

```

/*
Programma 1lcpp_Liste.cpp
Autore : Riontino Raffaele
        ITIS Ettore Molinari - Milano
        IV informatici corso serale 15/10/10
Funzionalità : gestione di una lista :
                scelta della dimensione iniziale della lista
                modifica della dimensione anche in corso del programma
                inserimento di elementi singoli o in modo random
                eliminazione dell'ultimo elemento inserito o degli ultimi n
elementi inseriti

*/

#include <iostream>
#include <stdlib.h>
#include <conio.h>
using namespace std;

struct nodo{ //struttura di un elemento singolo della lista
    int dato;
    struct nodo *next; //la variabile puntatore next punta ad una
struttura di tipo nodo
};

int MAX = 0; //dimensione della lista

void titolo();

void push(nodo *&testa,int &contatore,int valore); //inserimento di un nuovo
elemento nella lista

void pop(nodo *&testa,int &contatore); //eliminazione dell'ultimo elemento
di una lista

void visualizza(nodo* &testa,int contatore); //visualizzazione di tutti gli
elementi della lista

main()
{
    char scelta; //variabile per la gestione del menù
    int contatore = 0; // contatore degli elementi della lista
    int valore; //variabile del dato da inserire
    int dim; //variabile della quantità di elementi da inserire in modo rand
    int i; //indice del for
    srand((unsigned)time(NULL)); //inizializzazione dei numeri casuali
    nodo *testa = NULL; //creo una lista nulla

    cout << "\nInserisci il numero degli elementi massimi che deve contenere
la lista : ";
    do
        cin >> MAX; //inserimento della dimensione della lista
    while (MAX < 1);

    do{
        titolo(); //chiamata alla funzione titolo

```

```

        cout << "\n    dimensione nodo : " << sizeof(nodo) << " byte\tdimensione
lista : " << (sizeof(nodo) * contatore) << " byte\n";
dimensione attuale della lista
        cout << "\n    nodi inseriti : " << contatore << " di " << MAX << "
disponibili\n\n";
rimanente
        cout << "\n\t1 push manuale(inserisci 1 elemento)\n\t2 push random
(inserisci n elementi casuali)\n\t3 pop (elimina l'ultimo elemento
inserito)\n\t4 pop elimina gli ultimi n elementi inseriti\n\tm modifica la
dimensione della lista\n\tn esci\n"; //menù
        visualizza(testa,contatore); //visualizzo tutti gli elementi presenti
nella lista
        cout << "\nscelta : "; //aspetto l'inserimento una scelta del menu
        scelta = getch();
        switch (scelta)
        {
            case '1' : if ((1 + contatore) > MAX) //controllo se inserendo
un nuovo elemento non supero la dimensione della lista
                cout << "\nlista gia' piena\n";
            else {
                cout << "\ninserisci un valore : ";
                cin >> valore; //prelevo il valore del nuovo
elemento
                push(testa,contatore,valore);
                cout << "push valore inserito : " << testa-
>dato << endl;
            }
            system("pause");
            scelta = 's';
            break;

            case '2' : if (contatore == MAX)
                cout << "\nlista gia' piena\n";
            else
            {
                cout << "\nquantità di numeri casuali da inserire
: ";
                cin >> dim;
                if ((contatore + dim) > MAX)
                    cout << "\n\tdimensione
eccessiva!!\n\n";
                else {
                    for (i = 0 ; i < dim ; i++)
                    {
                        valore = rand()%100+1;
                        push(testa,contatore,valore);
                    }
                    cout << dim << " valori inseriti\n";
                }
            }
            system("pause");
            scelta = 's';
            break;

            case '3' : if (testa != NULL) //controllo se la lista non è
vuota
                {
                    pop(testa,contatore); //richiamo la funzione che
elimina l'ultimo elemento inserito

```

```

        system("pause");
    }
    else {
        cout << "\n la lista e' vuota\n\n";
        system("pause");
    }
    scelta = 's';
    break;

    case '4' : if (testa != NULL)
        {
            cout << "\nquantità di numeri da eliminare : ";
            cin >> dim; //inserisco la quantità degli
ultimi elementi da eliminare
            if (dim > contatore) //controllo se i numeri
da eliminare sono maggiori di quelli presenti nella lista
                {
                    cout << "dimensione eccessiva!!\n";
                }
            else {
                for (i = 0 ; i < dim ; i++)
                    pop(testa,contatore);
            }
        }
    else {
        cout << "\n la lista e' vuota\n";
    }
    system("pause");
    scelta = 's';
    break;

    case 'm' : cout << "\ninserisci la nuova dimensione della lista
: ";

    int temp;
    cin >> temp;
    if (temp >= contatore) //controllo se la nuova
dimensione non supera il numero di elementi già presenti nella lista
        {
            MAX = temp; //assegno a MAX la nuova dimensione
della lista

            cout << "dimensione modificata!\n";
        }
    else
        {
            cout << "\n Impossibile inserire una dimensione
inferiore agli elementi gia' inseriti!!\n";
        }
    system("pause");
    scelta = 's';
    break;

    default : if (scelta != 'n') scelta = 's';
} //chiusura switch
}while (scelta == 's');
} //chiusura main

//funzione titolo contiene il nome del programma e l'autore
void titolo()

```

```

    {
        system("cls");
        cout << "\t 1lcpp_Liste\tRiontino Raffaele IV Informatici";
        cout << "\n\n\tITIS Ettore Molinari - Milano corso serale
15/10/10\n\n";
        cout << "\t Aggiunta e rimozione di elementi ad una lista\n\n";
    }//chiusura titolo

    //funzione push
void push(nodo *&testa,int &contatore,int valore)
{
    nodo *nuovo; //dichiaro un puntatore a nodo
    nuovo = new nodo; //alloco spazio a nuovo delle dimensione del
nodo
    nuovo->dato = valore; //inserisco il valore nella parte dato del
nodo
    nuovo->next = testa; //collego la parte puntatore del nodo alla
testa della lista
    testa = nuovo; //collego la testa della lista al nuovo nodo appena
inserito
    contatore++; //incremento il numero degli elementi presenti
nella lista
} //chiusura push

    //funzione pop
void pop(nodo *&testa,int &contatore)
{
    nodo *nuovo;
    nuovo = new nodo;
    nuovo = testa; //assegno a nuovo il contenuto di esta (l'indirizzo
puntato da testa è il primo elemento della lista)
    cout << "elemento eliminato : " << nuovo->dato << endl;
    testa = testa->next; //faccio puntare a testa il valore secondo valore
della lista
    contatore--; //decremento il numero degli elementi presenti nella lista
delete nuovo; //elimino il puntatore nuovo
    /*
        il puntatore nuovo, dopo avergli assegnato il valore di testa,
puntava al primo elemento della lista
        dopo aver puntato testa al secondo valore della lista (testa =
testa->next), il secondo elemento
        della lista diventa il primo
        eliminando nuovo con delete, si elimina quello che era il primo
elemento della lista
    */
} //chiusura pop

    //funzione visualizza
void visualizza(nodo* &testa , int contatore)
{
    int i;
    nodo*cursore; //dichiaro un nuovo elemento da utilizzare come cursore
per scorrere la lista
    cursore = new nodo;

```

```
    cursore = testa; //faccio puntare cursore al primo elemento della lista
    cout << "\nelementi inseriti : | ";
    for (i = 0 ; i < contatore ; i++)
    {
        cout << cursore->dato << " | ";
        cursore = cursore->next; //faccio avanzare il cursore facendolo
//puntare alla parte next dell'elemento corrente
    }
} //chiusura visualizza
```