

```

/*
Programma : llcpp_Liste1.2
Autore : Riontino Raffaele classe 4 inf serale
        I.T.I.S. Molinari - Milano 17/10/10
Funzionalità : creazione di una lista gestione della
                dimensione, dell'inserimento e
                dell'eliminazione dei campi in testa
                alla lista.
*/

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

//variabili globali
struct nodo{
    int dato;
    struct nodo *next;
};
int MAX= 0; //contiene la dimensione della lista

//prototipi di funzioni
void titolo(); //visualizza il titolo dell'esercizio

char menu(); //visualizza un menu e ritorna la scelta come carattere

int visualizzaLista(nodo *testa); //visualizza il contenuto

void push(nodo *&testa,int dato); //inserisce gli elementi

void pop(nodo *&testa); //elimina l'ultimo elemento inserito

//main principale
main()
{
    nodo *testa = NULL; //creazione di una lista vuoto
    char scelta; //da utilizzare per le scelte nel menu
    int contatore = 0; //conta gli elementi della lista
    int dato; //contiene il dato da inserire in un elemento della lista
    int dim; //serve per cambiare la dimensione della lista
    do{
        titolo();
        contatore = visualizzaLista(testa);
        cout << "\n\n\tdimensione lista : " << MAX << " - dati inseriti : "
<< contatore;
        cout << " - campi liberi : " << (MAX - contatore) << endl << endl;

        scelta = menu();
        switch (scelta)
        {
            case '1' : if ((contatore + 1) > MAX)
                        {
                            cout << "\n\n\tlista gia' piena!\n\n\t";

```

```

        getch();
    }
    else
    {
        cout << "\n\n\tinserisci il dato da salvare : ";
        cin >> dato;
        push(testa,dato);
    }
    scelta = 's';
    break;

case '2' : pop(testa);
        scelta = 's';
        break;

case '3' : cout << "\n\n\tinserisci la dimensione della lista
: ";

        cin >> dim;
        if (dim >= contatore)
        {
            MAX = dim;
            cout << "\n\tDimensione modificata";
        }
        else
        {
            cout << "\n\n\tnon puoi inserire una
dimensione";

            cout << "\n\tminore del contenuto della
lista!";

        }
        getch();
        scelta = 's';

case 'e' : break;

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

//definizioni di funzioni
void titolo()
{
    system("cls");
    cout << "\t\tGestione di una lista\n\n";
} //chiusura titolo

char menu()
{
    char s;
    cout << "\n\n\t\tmenu\n\n";
    cout << "\n\t1 - push";
    cout << "\n\t2 - pop";
    cout << "\n\t3 - dimensione lista";
    cout << "\n\te - esci";
    s = getch(); // catturo un solo carattere

```

```

    s = tolower(s); //converto il valore inserito in minuscolo
    return s;
} //chiusura menu

int visualizzaLista(nodo *testa)
{
    int conto = 0;
    nodo * cursore; // puntatore da utilizzare per scorrere la lista
    cursore = testa; //assegno a cursore l'indirizzo puntato da testa
    if (cursore) cout << " | ";
    while (cursore != NULL)
    {
        cout << cursore->dato << " | "; //visualizza
        contenuto
        cursore = cursore->next; //passo all'elemento
        successivo
        conto++;
    }
    return conto;
} //chiusura visualizzaTesta

void push(nodo *&testa, int dato)
{
    nodo * nuovo; //dichiaro un nuovo puntatore a nodo
    nuovo = new nodo; //alloco spazio di nodo al puntatore nuovo
    nuovo->next = testa; //assegno a nuovo->next l'indirizzo puntato da
    testa
    nuovo->dato = dato; //inserisco il dato nell'elemento nuovo
    testa = nuovo; //assegno a testa l'indirizzo dell'elemento nuovo
} //chiusura push

void pop(nodo *&testa)
{
    if (testa != NULL)
    {
        nodo *nuovo; //dichiaro un nuovo puntatore a nodo
        nuovo = testa; // assegno al nuovo elemento l'indirizzo di testa
        testa = testa->next; //assegno a testa il valore successivo (2)
        delete nuovo; //elimino l'elemento che conteneva l'indirizzo di
        testa
    }
    else
    {
        cout << "\n\n\tlista vuota!!\n\n";
        getch();
    }
} //chiusura pop

```