

```

/*
Programma : cppListaTelefonica1.0.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.

*/

#include <iostream>
#include <conio.h>
#define MAX 1000 //massimo nodi inseribili
#define DIM 30   //lunghezza stringhe nomi cognomi
using namespace std;

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

unsigned attuali = 0; //numero di nodi inseriti


void titolo();

char menu();

void readString(char[],char[],unsigned); //input stringa

int readInt(char[]); //input intero

void aggiungi(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();

```

```

        aggiungi(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' : scelta = 'e';
        break;
    default : scelta = 's';
} //chiusura switch
}while (scelta != 'e');
return 1;
} //chiusura main

void titolo()
{ //descrizione del titolo
    system("cls");
    cout << "\n\t\tt13cppListaTelefonica1.0.cpp    Riontino Raffaele\n";
    cout << "\n\t\tITIS Molinari - Milano  classe 4 informatici corso serale
3/11/2010\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";
    cout << "\n\t( v ) visualizza lista";
    cout << "\n\t( c ) cerca contatto";
    cout << "\n\t( d ) cancella contatto";
    cout << "\n\t( e ) esci";
    cout << "\n\n\t digita la scelta : ";
    scelta = getch();
    scelta = tolower(scelta); //convertito il carattere in minuscolo
    return scelta;
} //chiusura menu

void aggiungi(nodo *&testa)
{ //aggiunta di un nuovo contatto
    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
    }
    nodo *nuovo = new nodo; //dichiarazione nuovo nodo
    cout << "\n\n\tNuovo contatto : \n\n";
    readString("\taggiungi cognome : ",nuovo->cognome,DIM);
    readString("\taggiungi nome : ",nuovo->nome,DIM);
    do
        nuovo->cellulare = readInt("\tinserisci numero telefonico : ");
    while(nuovo->cellulare < 1);
    nuovo->next = testa;
    testa = nuovo;
    attuali++;
    cout << "\n\n\tContatto aggiunto!!!";
    getch();
} //chiusura nodo

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

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

        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[DIM];
    short conto = 0;
    readString("\n\n\tInserisci il cognome del contatto da eliminare : ",
ricerca, DIM);
    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 readString(char messaggio[], char stringa[], unsigned dim)
{
    bool errore, flag; //per la gestione del controllo degli errori
    int i;
    char provvisorio[100]; //stringa provvisoria
    cout << messaggio; //visualizzo il messaggio ricevuto dalla chiamata
    do
    {
        errore = false;
        cin.getline(provvisorio, 100);
        if (strlen(provvisorio) > (dim-1) || strlen(provvisorio) < 1)
        {
            cout << " Dimensione non consentita. Ripeti!!\n";
            errore = true;
        }
        if (!errore)
        {

```

```

        for (i=0 ; i<dim ; i++) //conversione in minuscolo
            provvisorio[i]=tolower(provvisorio[i]);
        i=0;
        flag = false;
        while ((!flag) && (i < strlen(provvisorio)))
        {
            //controllo che vengano inseriti solo lettere,
            spazi e apostrofi
            if (!isalpha(provvisorio[i]) && provvisorio[i] != 32 &&
provvisorio[i] != 39)
            {
                cout << " Ci sono caratteri non consentiti.Ripeti!!\n";

                flag = true;
                errore = true;
            }

            i++;
        }
    }while(errore);
    provvisorio[dim] = '\0'; //aggiungo finestringa
    strcpy(stringa,provvisorio);
    //cin.ignore();
} //chiusura readString

int readInt(char messaggio[])
{
    int massimo = 2147483646; //massimo intero
    int minimo = -2147483647; //minimo intero
    int dato = 0,i;
    bool errore;
    const int dim = 11;
    char stringa[dim],carattere;
    cout << messaggio; //messaggio inviato dall'utente
    do{
        errore = false; //inizializzo errore a false
        i = 0;
        do{
            carattere = getch(); //catturo un carattere
            if (carattere >= '0' && carattere <= '9' && i <= dim ||
(carattere == '-' && i == 0))
            {
                cout << carattere;
                stringa[i] = carattere;
                i++;
            }
            if ((int)carattere == 8 && i > 0) //se tasto delete
            {
                cout << carattere << ' ' << carattere;
                i--;
            }
        }while ((int)carattere != 13); //ripeti se carattere diverso da
        invio

        stringa[i] = '\0'; //inserisco finestringa
        if ((atoi(stringa) >= minimo && atoi(stringa) <= massimo) && i <
dim) dato = atoi(stringa);
        else
        {
            cout << "\n\a Numero non consentito. Ripeti!!!";

```

```
        errore = true;
    }
    if (!strlen(stringa)) cout << 0;
    cout << endl;
}while(errore);
return dato;
} //chiusura readInt
```