

```
program investors (input, output, investorList, updatedInvestorList);
```

```
const
```

```
    MAXCLIENTS = 10;
```

```
    NAMELENGTH = 30;
```

```
    EMAILLENGTH = 30;
```

```
type
```

```
Client = record
```

```
    firstName : array [1..NAMELENGTH] of char;
```

```
    lastName  : array [1..NAMELENGTH] of char;
```

```
    income    : real;
```

```
    email     : array [1..EMAILLENGTH] of char;
```

```
end; (* Declaration of record client *)
```

```
ClientList = array [1..MAXCLIENTS] of Client;
```

```
var
```

```
    investorList : text;
```

```
    updatedInvestorList : text;
```

```
procedure readClientInformation (var tamjClientList : ClientList ;
                                var clientCount   : integer );

var
  i : integer;
begin;
  i := 1;
  writeln;
  reset(investorList);
  writeln('Opening file "investorList" for reading');
  clientCount := 0; (* No clients exist in list yet. *)
  while NOT EOF (investorList) do
  begin
    with tamjClientList[i] do
    begin
      readln(investorList, firstName);
      readln(investorList, lastName);
      readln(investorList, income);
      readln(investorList, email);
      readln(investorList);  (* Eat the newline between client records *)
    end;
  end;
```

```
    i := i + 1;  
    clientCount := clientCount + 1;  
end;  
close(investorList);  
end; (* Read client information *)
```

```
procedure displayInstructions;  
begin  
    writeln;  
    writeln('This program allows you track a list of clients each of which is');  
    writeln('is an investor. The initial investor information will be read');  
    writeln('from the file called "investorList". From there you can display');  
    writeln('the list of clients onscreen, add a client, modify the');  
    writeln('information for a client, erase clients from the list, recorder');  
    writeln('the list of clients or search for a client. When you are done');  
    writeln('quit the program and all of your changes will be written out to');  
    writeln('a file called "updatedInvestorList"');  
end;
```

```

procedure displayClientList (tamjClientList : ClientList;
                           clientCount  : integer );

var
  i : integer;
begin
  writeln;
  writeln('Displaying client list');
  for i := 1 to clientCount do
  begin
    with tamjClientList[i] do
    begin
      writeln('First name: ':12, firstName);
      writeln('Last name: ':12, lastName);
      writeln('Income :$':12, income:0:2);
      writeln('Email: ':12, email);
    end; (*with-do *)
    writeln;
  end; (* for *)
end; (* displayClientList *)

```

```
procedure addClient (var tamjClientList : ClientList;
                    var clientCount   : integer);
var
    newClient : Client;
begin
    writeln;
    writeln('Adding new client to list of clients');
    with newClient do
    begin
        write('Enter first name of client (max 30 characters): ');
        readln(firstName);
        write('Enter last name of client (max 30 characters): ');
        readln(lastName);
        write('Enter annual gross income of client (max 8 digits)$');
        readln(income);
        write('Enter email of client (max 30 characters):');
        readln(email);
        writeln;
    end; (* with-do *)
```

```
tamjClientList[clientCount+1] := newClient;  
clientCount := clientCount + 1;  
writeln('Added new client ', newClient.lastName);  
end; (* addClient *)
```

```
procedure swap (var first : Client;  
                var second : Client);
```

```
var
```

```
    temp : Client;
```

```
begin
```

```
    temp := first;
```

```
    first:= second;
```

```
    second := temp;
```

```
end; (* swap *)
```

```

procedure reorder (var tamjClientList : ClientList;
                  clientCount  : integer);
var
  i      : integer;
  isSorted : boolean;
begin
  begin
    isSorted := True;
    for i := 1 to (clientCount-1) do
      begin
        if (tamjClientList[i].lastName > tamjClientList[i+1].lastName) then
          begin
            swap(tamjClientList[i], tamjClientList[i+1]);
            isSorted := False;
          end; (* if-then *)
        end; (* for *)
      end; (* repeat-until *)
    until (isSorted = True);
  end; (* reorder *)

```

```
procedure saveClientInformation (tamjClientList : ClientList;
                                clientCount   : integer);
var
    i : integer;
begin
    writeln;
    rewrite(updatedInvestorList);
    writeln('Saving updated investor information to file');
    writeln(""updatedInvestorList");
    for i := 1 to clientCount do
    begin
        with tamjClientList[i] do
        begin
            writeln(updatedInvestorList, firstName);
            writeln(updatedInvestorList, lastName);
            writeln(updatedInvestorList, income:0:2);
            writeln(updatedInvestorList, email);
            writeln(updatedInvestorList);
        end; (* with-do *)
    end;
```



```
end; (* for loop *)
writeln(updatedInvestorList);
close(updatedInvestorList);
end; (* saveClientInformation *)
```

```
(* Main program *)
```

```
begin
```

```
  var tamjClientList : ClientList;
```

```
  var clientCount   : integer;
```

```
  var menuSelection : char;
```

```
  writeln;
```

```
  writeln('Options');
```

```
  writeln('(D)isplay list of clients');
```

```
  writeln('(A)dd a new client');
```

```
  writeln('(M)odify an existing client');
```

```
  writeln('(E)rase a client's record from the list');
```

```
  writeln('(R)eorder the client list by ascending last name');
```

```
  writeln('(S)earch for a client');
```

```
  writeln('(Q)uit the program.');
```

```
write('Enter your option: ');  
readln(menuSelection);
```

```
case (menuSelection) of
```

```
  'D', 'd' :
```

```
  begin
```

```
    displayClientList(tamjClientList, clientCount);
```

```
  end;
```

```
'A', 'a' :
```

```
  begin
```

```
    addClient(tamjClientList, clientCount);
```

```
  end;
```

```
'M', 'm' :
```

```
  begin
```

```
    writeln;
```

```
    writeln('Modify client: You need to write the code to do this');
```

```
    writeln('in your own program.');
```

```
  end;
```

'E', 'e' :

begin

writeln;

writeln('Erase client record: You need to write the code to do');

writeln('this in your own program.');

end;

'R', 'r' :

begin

writeln;

reorder (tamjClientList, clientCount);

end;

'S', 's' :

begin

writeln;

writeln('Search for a client: You need to write the code to do');

writeln('this in your own program.');

end;

'Q', 'q' :

begin

writeln;

writeln('Thank you for using the Investor 2000 (TM) program.');

writeln('Come again!');

writeln;

end;

otherwise

begin

writeln;

writeln('Please enter one of the following options: D, A, M, E, R');

writeln('S or Q');

writeln;

end;

end; (* Case with menu options*)

end; (* repeat-until loop *)

until (menuSelection = 'Q') OR (menuSelection = 'q');

saveClientInformation(tamjClientList, clientCount);

end.