

9608 Specimen paper 4

1c	<pre> procedure BinarySearch(Low, High : integer); var ItemFound, SearchFailed : Boolean; var Middle : integer; begin ItemFound := False; SearchFailed := False; Middle := (Low + High) DIV 2; if SearchData[Middle] = SearchItem then Found := True else if Low >= High then SearchFailed := True else if SearchData[Middle] < SearchItem then BinarySearch(Middle + 1, High) else BinarySearch(Low, Middle - 1); end; </pre>
2c	<pre> function Reject: Boolean; begin if ((G1Tests = True) and (G2Tests = False) and (G3Tests = False) or (G1Tests = False)) then Reject := True else Reject := False; end; </pre>
5b	<pre> interface type PassengerVehicle = class private regNo : String; noOfSeats : Integer; public procedure showRegNo; procedure showNoOfSeats; end; implementation procedure PassengerVehicle.showRegNo; begin </pre>

	<pre> WriteLn(regNo); end; procedure PassengerVehicle.showNoOfSeats; begin WriteLn(noOfSeats); end; end.</pre>
5c	<pre> interface type Bus = class(PassengerVehicle) private maxStanding : integer; public constructor Create(r : string; n, m : integer); procedure showMaxStanding; end; implementation constructor Bus.Create(r : string; n, m : integer); begin inherited create(r,n); maxStanding := m; end; procedure Bus.showMaxStanding; begin WriteLn(maxStanding); end; end.</pre>
5di	<pre> var pv1 : bus; pv1 := Bus.Create('NBR 123', 51, 10);</pre>
5dii	<pre> pv1.showRegNo; pv1.showNoOfSeats; pv1.showMaxStanding;</pre>