```cpp
/*
  Programma 12cppEse1.cpp
  Autore : Riontino Raffaele 4 informatici
           ITIS Molinari - Milano corso serale
           2/11/2010
  Testo : Si vuole realizzare un tipo struct, utilizzato per
          informazioni su biglietti per voli aerei, avente i
          seguenti campi:
          • nome: stringa di lunghezza inferiore a 30 caratteri
            rappresentante il nome del passeggero
          • prezzo: sotto-struttura di 3 campi (prezzo, tasse,
            prezzoTotale) indicante il costo del biglietto
          • partenza: aeroporto di partenza (stringa di 3 caratteri)
          • arrivo: aeroporto di arrivo (stringa di 3 caratteri)
          • orarioAndata, orarioRitorno: 2 sotto-strutture contenenti
            data e ora di partenza del volo di andata a del volo di
            ritorno.
          Con tale tipo si vogliono realizzare liste dinamiche.
          La struct deve quindi essere realizzata come struttura
          ricorsiva, con puntatore a un dato dello stesso tipo.
          Si definiscano in C il tipo struct, utilizzando due diversi
          schemi: (a) una struct a un solo livello, contenente tutti i campi,
          puntatore ricorsivo compreso; (b) una struct a due livelli,
          nella quale a primo livello si accede a un puntatore ricorsivo e
          ad una sotto-struttura contenente il resto dei dati (altre
          sotto-strutture comprese).
*/

#include <iostream>
#include <conio.h>
#include <string.h>
#define DIM 16  //dimensione nome + 1 per aggiunta \0
#define DIM3 4  //dimensione partenza e arrivo
using namespace std;

struct costoBiglietto {  //struttura per prezzo
                      float prezzo;
                      float tasse;
                      float prezzoTotale;
                      };

struct orarioAndata{ //struttura data andata
                             int giorno;
                             int mese;
                             int anno;
                             int ora;
                             int minuti;
                             };

struct orarioRitorno{ //struttura data ritorno
                             int giorno;
                             int mese;
                             int anno;
                             int ora;
                             int minuti;
                             };

struct biglietto
```

```
        {  //struttura biglietto
         char nome[DIM];
           costoBiglietto *costo;
         char partenza[DIM3];
         char rientro[DIM3];
           orarioAndata *andata;
           orarioRitorno *ritorno;
         struct biglietto * next;
        };

void titolo();

void aggiungi(biglietto *&testa); //aggiunge nuovi elementi alla lista

int visualizza(biglietto *testa); //visualizza la lista completa

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

float readFloat(char []); //ritorna un float

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

void pop(biglietto *&); //elimina elemento

char menu(); //ritorna la scelta dal menu


main()
      {
       char scelta;  //variabile per scelta
       biglietto *testa = NULL; //lista vuota
       do{
          titolo();
          scelta = menu();
          switch (scelta)
              {
               case 'a' : titolo();
                          aggiungi(testa);
                          scelta = 's';
                          break;
               case 'v' : titolo();
                          visualizza(testa);
                          system("pause");
                          scelta = 's';
                          break;
               case 'd' : pop(testa);
                          scelta = 's';
                          break;
               case 'e' : scelta = 'n';
                          break;
               default : scelta = 's';
              }//chiusura switch
          }while (scelta == 's');

      }//chiusura main


void titolo()
```

```
     {
     system("cls");
     cout << "\n\t\tbiglietto aereo\n\n";
     }//chiusura titolo

char menu()
     {
      char s;
      cout << "\n\ta - aggiungi biglietto";
      cout << "\n\tv - visualizza lista";
      cout << "\n\td - elimina primo";
      cout << "\n\te - esci";
      cout << "\n\n\tscelta = ";
      s = getch();
      s = tolower(s);
      return s; //ritorno la scelta
     }//chiusura menu


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 << " Dimenisone 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


float readFloat(char messaggio[])
      {
       float dato = 0;
       const int dim = 7; //6
       int i;
       float minimo = 1.17549e-038;
       float massimo = 3.40282e+038;
       bool errore,punto;
       char stringa[dim],carattere;
       cout << messaggio;
       do{
         errore = false;  //inizializzo errore a false
         punto = false;
         i = 0;
         do{
            carattere = getch(); //catturo un carattere
            if (carattere >= '0' && carattere <= '9' && i <= dim || (carattere
== '-' && i == 0) || (carattere == '.' && !punto && i > 0))
               {
                cout << carattere;
                if (carattere == '.') punto = true; //se il carattere è il
punto memorizzo
                stringa[i] = carattere;
                i++;
               }
            if ((int)carattere == 8 && i > 0)  //se tasto delette
                        {
                         cout << carattere << ' ' << carattere;
                         i--;
                         if (stringa[i] == '.') punto = false;
                         }

           }while ((int)carattere != 13); //ripeti se carattere diverso da invio
           stringa[i] = '\0';  //inserisco finestringa
         if ((atof(stringa) >= minimo && atof(stringa) <= massimo) && i < dim)
dato = atof(stringa);
         else
              {
              cout << "\n\a Numero non consentito. Ripeti!!\n";
              errore = true;
              }
         if (!strlen(stringa)) cout << 0;
         cout << endl;
        }while(errore);
       return dato;
      }//chiusura readFloat


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 delette
                        {
                         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



void aggiungi(biglietto *&testa)
     {
      biglietto * nuovo;
      nuovo = new biglietto;
      nuovo->next = testa;
      readString("Inserisci il nome del passeggero : ",nuovo->nome,DIM);
      nuovo->costo = new costoBiglietto;
      cout << "\ncosto biglietto\n";
      nuovo->costo->prezzo = readFloat("prezzo : ");
      nuovo->costo->tasse = readFloat("tasse : ");
      nuovo->costo->prezzoTotale = nuovo->costo->prezzo + nuovo->costo->tasse;
      cout << "prezzo totale : " << nuovo->costo->prezzoTotale;
      readString("\n\nareoporto di partenza (max 3 caratteri) : ",nuovo-
>partenza,DIM3);

      nuovo->andata = new orarioAndata;
      cout << "\n\ndata partenza\n";

      nuovo->andata->giorno = readInt("giorno : ");

      nuovo->andata->mese = readInt("mese : ");
```

```
      nuovo->andata->anno = readInt("anno : ");

      nuovo->andata->ora = readInt("ora : ");

      nuovo->andata->minuti = readInt("minuti : ");
     readString("\naeroporto di arrivo (max 3 caratteri) : ",nuovo->rientro,
DIM3);

      nuovo->ritorno = new orarioRitorno;
      cout << "\n\ndata ritorno\n";

      nuovo->ritorno->giorno = readInt("giorno : ");

      nuovo->ritorno->mese = readInt("mese : ");

      nuovo->ritorno->anno = readInt("anno : ");

      nuovo->ritorno->ora = readInt("ora : ");

      nuovo->ritorno->minuti = readInt("minuti : ");
      testa = nuovo;
      cout << "\n\ndati inseriti!!";
      getch();
     }//chiusura aggiungi

int visualizza(biglietto *testa)
    {
     int conto = 0;
     titolo();
     biglietto *cursore;
     cursore = testa;
     while(cursore != NULL)
       {
        cout << "nome passeggero : " << cursore->nome;
        cout << "\nPartenza : " << cursore->partenza << " - " << cursore-
>andata->giorno << "/" << cursore->andata->mese << "/" << cursore->andata->anno;
        cout << " - ore : " << cursore->andata->ora << ":" << cursore->andata-
>minuti;
        cout << "\nRitorno : " << cursore->rientro << " - " << cursore->ritorno-
>giorno << "/" << cursore->ritorno->mese << "/" << cursore->ritorno->anno;
        cout << " - ore : " << cursore->ritorno->ora << ":" << cursore->ritorno-
>minuti;
        cout << "\n\nprezzo : " << cursore->costo->prezzo;
        cout << "\ntasse : " << cursore->costo->tasse;
        cout << "\nprezzo totale : " << cursore->costo->prezzoTotale << endl <<
endl;
        getch();
        cursore = cursore->next;
        conto++;
       }
     if (!testa) cout << "\n\n\tLista vuota!!!";
    }//chiusura visualizza


void pop(biglietto *&testa)
     {
      if (testa != NULL)
         {
```

```
      biglietto *nuovo;
      nuovo = testa;
      testa = testa->next;
      delete nuovo;
      cout << "\n Elemento eliminato!!\n";
      getch();
     }
 else
      {
       cout << "\n Lista gia' vuota!!\n";
       getch();
       }
}//chiusura pop
```