c>= 'a' && c <= 'z') || (c>= 'A' && c <= 'Z')) {
            elem* r = new elem;
            r->info = c;
            elem *p, *q;
            for (q=testa; (q!= NULL) && (q->info < c); q=q->pun)
                p=q;
            if (q==testa) {
                r->pun = testa;
                testa = r;
            }
            else {
                p->pun = r;
                r->pun = q;
            }
        } //fine if
    } // fine while
    in.close();
}
```

Esercizio 2

```
int* nuovamatrice(int* m, int r, int c, int k) {

    int* m1 = new int [r*c] ;
    for (int i=0; i< r; i++)
        for (int j=0; j<c; j++) {
            int A=0, B=0;
            for (int s=1; s<=k; s++) {
                if (j+s < c)
                    A+= m[i*c + j + s] ;
                if (j-s >= 0)
                    A+= m[i*c + j - s] ;
            } // fine somma riga
            for (int t=1; t<=k; t++) {
                if (i+t < r)
                    B+= m[(i+t)*c + j] ;
                if (i-t >= 0)
                    B+= m[(i-t)*c + j] ;
            } // fine somma colonna
            m1[i*c + j] = A*B ;
        }
    return m1;
}
```

Esercizio 3

```
stampaX(5,0);
```

```
void stampaX(int n, int k){  
    if (k==n)  
        return;  
    for (int j=0; j<n; j++){  
        if ((j==k) || (j==n-k-1) )  
            cout<<"*";  
        else  
            cout<<" ";  
    }  
    cout<<endl;  
    stampaX(n, k+1);  
}
```