

```

unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Grids, ComCtrls, StdCtrls, Buttons;

const
  MAXSTUD=100;
  MAXCLASSI=10;
type
  TfrmPrincipale = class(TForm)
    txtRighe: TEdit;
    sbtRighe: TUpDown;
    grdDati: TStringGrid;
    btnEsci: TBitBtn;
    btnInizio: TBitBtn;
    btnSalva: TBitBtn;
    btnLeggiDati: TBitBtn;
    grdDati2: TStringGrid;
    sbtRighe2: TUpDown;
    txtRighe2: TEdit;
    GroupBox1: TGroupBox;
    txtCognome: TEdit;
    txtNome: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    btnTrova: TBitBtn;
    lblClasse: TLabel;
    lblSede: TLabel;
    lblAula: TLabel;
    btnConferma: TBitBtn;
    procedure txtRigheChange(Sender: TObject);
    procedure btnEsciClick(Sender: TObject);
    procedure btnInizioClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure btnSalvaClick(Sender: TObject);
    procedure btnLeggiDatiClick(Sender: TObject);
    procedure txtRighe2Change(Sender: TObject);
    procedure btnConfermaClick(Sender: TObject);
    procedure btnTrovaClick(Sender: TObject);
  private
    procedure Ordina;
  { Private declarations }
  public
    { Public declarations }
  end;

  persona=record // record programma precedente
    nome: string[30];
    altezza: real;
  end;

  studente=record
    cognome: string[40];
    nome: string[60];
    classe: string[4];
  end;

  classe=record
    classe: string[4];
    sede: string[60];
    aula: integer;
  end;

var
  frmPrincipale: TfrmPrincipale;

```

Unit1.pas

```

aPersona: array[1..100] of persona;
aStudenti: array[1..MAXSTUD] of studente;
aClassi: array[1..MAXCLASSI] of classe;
flDocumento: file of persona; // file sequenziale programma precedente

```

implementation

```
{SR *.dfm}
```

```

procedure TfrmPrincipale.txtRigheChange(Sender: TObject);
begin
    grdDati.RowCount := strtoint(txtRighe.text) + 1;
end;

```

```

procedure TfrmPrincipale.btnEsciClick(Sender: TObject);
begin
    close;
end;

```

```

procedure TfrmPrincipale.Ordina();
var
    i,j: integer;
    alt: real;
    nome: string[30];
begin
    // ordinamento semplice dell'array di record secondo l'altezza.
    for i:= 1 to (grdDati.RowCount - 2) do
        for j:=i+1 to (grdDati.RowCount - 1) do
            begin
                if aPersona[i].altezza > aPersona[j].altezza then
                    begin
                        // scambio la posizione dei dati, in tutte le componenti del record.
                        nome:=aPersona[i].nome;
                        alt:=aPersona[i].altezza;
                        aPersona[i].nome:=aPersona[j].nome;
                        aPersona[i].altezza:=aPersona[j].altezza;
                        aPersona[j].nome:=nome;
                        aPersona[j].altezza:=alt;
                    end;
            end;
        end;
    end;
end;

```

```

procedure TfrmPrincipale.btnInizioClick(Sender: TObject);
var
    i: integer;
begin
    // prelevo i dati dalla tabella
    for i:=1 to (grdDati.RowCount - 1) do
        begin
            aPersona[i].nome := grdDati.Cells[0,i];
            aPersona[i].altezza := strtofloat(trim(grdDati.Cells[1,i]));
        end;
    end;

```

```

// ordina i dati caricati nell'array di record
Ordina;

```

```

// ripongo i dati ordinati nella tabella per visualizzarli
for i:=1 to (grdDati.RowCount - 1) do
    begin
        grdDati.Cells[0,i] := aPersona[i].nome;
        grdDati.Cells[1,i] := floattostr(aPersona[i].altezza);
    end;
end;

```

```

procedure TfrmPrincipale.FormShow(Sender: TObject);
begin
    grdDati.Cells[0,0] := 'Cognome';
    grdDati.Cells[1,0] := 'Nome';

```

```

grdDati.Cells[2,0] := 'Classe';

grdDati2.Cells[0,0] := 'Classe';
grdDati2.Cells[1,0] := 'Sede';
grdDati2.Cells[2,0] := 'Aula';
end;

procedure TfrmPrincipale.btnSalvaClick(Sender: TObject);
var
  i: integer;
begin
  // salvo i dati della tabella su un file
  assignfile(flDocumento, 'archivio.dat');
  // apro il file in scrittura
  rewrite(flDocumento);

  for i:=1 to (grdDati.RowCount - 1) do
  begin
    // recupero i dati presenti nella tabella
    aPersona[i].nome := grdDati.Cells[0,i];
    aPersona[i].altezza := strtofloat(trim(grdDati.Cells[1,i]));
    // scrivo ogni singolo record nel file
    write(flDocumento, aPersona[i]);
  end;
  // chiudo il file
  closefile(flDocumento);
end;

procedure TfrmPrincipale.btnLeggiDataClick(Sender: TObject);
var
  i, numrec: integer;
begin
  // salvo i dati della tabella su un file
  assignfile(flDocumento, 'archivio.dat');
  // apro il file in lettura
  reset(flDocumento);

  // recupero il numero dei record così posso dimensionare la tabella.
  numrec := filesize(flDocumento);
  // ora posso dire quante righe ha la tabella, +1 perchè c'è la riga fissa
  grdDati.RowCount := numrec + 1;

  i:=1;
  while not(eof(flDocumento)) do
  begin
    // scrivo ogni singolo record nel file
    read(flDocumento, aPersona[i]);
    // recupero i dati presenti nella tabella
    grdDati.Cells[0,i] := aPersona[i].nome;
    grdDati.Cells[1,i] := floattostr(aPersona[i].altezza);

    // incremento contatore, sono su un ciclo while
    i := i + 1;
  end;
  // chiudo il file
  closefile(flDocumento);
end;

procedure TfrmPrincipale.txtRighe2Change(Sender: TObject);
begin
  grdDati2.RowCount := strtoint(txtRighe2.text) + 1;
end;

procedure TfrmPrincipale.btnConfermaClick(Sender: TObject);
var
  i: integer;
begin
  for i:=1 to grddati.RowCount - 1 do

```

Unit1.pas

```

begin
    astudenti[i].cognome := grddati.Cells[0,i];
    astudenti[i].nome := grddati.Cells[1,i];
    astudenti[i].classe := grddati.Cells[2,i];
end;

for i:=1 to grddati2.RowCount - 1 do
begin
    aclassi[i].classe := grddati2.Cells[0,i];
    aclassi[i].sede := grddati2.Cells[1,i];
    aclassi[i].aula := strtoint(trim(grddati2.Cells[2,i]));
end;

end;

procedure TfrmPrincipale.btnTrovaClick(Sender: TObject);
var
    i: integer;
    trovato:boolean;
    nome, cognome, classe: string;
begin
    trovato:= false;
    i:=0;

    nome := txtNome.Text;
    cognome := txtCognome.Text;

    lblclasse.Caption := '';
    lblsede.Caption := '';
    lblaula.Caption := '';

    while not(trovato) and (i<=MAXSTUD) do
    begin
        if (cognome=astudenti[i].cognome) and (nome=astudenti[i].nome)
            then begin
                classe:=astudenti[i].classe;
                trovato:= true;
                lblclasse.Caption := classe;
            end;
        i:= i+1;
    end;

    if trovato
        then begin
            trovato:=false;
            i:=0;
            while not(trovato) and (i<=MAXCLASSI) do
            begin
                if classe=aclassi[i].classe
                    then begin
                        trovato:=true;
                        lblsede.Caption := aclassi[i].sede;
                        lblaula.Caption := inttostr(aclassi[i].aula);
                    end;
                i:= i+1;
            end;
        end;
    end;

end.

```