

```

/*
Programma : cppListaTelefonica1.1.cpp
Autore : Riontino Raffaele 4 informatici
        ITIS Molinari - Milano corso serale
        4/11/2010
Funzionalità : gestione di una lista contenente
               contatti telefonici.
               Il programma permette di:
               aggiungere nuovi contatti;
               visualizzare tutta la lista;
               visualizzare il contatto cercato;
               eliminare il contatto cercato.
Aggiornamento : modificate le funzioni per
5/11/2010      l'inserimento dei dati con readNomi
               e readCell (per i nomi e per i numeri
               di cellulare inseriti i controlli sui caratteri)
Aggiornamento : modifica alla funzione dell'aggiunta dei contatti
10/11/2010     l'inserimento dei contatti avviene in ordine
               alfabetico per cognome e, in caso di cognome
               già presente , per nome.

*/

#include <iostream>
#include <conio.h>
#include <windows.h>
#define MAX 1000 //massimo nodi inseribili
#define DIMC 31  //lunghezza stringhe nomi cognomi
#define DIMN 15  //lunghezza stringhe cellulare
using namespace std;

struct nodo{ //struttura del contatto
    char nome[DIMC];
    char cognome[DIMC];
    char cellulare[DIMN];
    struct nodo *next;
};

unsigned attuali = 0; //numero di nodi inseriti


void titolo();

char menu();

void readNomi(char[],char[],unsigned); //input stringa
void readCell(char[],char[],unsigned); //input stringa
void creaNodo(nodo*&);

void aggiungi(nodo *&,nodo*); //aggiunge contatti alla lista
void visualizza(nodo*); //visualizza intera lista
void cerca(nodo*); //visualizza singolo contatto
void cancellaC(nodo*&); //elimina contatto scelto

```

```
int main()
{
    system("color F1");
    nodo *testa = NULL; //creazione lista vuota
    char scelta;
    do{
        titolo();
        scelta = menu(); //input scelta
        switch(scelta)
        {
            case 'a' : titolo();
                        creaNodo(testa); //aggiungo lelemento
                        scelta = 's';
                        break;
            case 'v' : titolo();
                        visualizza(testa); //visualizzo lista
                        scelta = 's';
                        break;
            case 'c' : titolo();
                        cerca(testa); //visualizzo ricerca
                        scelta = 's';
                        break;
            case 'd' : titolo();
                        cancellaC(testa); //cancello contatto cercato
                        scelta = 's';
                        break;
            case 'e' : titolo();
                        cout << "\n\n\n\t\t\t Programma terminato!!!";
                        Sleep(1500);
                        scelta = 'e';
                        break;
            default : scelta = 's';
        }//chiusura switch
    }while (scelta != 'e');
    return 1;
} //chiusuta main


void titolo()
{
    //descrizione del titolo
    system("cls");
    cout << "\n\t\tl3cppListaTelefonica1.1.cpp      Riontino Raffaele\n";
    cout << "\n\tITIS Molinari - Milano classe 4 informatici corso serale
10/11/2010\n";
    cout << "\n\t\t\tGESTIONE DI UNA RUBRICA TELEFONICA\n";
    cout << "\n\t\t\tcontatti presenti : " << attuali << "\tspazio disponibile
: " << (MAX - attuali) << endl;
    } //chiusura titolo


char menu()
{
    //visualizza le scelte disponibili e ritorna la scelta alla chiamata
    char scelta;
    cout << "\n\t\tMENU'\n";
    cout << "\n\t( a ) Aggiungi contatto\n";
    cout << "\n\t( v ) Visualizza lista\n";
    cout << "\n\t( c ) Cerca contatto\n";
    cout << "\n\t( d ) Cancella contatto\n";
```

```

    cout << "\n\t( e ) Esci\n";
    cout << "\n\tdigita la scelta : ";
    scelta = getch();
    scelta = tolower(scelta); //converto il carattere in minuscolo
    return scelta;
} //chiusura menu

void creaNodo(nodo*&testa)
{
    titolo();
    nodo *nuovo = new nodo; //dichiarazione nuovo nodo
    cout << "\n\n\tNuovo contatto : \n\n";
    readNomi("\n\taggiungi cognome : ",nuovo->cognome,DIMC);
    readNomi("\n\taggiungi nome : ",nuovo->nome,DIMC);
    readCell("\n\tinserisci numero telefonico : ",nuovo->cellulare,DIMN);
    aggiungi(testa,nuovo);
} //chiusura creaNodo

void aggiungi(nodo*& testa,nodo* nuovo)
{
    if ((attuali+1) > MAX) //controllo se con nuovo contatto supero il limite
    {
        titolo();
        cout << "\n\n\tImpossibile aggiungere altri contatti. Lista gia'
piena!!";
        getch();
        return; //blocco l'esecuzione della funzione
    }
    titolo();
    nodo *scorri,*bloc;
    //se la lista è vuota o se l'elemento va in testa
    if (testa == NULL || strcmp(testa->cognome,nuovo->cognome) > 0 ||
(strcmp(testa->cognome,nuovo->cognome) == 0 && strcmp(testa->nome,nuovo->nome)
>= 0))
    {
        nuovo->next = testa;
        testa = nuovo;
        attuali++;
        cout << "\n\n\tContatto aggiunto!!";
        getch();
        return;
    }
    scorri = testa; //se ha più elementi
    while(scorri->next != NULL)
    {
        if (strcmp(scorri->next->cognome,nuovo->cognome) > 0 ||
(strcmp(scorri->next->cognome,nuovo->cognome)==0 && strcmp(scorri->next->nome,
nuovo->nome) >= 0))
        {
            bloc = scorri->next;
            scorri->next = nuovo;
            nuovo->next = bloc;
            attuali++;
            cout << "\n\n\tContatto aggiunto!!";
            getch();
            return;
        }
    }
}

```

```

        scorri = scorri->next;
    }
    //se arriva qui va in fondo
    nuovo->next = NULL;
    scorri->next = nuovo;
    attuali++;
    cout << "\n\n\tContatto aggiunto!!";
    getch();
    return;

} //chiusura aggiungi

void visualizza(nodo *testa)
{
    if (!testa)
    {
        cout << "\n\n\tLista vuota!!";
        getch();
    }
    int conto = 1;
    for (nodo * scorri = testa ; scorri != NULL ; scorri = scorri->next)
    {
        titolo();
        cout << "\n\n\tContatto n." << conto << endl;
        cout << "\n\n\tCognome : " << scorri->cognome << endl;
        cout << "\tNome : " << scorri->nome << endl;
        cout << "\tCellulare : " << scorri->cellulare << "\n\n\t";
        system("pause");
        conto++;
    }
} //chiusura visualizza

void cerca(nodo *testa)
{
    if (!testa)
    {
        cout << "\n\n\tLista vuota!!!";
        getch();
    }
    else {
        char ricerca[DIMC];
        short conto = 0;
        readNomi("\n\n\tInserisci il cognome del contatto : " , ricerca,
DIMC);

        for (nodo * scorri = testa ; scorri != NULL ; scorri = scorri->next)
        {
            if (strcmp(ricerca, scorri->cognome) == 0)
            {
                titolo();
                cout << "\n\n\tCognome : " << scorri->cognome << endl;
                cout << "\tNome : " << scorri->nome << endl;
                cout << "\tCellulare : " << scorri->cellulare << "\n\n\t";
                conto++;
                getch();
            }
        }
    }
}

```

```

        if (!conto)
        {
            cout << "\n\n\tContatto non in lista!!!";
            getch();
        }
        else {
            cout << "\n\n\tContatti trovati : " << conto;
            getch();
        }
    }
} //chiusura cerca

void cancellaC(nodo*&testa)
{
    if (testa == NULL)
    {
        cout << "\n\n\tLista vuota!!!";
        getch();
        return;
    }
    char scelta;
    char ricerca[DIMC];
    short conto = 0;
    readNomi("\n\n\tInserisci il cognome del contatto da eliminare : ",
ricerca, DIMC);
    nodo *scorri, *del;
    if (testa->next != NULL)
    {
        scorri = testa;
        while(scorri->next != NULL)
        {
            if (strcmp(ricerca, scorri->next->cognome) == 0)
            {
                conto++;
                titolo();
                cout << "\n\n\tContatto trovato : ";
                cout << "\n\n\tcognome : " << scorri->next->cognome;
                cout << "\n\n\tnome : " << scorri->next->nome;
                cout << "\n\n\tcellulare : " << scorri->next->cellulare;
                cout << "\n\n\tCancellare questa contatto (s / n) ? ";
                do
                {
                    scelta = getch();
                    scelta = tolower(scelta);
                } while(scelta != 's' && scelta != 'n');
                if (scelta == 's')
                {
                    del = scorri->next;
                    scorri->next = scorri->next->next;
                    delete del;
                    cout << "\n\n\tContatto eliminato!!!";
                    getch();
                    attuali--;
                    return;
                }
            }
            else scorri = scorri->next;
        }
    }
}

```

```

        }
        else scorri = scorri->next;
    }
}
if (strcmp(ricerca, testa->cognome) == 0)
{
    conto++;
    titolo();
    cout << "\n\n\tContatto trovato : ";
    cout << "\n\n\tcognome : " << testa->cognome;
    cout << "\n\n\tnome : " << testa->nome;
    cout << "\n\n\tcellulare : " << testa->cellulare;
    cout << "\n\n\tCancellare questa contatto (s / n) ? ";
    do
    {
        scelta = getch();
        scelta = tolower(scelta);
    } while (scelta != 's' && scelta != 'n');
    if (scelta == 's')
    {
        del = testa;
        testa = testa->next;
        delete del;
        cout << "\n\n\tContatto eliminato!!";
        getch();
        attuali--;
    }
}
if (conto == 0)
{
    cout << "\n\n\tNessun contatto trovato con : " << ricerca << " ";
    getch();
}
} //chiusura cancellaC

void readNomi(char messaggio[], char stringa[], unsigned dim)
{
    short i;
    char *carattere = new char;
    cout << messaggio; //messaggio inviato dall'utente
    i = 0;
    do{
        *carattere = getch(); //catturo un carattere
        if (((*carattere >= 'a' && *carattere <= 'z') || (*carattere >=
'A' && *carattere <= 'Z') || *carattere == 32 || *carattere == '\\') && i < dim
)
        {
            cout << *carattere;
            stringa[i] = *carattere;
            i++;
        }
        if ((int)*carattere == 8 && i > 0) //se tasto delete
        {
            cout << *carattere << ' ' << *carattere;
            i--;
        }
    } while ((int)*carattere != 13 || i == 0); //ripeti se carattere
diverso da invio

```

```

        stringa[i] = '\\0'; //inserisco finestringa
        for (i = 0 ; i < strlen(stringa) ; i++)
            stringa[i] = tolower(stringa[i]);
        delete carattere;
    } //chiusura readNomi

void readCell(char messaggio[],char stringa[],unsigned dim)
{
    short i;
    char *carattere = new char;
    cout << messaggio; //messaggio inviato dall'utente
    i = 0;
    do{
        *carattere = getch(); //catturo un carattere
        if ((*carattere >= '0' && *carattere <= '9')) && i < dim )
        {
            cout << *carattere;
            stringa[i] = *carattere;
            i++;
        }
        if ((int)*carattere == 8 && i > 0) //se tasto delete
        {
            cout << *carattere << ' ' << *carattere;
            i--;
        }
    }while ((int)*carattere != 13 || i == 0); //ripeti se carattere
diverso da invio
    stringa[i] = '\\0'; //inserisco finestringa
    delete carattere;
} //chiusura readNomi

```