

```

/*
Programma : cppListaTelefonica1.2.cpp
Autore : Riontino Raffaele 4 informatici
        ITIS Molinari - Milano corso serale
        3/12/2010
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.
Aggiornamento : Aggiunto il salvataggio dei dati su un file csv
22/11/2010

*/

```

```

#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

bool modifica;

char nomefile[20] = {"lista.csv"};

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 cancella(nodo*&); //elimina contatto scelto

void cancellaContatto(nodo*&,nodo*); //elimina contatto scelto

void salva(nodo*);

void carica(nodo*&);

void modificaContatto(nodo*&);

int main()
{
    system("color F1");
    nodo *testa = NULL; //creazione lista vuota
    carica(testa);
    char scelta;
    modifica = false;
    do{
        titolo();
        scelta = menu(); //input scelta
        switch(scelta)
        {
            case 'a' : titolo();
                creaNodo(testa); //aggiungo lelemento
                cout << "\n\n\tContatto aggiunto!!";
                Sleep(1000);
                break;
            case 'v' : titolo();
                visualizza(testa); //visualizzo lista
                break;
            case 'c' : titolo();
                cerca(testa); //visualizzo ricerca
                break;
            case 'd' : titolo();
                cancella(testa); //cancello contatto cercato
                break;
            case 'm' : modificaContatto(testa);
                break;
            case 's' : salva(testa);
                break;
            case 'f' : system(nomefile);
                break;
            case 'e' : titolo();
                if (modifica)
                {
                    cout << "\n\n\tLa lista e' stata modificata, vuoi
salvare i dati (s / n) ?";
                    char *s = new char;
                    do
                    {
                        *s = getch();
                        *s = tolower(*s);
                    }while(*s != 's' && *s != 'n');
                }
        }
    }
}

```



```

        return; //blocco l'esecuzione della funzione
    }
    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++;
        modifica = true;
        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++;
            modifica = true;
            return;
        }
        scorri = scorri->next;
    }
    //se arriva qui va in fondo
    nuovo->next = NULL;
    scorri->next = nuovo;
    attuali++;
    modifica = true;
    return;
} //chiusura aggiungi

```

```

void visualizza(nodo *testa)
{
    if (!testa)
    {
        cout << "\n\n\tLista vuota!!";
        getch();
    }
    int conto = 1;
    char scelta;
    for (nodo * scorri = testa ; scorri != NULL && scelta != 'm'; 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";
        cout << "\n\n\tm -> torna al menu\tc -> continua";
        do{

```

```

        scelta = getch();
        scelta = tolower(scelta);
        }while (scelta != 'm' && scelta != 'c');
        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 cancella(nodo*&testa)
{
    if (testa == NULL)
    {
        cout << "\n\n\tLista vuota!!!";
        getch();
        return;
    }
    char scelta;
    char ricerca[DIMC];
    short conto = 0;
    titolo();

```

```

    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\tnome : " << scorri->next->nome;
                cout << "\n\tcellulare : " << scorri->next->cellulare;
                cout << "\n\n\tEliminare questo contatto (s / n) ? ";
                do
                {
                    scelta = getch();
                    scelta = tolower(scelta);
                }while(scelta != 's' && scelta != 'n');
                if (scelta == 's')
                {
                    cancellaContatto(testa, scorri);
                    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\tnome : " << testa->nome;
        cout << "\n\tcellulare : " << testa->cellulare;
        cout << "\n\n\tEliminare questo contatto (s / n) ? ";
        do {
            scelta = getch();
            scelta = tolower(scelta);
        }while(scelta != 's' && scelta != 'n');
        if (scelta == 's')
        {
            cancellaContatto(testa, testa);
            return;
        }
    }
    if (conto == 0)
    {
        cout << "\n\n\tNessun contatto trovato con : " << ricerca << " ";
        getch();
    }
} //chiusura cancella

```

```

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

```

```

void salva(nodo* testa)

```

```

{
FILE *file;
file = fopen(nomefile, "wt");
fputs("COGNOME ; NOME ; TELEFONO \n\n", file);
for (nodo * scorri = testa ; scorri != NULL ; scorri = scorri->next)
{
fputs(scorri->cognome, file);
fputc(';', file);
fputs(scorri->nome, file);
fputc(';', file);
fputs(scorri->cellulare, file);
fputc('\n', file);

}
fclose(file);
cout << "\n\n\tContatti salvati!!\n\n";
modifica = false;
} //chiusura salva

void carica(nodo*& testa)
{
FILE *file;
char temp[DIMC*3];
if (file = fopen(nomefile, "rt"))
{
int i = 0;
system("cls");
cout << "\n\tCaricamento contatti da file\n\n";
while (fgets(temp, (DIMC*3), file))
{
if (i > 1)
{
nodo* nuovo = new nodo;
char * nome, *cognome, *telefono;
cognome = strtok(temp, ";");
strcpy(nuovo->cognome, cognome);
nome = strtok(NULL, ";");
strcpy(nuovo->nome, nome);
telefono = strtok(NULL, "\n");
strcpy(nuovo->cellulare, telefono);
cout << "\tcontatto [" << (i-1) << " ] : " << nuovo->cognome
<< " " << nuovo->nome << " " << nuovo->cellulare << endl;
aggiungi(testa, nuovo);
Sleep(100);
}
i++;
}
cout << "\n\n\t " << (i-2) << " Contatti caricati correttamente!!";
Sleep(2500);
}
else {
titolo();
cout << "\n\n\tNessuna lista di contatti trovata!!";
Sleep(2500);
return;
}
}

```

```

        fclose(file);
    } //chiusura carica

void modificaContatto(nodo* &testa)
{
    if (testa == NULL)
    {
        cout << "\n\n\tLista vuota!!!";
        getch();
        return;
    }
    char scelta;
    char ricerca[DIMC];
    short conto = 0;
    titolo();
    readNomi("\n\n\tInserisci il cognome del contatto da modificare : ",
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\tnome : " << scorri->next->nome;
                cout << "\n\tcellulare : " << scorri->next->cellulare;
                cout << "\n\n\tModificare questo contatto (s / n) ? ";
                do
                {
                    scelta = getch();
                    scelta = tolower(scelta);
                } while(scelta != 's' && scelta != 'n');
                if (scelta == 's')
                {
                    nodo *nuovo = new nodo; //dichiarazione nuovo nodo
                    titolo();
                    cancellaContatto(testa, scorri);
                    cout << "\n\n\tNuovi dati : \n\n";
                    readNomi("\n\tcognome : ", nuovo->cognome, DIMC);
                    readNomi("\n\tnome : ", nuovo->nome, DIMC);
                    readCell("\n\tnumero telefonico : ", nuovo->cellulare,
DIMN);

                    aggiungi(testa, nuovo);
                    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\tModificare questo contatto (s / n) ? ";
    do {
        scelta = getch();
        scelta = tolower(scelta);
    }while(scelta != 's' && scelta != 'n');
    if (scelta == 's')
    {
        nodo *nuovo = new nodo; //dichiarazione nuovo nodo
        titolo();
        cancellaContatto(testa, testa);
        cout << "\n\n\tNuovi dati : \n\n";
        readNomi("\n\tcognome : ", nuovo->cognome, DIMC);
        readNomi("\n\tnome : ", nuovo->nome, DIMC);
        readCell("\n\tnumero telefonico : ", nuovo->cellulare, DIMN);
        aggiungi(testa, nuovo);
        return;
    }
}
if (conto == 0)
{
    cout << "\n\n\tNessun contatto trovato con : " << ricerca << " ";
    getch();
}
} //chiusura modificaContatto

void cancellaContatto(nodo*&testa, nodo*ricerca)
{
    if (testa == NULL)
    {
        cout << "\n\n\tLista vuota!!!";
        getch();
        return;
    }
    nodo *scorri, *del;
    if (testa->next != NULL)
    {
        scorri = testa;
        while(scorri->next != NULL)
        {
            if (strcmp(ricerca->cognome, scorri->next->cognome) == 0 &&
strcmp(ricerca->nome, scorri->next->nome) == 0 && strcmp(ricerca->cellulare,
scorri->next->cellulare) == 0)
            {
                del = scorri->next;
                scorri->next = scorri->next->next;
                delete del;
                attuali --;
                modifica = true;
                return;
            }
        }
    }
}

```

```
        else scorri = scorri->next;
    }
}
if (strcmp(ricerca->cognome, testa->cognome) == 0 && strcmp(ricerca->nome,
testa->nome) == 0 && strcmp(ricerca->cellulare, testa->cellulare) == 0)
{
    del = testa;
    testa = testa->next;
    delete del;
    attuali--;
    modifica = true;
    return;
}
} //chiusura cancellaContatto
```