

Sorting (Bubble)

In this section of notes you will learn one technique for ordering a list

Algorithm

1st pass

E	A	D	B	A
---	---	---	---	---

2nd pass

A	D	B	A	E
---	---	---	---	---

3rd pass

A	B	A	D	E
---	---	---	---	---

4th pass

A	A	B	D	E
---	---	---	---	---

A Sorting Example

A complete version of this example program can be found under
/home/231/examples/sorting/investors.p

```
program investors (input, output);

const
  MAX_CLIENTS   = 10;
  NAME_LENGTH   = 30;
  EMAIL_LENGTH  = 30;

type
  Client = record
    firstName : array [1..NAME_LENGTH] of char;
    lastName  : array [1..NAME_LENGTH] of char;
    income    : real;
    email     : array [1..EMAIL_LENGTH] of char;
  end; (* Declaration of record client *)

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

A Sorting Example (2)

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

var
  i : integer;
begin;
  i := 1;
  writeln;
  reset(investorData, 'investorList');
  writeln('Opening file "investorList" for reading');
  clientCount := 0;
```

A Sorting Example (3)

```
while NOT EOF (investorData) do
begin
  with tamjClientList[i] do
  begin
    readln(investorData, firstName);
    readln(investorData, lastName);
    readln(investorData, income);
    readln(investorData, email);
    readln(investorData);
  end; (* End with-do: read information for a single client from a file *)
  i := i + 1;
  clientCount := clientCount + 1;
end; (* End while: reading all client information from a file *)
close(investorData);
end; (* End of procedure readClientInformation *)
```

A Sorting Example (4)

```
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;
```

A Sorting Example (5)

```
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: ', firstName);
      writeln('Last name: ', lastName);
      writeln('Income :$', income:0:2);
      writeln('Email: ', email);
    end; (*End with-do: Showing information for a single client *)
    writeln;
  end; (* End for: Showing information about all clients *)
end; (* End of procedure displayClientList *)
```

A Sorting Example (6)

```
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 with no commas)$');
    readln(income);
    write('Enter email of client (max 30 characters):');
    readln(email);
    writeln;
  end; (* End with-do: Getting information for the client *)
```

A Sorting Example (7)

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

A Sorting Example (8)

```
procedure swap (var first, second : Client);  
var  
    temp : Client;  
begin  
    temp := first;  
    first := second;  
    second := temp;  
end;
```

A Sorting Example (9)

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

var
  i      : integer;
  isSorted : boolean;
begin
  repeat
  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; (* End if-then: Check if two clients are out of order *)
    end; (* End for: Going thru entire list *)
  end; (* End repeat-until: Going thru entire list until it is in sorted order *)
  until (isSorted = True);
end; (* End of procedure reorder *)
```

The Bubble Sort in Pascal

James Tam

A Sorting Example (10)

```
procedure saveClientInformation (      tamjClientList  : ClientList;
                                     clientCount     : integer;
                                     var upDatedInvestorData : text);

var
  i : integer;
begin
  writeln;
  rewrite(updatedInvestorData,'updatedInvestorList');
  writeln('Saving updated investor information to file');
  writeln("updatedInvestorList");
```

The Bubble Sort in Pascal

James Tam

A Sorting Example (11)

```
for i := 1 to clientCount do
begin
  with tamjClientList[i] do
  begin
    writeln(updatedInvestorData, firstName);
    writeln(updatedInvestorData, lastName);
    writeln(updatedInvestorData, income:0:2);
    writeln(updatedInvestorData, email);
    writeln(updatedInvestorData);
    end; (* End with-do: Saving information for a single client to a file *)
  end; (* End for: Saving information for all clients to a file *)
  writeln(updatedInvestorData);
  close(updatedInvestorData);
end; (* End of procedure saveClientInformation *)
```

A Sorting Example (12)

```
procedure displayMenu;
begin
  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 selection: ');
end;
```

A Sorting Example (13)

```
(***** START OF THE MAIN PROGRAM *****)
begin
  (* Local variable declarations *)
  var tamjClientList      : ClientList;
  var clientCount        : integer;
  var menuSelection      : char;
  var investorData       : text;
  var updatedInvestorData : text;

  readClientInformation(tamjClientList, clientCount, investorData);
  displayInstructions;
```

A Sorting Example (14)

```
repeat
begin
  displayMenu;
  readln(menuSelection);

  case (menuSelection) of
    'D', 'd' :
      begin
        displayClientList(tamjClientList, clientCount);
      end;

    'A', 'a' :
      begin
        addClient(tamjClientList, clientCount);
      end;
```


A Sorting Example (15)

```
'M', 'm' :  
begin  
  writeln;  
  writeln('Modify client: You need to write the code to do this');  
  writeln('in your own program.');
```

```
'E', 'e' :  
begin  
  writeln;  
  writeln('Erase client record: You need to write the code to do');  
  writeln('this in your own program.');
```

```
'R', 'r' :  
begin  
  writeln;  
  reorder (tamjClientList, clientCount);  
end;
```

A Sorting Example (16)

```
'S', 's' :  
begin  
  writeln;  
  writeln('Search for a client: You need to write the code to do');  
  writeln('this in your own program.');
```

```
'Q', 'q' :  
begin  
  writeln;  
  writeln('Thank you for using the investor 2000 (TM) program.');
```

```
  writeln('Come again!');
```

```
  writeln;  
end;
```

A Sorting Example (17)

```
else
begin
  writeln;
  writeln('Please enter one of the following options: D, A, M, E, R');
  writeln('S or Q');
  writeln;
  end; (* End else *)
end; (* End Case with menu options*)
end; (* End repeat-until loop *)
until (menuSelection = 'Q') OR (menuSelection = 'q');

saveClientInformation(tamjClientList, clientCount, upDatedInvestorData);
end. (* END OF MAIN PROGRAM *)
```

You Should Now Know

How to trace through a program that employs the Bubble Sort
How to write the code for a bubble sort to reorder an array