

TESTO

Esercizio 1

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

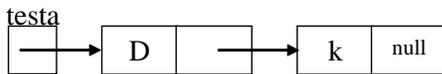
Scrivere una funzione che, data una lista di elementi di tipo `elem`, ed un file di caratteri, legge caratteri dal file e per ogni carattere letto elimina tutti gli elementi con campo informazione uguale al carattere letto dalla lista. La lista passata come argomento alla funzione può contenere duplicati.

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



e nel file sono presenti i seguenti caratteri: `W r eof`

la lista viene modificata come segue:



Esercizio 2

Scrivere una funzione che prende come argomento una matrice M quadrata di dimensione $n \times n$, e restituisce una matrice M' di dimensione $2n \times 2n$ che contiene quattro volte la matrice M come indicato di seguito:

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

La funzione deve inoltre salvare la matrice nel file di nome `output.txt`.

Esercizio 3

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

Esempio con $N=5$:

```
  X
 XXX
XXXXX
 XXX
  X
```

SOLUZIONI

Esercizio 1

```
void elimina (elem*& L, char c)
{ if (L== NULL) return;
  elem *q = L, *p;
  while(q != NULL){
    for( ; q!= NULL && q->info!=n; q=q->pun)
      p = q;
    if (q==NULL) return;
    // elimino elemento puntato da q
    if (q == L) {
      L = L->pun;
      delete q;
      q=L;
    }
    else {
      p->pun = q->pun;
      delete q;
      q=p->pun;
    }
  }
}
```

```
void modificaLista(elem*& p, const char* nomefile) {
  ifstream in(nomefile);
  if (!in)
    return;
  char v;
  while (in >> v)
    elimina(p, v);
}
```

Esercizio 2

```
char* creaMatrice(char* M, int n) {
  int dim = 2*n;
  char* M1 = new char [dim*dim];
  for (int i=0; i< dim; i++)
    for (int j=0; j< dim;j++)
      M1[i*dim +j] = M[(i%n)*n + (j%n)];

  ofstream outf("output.txt");
  for (int i=0; i< dim; i++) {
    for (int j=0; j< dim;j++)
      outf << M1[i*dim+j];
    outf << endl;
  }
  return M1;
}
```

Esercizio 3

Parte superiore:

```
// N/2 -i bianchi  
// (i*2) +1 X
```

Parte inferiore:

```
// i - N/2 bianchi  
// (N-1-i) *2 +1 X
```

```
void stampaRombo(int N, int i=0) {  
    if (i==N)  
        return;  
    if (i < N/2) {  
        for (int h=0; h < (N/2-i); h++)  
            cout << ' '  
        for (int k=0; k < (2*i+1); k++)  
            cout << 'X';  
        }  
    if (i == N/2)  
        for (int k=0; k < N; k++)  
            cout << 'X';  
    if (i > N/2){  
        for (int h=0; h < (i - N/2); h++)  
            cout << ' '  
        for (int k=0; k < (N-1-i)*2 +1; k++)  
            cout << 'X';  
        }  
    cout << endl;  
    stampaRombo(N, i+1);  
}
```