INFORMATION TECHNOLOGY: PAPER II

MARKING GUIDELINES

Time: 3 hours

These marking guidelines are prepared for use by examiners and sub-examiners, all of whom are required to attend a standardisation meeting to ensure that the guidelines are consistently interpreted and applied in the marking of candidates' scripts.

The IEB will not enter into any discussions or correspondence about any marking guidelines. It is acknowledged that there may be different views about some matters of emphasis or detail in the guidelines. It is also recognised that, without the benefit of attendance at a standardisation meeting, there may be different interpretations of the application of the marking guidelines.
SECTION A  SQL ANSWER SHEET

QUESTION 1

1.1:
SELECT Country, FullName 
FROM tblPlayers 
ORDER BY Country, Fullname  

1.2:
SELECT * 
FROM tblFinals  
WHERE FinalYear < 2000  

1.3:
DELETE * 
FROM tblPlayers  
WHERE PlayerID = 101  

1.4:
UPDATE tblTournaments  
SET PrizeMoneyZAR = PrizeMoney * 11.51  

1.5:
SELECT MONTH(DateOfBirth) AS BirthMonth, COUNT(*) AS NumBirthdays 
FROM tblPlayers  
GROUP BY MONTH(DateOfBirth)  
MYSQL: GROUP BY BirthMonth  

1.6:
SELECT Country, COUNT(*) AS TotalPlayers 
FROM tblPlayers  
GROUP BY Country 
HAVING COUNT(*) > 10  
ORDER BY COUNT(*) DESC  
MYSQL: HAVING TotalPlayers > 10  
ORDER BY TotalPlayers DESC  

1.7:
SELECT TournamentName 
FROM tblTournaments  
WHERE PrizeMoney > (SELECT AVG(PrizeMoney) FROM tblTournaments)  

1.8:

```
SELECT FullName, TournamentName, FinalYear
FROM tblFinals, tblPlayers, tblTournaments
WHERE tblFinals.TournamentID = tblTournaments.TournamentID
AND tblFinals.WinnerID = tblPlayers.PlayerID
AND FinalYear MOD 4 = 0 AND FinalYear MOD 400 <> 0 AND FinalYear > 1995

OR:

SELECT FullName, TournamentName, FinalYear
FROM tblFinals
INNER JOIN tblPlayers
ON tblFinals.WinnerID = tblPlayers.PlayerID
INNER JOIN tblTournaments
ON tblFinals.TournamentID = tblTournaments.TournamentID
WHERE FinalYear MOD 4 = 0 AND FinalYear MOD 400 <> 0 AND FinalYear > 1995

(or any solution that works for future years past 2015 and any calculation equivalent to the MOD operator)
```

1.9:

```
INSERT INTO tblFinals (TournamentID, FinalYear, FinalName, WinnerID, RunnerUpID)
SELECT 2, FinalYear, FinalName, WinnerID, RunnerUpID
FROM tblFinals
WHERE TournamentID = 1 AND FinalYear = 2015
```

[40]
SECTION B  OBJECT-ORIENTED PROGRAMMING

JAVA SOLUTION:

QUESTION 2

// Question 2.1
public class Player{
    // Question 2.2
    private int seed;
    private String fullName;
    private String country;

    // Question 2.3
    public Player(int inSeed, String inFullName, String inCountry){
        seed = inSeed;
        fullName = inFullName;
        country = inCountry;
    }

    // Question 2.4
    public String getCountryCode(){
        if (country.indexOf(" ") > -1){
            return ("" + country.charAt(0) + country.substring(country.indexOf(" ") + 1, country.indexOf(" ") + 3)).toUpperCase();
        }
        return country.substring(0, 3).toUpperCase();
    }

    // Question 2.5
    public String toString(){
        return fullName + " (" + getCountryCode() + ")";
    }
}

[14]
QUESTION 3

// Question 3.1
public class Match
{
    // Question 3.2
    private String matchCode;
    private Player player1;
    private Player player2;
    private String score;

    // Question 3.3
    public Match(String inMatchCode, Player inPlayer1, Player inPlayer2)
    {
        matchCode = inMatchCode;
        player1 = inPlayer1;
        player2 = inPlayer2;
        score = "X";
    }

    // Question 3.4
    public String getMatchCode()
    {
        return matchCode;
    }

    // Question 3.5
    public void setScore(String inScore)
    {
        score = inScore;
    }

    // Question 3.6
    public String toString()
    {
        String rString;
        rString = matchCode + " " + player1 + " vs. "+ player2;
        if (score.equals("X"))
        {
            rString = rString + " Not Yet Played";
        }
        else
        {
            rString = rString + " " + score;
        }
        return rString;
    }
}

[16]
import java.io.*;
import java.util.Scanner;

// Question 4.1
public class TournamentManager ✓
{
    // Question 4.2
    private Player[] players = new Player[32];
    private Match[] matches = new Match[16];

    // Question 4.3
    public TournamentManager(String inFileName)
    { try
        {
            Scanner scanIn = new Scanner(new FileReader(inFileName)); ✓
            for(int loop = 0; loop < 32; loop++) ✓
            {
                String line = scanIn.nextLine(); ✓
                Scanner scanTokens = new Scanner(line).useDelimiter(","); ✓
                int seed = scanTokens.nextInt(); ✓
                String playerName = scanTokens.next();
                String country = scanTokens.next();
                players[loop] ✓ = new Player ✓(seed, playerName, country ✓);
            }
            scanIn.close();
        }
        catch (FileNotFoundException fne)
        {
            System.out.println("File not found"); ✓
        }
    }

    // Question 4.4
    public String listAllPlayers() ✓
    { String rString = "";
        for(int loop = 0; loop < players.length; loop++) ✓
        { rString = rString + players[loop] ✓ + "\n" ✓;
        }
        return rString ✓;
    }

    // Question 4.5
    public String populateMatches() ✓
    { String rString = "";
        for(int loop = 0; loop < matches.length; loop++) ✓
        { int opponent = 32 - loop - 1; ✓
            matches[loop] ✓ = new Match("M", (loop+1) + (opponent+1) ✓,
            players[loop], players[opponent] ✓);
            rString = rString + matches[loop] + "\n" ✓;
        }
    }

    Attributes are private
    ✓ Sizes of both arrays are correct
// Question 4.6
public Match findMatch(String matchCode)
{
    for(int loop = 0; loop < matches.length; loop++)
    {
        if (matches[loop].getMatchCode().equals(matchCode))
        {
            return matches[loop];
        }
    }
    return null;
}

// Question 6.1
public String processResults() throws FileNotFoundException
{
    Scanner resultsIn = new Scanner(new FileReader("results.txt"));
    String matchCode = resultsIn.next();
    Match match = findMatch(matchCode);
    int[] games = new int[2];
    int[] sets = new int[2];
    String score = "";
    while (resultsIn.hasNext() && sets[0] != 3 && sets[1] != 3)
    {
        int playerIndex = resultsIn.nextInt() - 1;
        games[playerIndex]++;
        if ((games[playerIndex] >= 6) && (games[playerIndex] -
            games[(playerIndex+1) % 2] >= 2))
        {
            sets[playerIndex]++;
            score = score + games[0] + "-" + games[1] + " ";
            games[0] = games[1] = 0;
        }
        if (sets[playerIndex] == 3)
        {
            score = score + "Player " + playerIndex + " wins";
        }
    }
    resultsIn.close();
    match.setScore(score);
    return match.toString();
}
QUESTION 5 AND 6.2

// Question 5.1
public class TennisUI
{
    // Question 5.2
    private static TournamentManager tArr = new TournamentManager("players.txt");

    public static void main(String[] args) throws Exception
    {
        // Question 5.3
        System.out.println("All Players\n================\n" + tArr.listAllPlayers());
        System.out.println("\nAll Matches\n================\n" + tArr.populateMatches());

        // Question 6.2
        System.out.println("\nMatchResult : " + tArr.processResults());
    }
}

ALTERNATE JAVA SOLUTION FOR QUESTION 4.3

public TournamentManager(String inFileName)
{
    try
    {
        BufferedReader fileIn = new BufferedReader(new FileReader(inFileName));

        for(int loop = 0; loop < 32; loop++)
        {
            String line = fileIn.readLine();
            String tokens[] = line.split\(",\);
            int seed = Integer.parseInt(tokens[0]);
            String playerName = tokens[1];
            String country = tokens[2];
            players[loop] = new Player (seed, playerName, country);
        }

        fileIn.close();
    }
    catch (FileNotFoundException fne)
    {
        System.out.println("File not found");
    }
    catch (IOException fne)
    {
        System.out.println("File not found");
    }
}
**ALTERNATE JAVA SOLUTION FOR QUESTION 6.1**

**Match.java**

```java
public class Match {
    protected String matchCode;
    protected Player player1;
    protected Player player2;
    protected String score;

    public Player getPlayer1() {
        return player1;
    }

    public Player getPlayer2() {
        return player2;
    }
}
```

**PlayedMatch.java**

```java
public class PlayedMatch extends Match {
    private int[] games = new int[2];
    private int[] sets = new int[2];

    public PlayedMatch(String inMatchCode, Player inPlayer1, Player inPlayer2) {
        super(inMatchCode, inPlayer1, inPlayer2);
    }

    public void playPoint(int player) {
        games[player-1]++; // ✓
        if ((games[player-1] >= 6) && (games[player-1] - games[(player) % 2] >= 2)) { // ✓
            sets[player-1]++; // ✓
            score = score + games[0] + "-" + games[1] + " "; // ✓
            games[0] = games[1] = 0; // ✓
        }
        if (sets[player-1] == 3) { // ✓
            score = score + "Player " + player + " wins";
        }
    }

    public int getSets(int player) {
        return sets[player-1];
    }
}
```

✓ Set Score
TournamentManager.java

```java
public String processResults() throws FileNotFoundException {
    Scanner resultsIn = new Scanner(new FileReader("results.txt"));
    String matchCode = resultsIn.next();
    Match match = findMatch(matchCode);
    PlayedMatch pMatch = new PlayedMatch(match.getMatchCode(),
                                            match.getPlayer1(), match.getPlayer2());
    pMatch.setScore("");
    while (resultsIn.hasNext() && pMatch.getSets(1) != 3 &&
          pMatch.getSets(2) != 3) {
        int playerIndex = resultsIn.nextInt();
        pMatch.playPoint(playerIndex);
    }
    resultsIn.close();
    return pMatch.toString();
}
```

[13]
DELPHI SOLUTION:

QUESTION 2

unit uPlayer;

interface

uses SysUtils;

// Question 2.1
type TPlayer = class 
// Question 2.2
private
  seed : integer;
  fullname : string;
  country : string;
public
  constructor Create(inSeed : integer; inFullname, inCountry : String);
  function getCountryCode : String;
  function toString : String;
end;

implementation

{ TPlayer }

// Question 2.3
constructor TPlayer.Create(inSeed : integer; inFullname, inCountry : String);
begin
  seed := inSeed;
  fullname := inFullname;
  country := inCountry;
end;

// Question 2.4
function TPlayer.getCountryCode: String;
begin
  if (Pos(' ', country) > 0) then
  begin
    Result := UpCase(country[1]) + UpperCase(Copy(country, Pos(' ', country) + 1, 2));
  end
  else
  begin
    Result := UpperCase(Copy(country, 1, 3));
  end;
end;

// Question 2.5
function TPlayer.toString: String;
begin
  Result := fullname + ' (' + getCountryCode + ')';
end;
end.
QUESTION 3

unit uMatch;

interface

uses uPlayer;

// Question 3.1
type TMatch = class ✓
// Question 3.2
private
  matchCode : String;
  player1, player2 : TPlayer;
  score : String;
public
  constructor Create(inMatchCode: String; inPlayer1, inPlayer2 : TPlayer);
  procedure setScore(inScore : String);
  function getMatchCode : String;
  function toString : String;
end;

implementation

{ TMatch }

// Question 3.3
constructor TMatch.Create (inMatchCode: String; ✓ inPlayer1, inPlayer2: ✓ TPlayer);
begin
  matchCode := inMatchCode;
  player1 := inPlayer1;
  player2 := inPlayer2;
  score := 'X'; ✓
end;

// Question 3.4
function TMatch.getMatchCode : String; ✓
begin
  Result := matchCode;
end;

// Question 3.5
procedure TMatch.setScore(inScore: String); ✓
begin
  score := inScore;
end;

// Question 3.6
function TMatch.toString: String; ✓
begin
  Result := matchCode + ' ' + player1.toString + ' vs. ' ✓ + player2.toString ✓;
  if (score = 'X') then ✓
    begin
      Result := result + ' Not Yet Played'; ✓
    end
  else
    begin
      Result := result + ' ' + score; ✓
    end;
end;
end.

✓ matchCode and score declared correctly
✓ player1 and player2 declared as TPlayer
✓ Attribute values assigned correctly
QUESTION 4 AND 6.1

unit uTournamentManager;

interface

uses uPlayer, uMatch, SysUtils, Classes, Dialogs;

// Question 4.1
type TTournamentManager = class
private
  // Question 4.2
  players : array[1..32] of TPlayer;
  matches : array[1..16] of TMatch;
public
  constructor Create(filename : String);
  function listAllPlayers : String;
  function populateMatches : String;
  function findMatch(matchCode : String) : TMatch;
  function processResults : String;
end;

implementation

{ TTournamentManager }

// Question 4.3
constructor TTournamentManager.Create(filename: String); ✓
var
  inFile : textfile;
  curLine : String;
  playerName, country : String;
  seed : integer;
  loop : integer;
begin
  if (FileExists(filename)) then begin
    AssignFile(inFile, filename); ✓
    Reset(inFile);

    for loop := 1 to 32 do ✓
      begin
        ReadLn(infile, curLine); ✓
        ✓
        seed := StrToInt(Copy(curLine, 1, Pos(',', curLine) - 1));
        Delete(curLine, 1, Pos(',', curLine));
        playerName := Copy(curLine, 1, Pos(',', curLine) - 1);
        Delete(curLine, 1, Pos(',', curLine));
        country := curLine;
        players[loop] ✓ := TPlayer.Create(seed, playerName, country); ✓
      end;
    CloseFile(inFile);
  end
  else showMessage('File not found'); ✓
end;
// Question 4.4
function TTournamentManager.listAllPlayers: String; ✓
var
  rString : String;
  loop : integer;
begin
  rString := ' ';
  for loop := 1 to Length(players) do ✓
    begin
      rString := rString + players[loop].toString ✓ + #13 ✓;
    end;
  Result := rString; ✓
end;

// Question 4.5
function TTournamentManager.populateMatches : String; ✓
var
  loop, opponent : integer;
  rString : String;
begin
  for loop := 1 to Length(matches) do ✓
    begin
      opponent := 32 - loop + 1; ✓
      matches[loop] ✓ := TMatch.Create('M' + IntToStr(loop) +
                                         IntToStr(opponent), players[loop], players[opponent]);
      rString := rString + matches[loop].toString + #13✓;
    end;
  Result := rString;
end;

// Question 4.6
function TTournamentManager.findMatch(matchCode: String): TMatch; ✓
var
  loop : integer;
begin
  for loop := 1 to Length(matches) do ✓
    begin
      if (matches[loop].getMatchCode = matchCode) then ✓
        Result := matches[loop]; ✓
      end;
end;

// Question 6.1
function TTournamentManager.processResults: String;
var
  infile : textfile;
  matchCode, line : String;
  games, sets : array[1..2] of Integer;
  match : TMatch;
  score : String;
  playerIndex : Integer;
begin
  AssignFile(infile, 'results.txt'); ✓
  Reset(infile);
  ReadLn(infile, matchCode);
  match := findMatch(matchCode); ✓ ✓
games[1] := 0;
games[2] := 0;
sets[1] := 0;
sets[2] := 0;
score := '';

while (NOT(EOF(infile)) AND (sets[1] <> 3) AND (sets[2] <> 3)) do
  begin
    ReadLn(infile, line);
    playerIndex := StrToInt(line);
    Inc(games[playerIndex]);
    if ((games[playerIndex] >= 6) AND (games[playerIndex] - games[(playerIndex MOD 2) + 1] >= 2)) then
      begin
        Inc(sets[playerIndex]);
        score := score + IntToStr(games[1]) + '-' + IntToStr(games[2]) + ' ';
        games[1] := 0;
        games[2] := 0;
      end;
  end;
  if (sets[playerIndex] = 3) then
  begin
    score := score + 'Player ' + IntToStr(playerIndex) + ' wins';
  end;

match.setScore(score);
CloseFile(infile);
Result := match.toString;
end;

[13]

QUESTIONS 5 AND 6.2

unit uFrmTennisUI;

interface

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

type
  // Question 5.1
  TfrmTennis = class(TForm)
    rchOutput: TRichEdit;
    procedure FormActivate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
frmTennis: TfrmTennis;
  tArr : TTournamentManager;

implementation

{$R *.dfm}

procedure TfrmTennis.FormActivate(Sender: TObject);
begin
  // Question 5.2
  tArr := TTournamentManager.Create('players.txt'); ✓

  // Question 5.3
  rchOutput.Lines.Add('All Players' + #13 + '-----------' + #13);
  rchOutput.Lines.Add(tarr.listAllPlayers);
  rchOutput.Lines.Add('#13 + 'All Matches' + #13 + '-----------' + #13);
  rchOutput.Lines.Add(tarr.populateMatches);

  // Question 6.2
  rchOutput.Lines.Add('#13 + 'Match Result :' + tArr.processResults); ✓

  end;
end.
ALTERNATE DELPHI SOLUTION FOR QUESTION 6.1

uMatch.pas

unit uMatch;

protected
  matchCode : String;
  player1, player2 : TPlayer; // 3
  score : String;

... function getPlayer1 : TPlayer;
  function getPlayer2 : TPlayer;
end;

... function TMatch getPlayer1: TPlayer;
begin
  Result := Player1;
end;

function TMatch.getPlayer2: TPlayer;
begin
  Result := Player2;
end;

uPlayedMatch.pas

unit uPlayedMatch;

interface

uses uMatch, uPlayer, SysUtils;

type TPlayedMatch = class(TMatch)
private
  games : array[1..2] of Integer; ✓
  sets : array[1..2] of Integer; ✓
public
  constructor Create(inMatchCode: String; inPlayer1, inPlayer2 : TPlayer);
  override;
  procedure playPoint(p : Integer);
  function getSets(p : Integer) : Integer;
end;

implementation

{ TPlayedMatch }

constructor TPlayedMatch.Create(inMatchCode: String; inPlayer1, inPlayer2: TPlayer);
begin
  inherited Create(inMatchCode, inPlayer1, inPlayer2);
  games[1] := 0;
  games[2] := 0;
  sets[1] := 0;
  sets[2] := 0;
end;

function TPlayedMatch.getSets(p: Integer): Integer;
begin
Result := sets[p];
end;

procedure TPlayedMatch.playPoint(p: Integer);
begin
  Inc(games[p]);
  if ((games[p] >= 6) AND (games[p] - games[(p MOD 2) + 1] >= 2)) then
    begin
      Inc(sets[p]);
      score := score + IntToStr(games[1]) + '-' + IntToStr(games[2]) + ' ';
      games[1] := 0;
      games[2] := 0;
    end;
  if (sets[p] = 3) then
    begin
      score := score + 'Player ' + IntToStr(p) + ' wins';
    end;
end;
end.

uTournamentManager.pas

function TTournamentManager.processResults: String;
var
  infile : textfile;
  matchCode, line : String;
  match : TMatch;
  pMatch : TPlayedMatch;
  playerIndex : Integer;
begin
  AssignFile(infile, 'results.txt');
  Reset(infile);
  ReadLn(infile, matchCode);
  match := findMatch(matchCode);
  pMatch := TPlayedMatch.Create(matchCode, match.getPlayer1, match.getPlayer2);
  pMatch.setScore('');
  while (NOT(EOF(infile)) AND (pMatch.getSets(1) <> 3) AND (pMatch.getSets(2) <> 3)) do
    begin
      ReadLn(infile, line);
      playerIndex := StrToInt(line);
      pMatch.playPoint(playerIndex);
    end;
  CloseFile(infile);
  Result := pMatch.toString;
end;
OUTPUT

SECTION A

STRUCTURED QUERY LANGUAGE

QUESTION 1.1

<table>
<thead>
<tr>
<th>Question1_1</th>
<th>Country</th>
<th>FullName</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ARG</td>
<td>David Nalbandian</td>
</tr>
<tr>
<td></td>
<td>ARG</td>
<td>Gabriela Sabatini</td>
</tr>
<tr>
<td></td>
<td>ARG</td>
<td>Gastón Gaudio</td>
</tr>
<tr>
<td></td>
<td>ARG</td>
<td>Guillermo Coria</td>
</tr>
<tr>
<td></td>
<td>ARG</td>
<td>Juan Martín del Potro</td>
</tr>
<tr>
<td></td>
<td>ARG</td>
<td>Mariano Puerta</td>
</tr>
<tr>
<td></td>
<td>AUS</td>
<td>Lleyton Hewitt</td>
</tr>
<tr>
<td></td>
<td>AUS</td>
<td>Mark Philippoussis</td>
</tr>
<tr>
<td></td>
<td>AUS</td>
<td>Patrick Rafter</td>
</tr>
<tr>
<td></td>
<td>AUS</td>
<td>Samantha Stosur</td>
</tr>
<tr>
<td></td>
<td>AUT</td>
<td>Thomas Muster</td>
</tr>
<tr>
<td></td>
<td>BEL</td>
<td>Justine Henin</td>
</tr>
<tr>
<td></td>
<td>BEL</td>
<td>Kim Clijsters</td>
</tr>
<tr>
<td></td>
<td>BLR</td>
<td>Victoria Azarenka</td>
</tr>
<tr>
<td></td>
<td>BRA</td>
<td>Gustavo Kuerten</td>
</tr>
<tr>
<td></td>
<td>CAN</td>
<td>Eugenie Bouchard</td>
</tr>
<tr>
<td></td>
<td>CHI</td>
<td>Fernando González</td>
</tr>
<tr>
<td></td>
<td>CHI</td>
<td>Marcelo Ríos</td>
</tr>
<tr>
<td></td>
<td>CHN</td>
<td>Li Na</td>
</tr>
<tr>
<td></td>
<td>CRO</td>
<td>Goran Ivanišević</td>
</tr>
<tr>
<td></td>
<td>CRO</td>
<td>Iva Majoli</td>
</tr>
<tr>
<td></td>
<td>CRO</td>
<td>Marín Cilic</td>
</tr>
<tr>
<td></td>
<td>CYP</td>
<td>Marcos Baghdatis</td>
</tr>
<tr>
<td></td>
<td>CZE</td>
<td>Jana Novotná</td>
</tr>
<tr>
<td></td>
<td>CZE</td>
<td>Petr Korda</td>
</tr>
<tr>
<td></td>
<td>CZE</td>
<td>Petra Kvitová</td>
</tr>
<tr>
<td></td>
<td>CZE</td>
<td>Tomáš Berdych</td>
</tr>
<tr>
<td></td>
<td>DEN</td>
<td>Caroline Wozniacki</td>
</tr>
<tr>
<td></td>
<td>ECU</td>
<td>Andrés Gómez</td>
</tr>
<tr>
<td></td>
<td>ESP</td>
<td>Albert Costa</td>
</tr>
<tr>
<td></td>
<td>ESP</td>
<td>Alberto Berasategui</td>
</tr>
<tr>
<td></td>
<td>ESP</td>
<td>Àlex Corretja</td>
</tr>
<tr>
<td></td>
<td>ESP</td>
<td>Arantxa Sánchez Vicario</td>
</tr>
<tr>
<td></td>
<td>ESP</td>
<td>Carlos Moyá</td>
</tr>
<tr>
<td></td>
<td>ESP</td>
<td>Carlos Moyà</td>
</tr>
<tr>
<td></td>
<td>ESP</td>
<td>Conchita Martínez</td>
</tr>
<tr>
<td></td>
<td>ESP</td>
<td>David Ferrer</td>
</tr>
<tr>
<td></td>
<td>ESP</td>
<td>Juan Carlos Ferrero</td>
</tr>
<tr>
<td></td>
<td>ESP</td>
<td>Rafael Nadal</td>
</tr>
<tr>
<td></td>
<td>ESP</td>
<td>Sergi Bruguera</td>
</tr>
<tr>
<td></td>
<td>FRA</td>
<td>Amélie Mauresmo</td>
</tr>
<tr>
<td></td>
<td>FRA</td>
<td>Arnaud Clément</td>
</tr>
<tr>
<td></td>
<td>FRA</td>
<td>Cédric Pioline</td>
</tr>
<tr>
<td></td>
<td>FRA</td>
<td>Jo-Wilfried Tsonga</td>
</tr>
<tr>
<td></td>
<td>FRA</td>
<td>Marion Bartoli</td>
</tr>
<tr>
<td></td>
<td>FRA</td>
<td>Mary Pierce</td>
</tr>
<tr>
<td></td>
<td>FRA</td>
<td>Nathalie Tausiat</td>
</tr>
<tr>
<td></td>
<td>FRA</td>
<td>Steffi Graf</td>
</tr>
<tr>
<td></td>
<td>GBR</td>
<td>Andy Murray</td>
</tr>
<tr>
<td></td>
<td>GBR</td>
<td>Greg Rusedski</td>
</tr>
<tr>
<td></td>
<td>GER</td>
<td>Anke Huber</td>
</tr>
<tr>
<td></td>
<td>GER</td>
<td>Boris Becker</td>
</tr>
<tr>
<td></td>
<td>GER</td>
<td>Michael Stich</td>
</tr>
<tr>
<td></td>
<td>GER</td>
<td>Rainer Schüttler</td>
</tr>
<tr>
<td></td>
<td>GER</td>
<td>Sabine Lisicki</td>
</tr>
<tr>
<td></td>
<td>ITA</td>
<td>Francesca Schiavone</td>
</tr>
<tr>
<td></td>
<td>ITA</td>
<td>Sara Errani</td>
</tr>
<tr>
<td></td>
<td>JPN</td>
<td>Kei Nishikori</td>
</tr>
<tr>
<td></td>
<td>NED</td>
<td>Martin Verkerk</td>
</tr>
<tr>
<td></td>
<td>NED</td>
<td>Richard Krajicek</td>
</tr>
<tr>
<td></td>
<td>POL</td>
<td>Agnieszka Radwańska</td>
</tr>
<tr>
<td></td>
<td>ROU</td>
<td>Simona Halep</td>
</tr>
<tr>
<td></td>
<td>RUS</td>
<td>Anastasia Myskina</td>
</tr>
<tr>
<td></td>
<td>RUS</td>
<td>Dinara Safina</td>
</tr>
<tr>
<td></td>
<td>RUS</td>
<td>Elena Dementieva</td>
</tr>
<tr>
<td></td>
<td>RUS</td>
<td>Marat Safin</td>
</tr>
<tr>
<td></td>
<td>RUS</td>
<td>Maria Sharapova</td>
</tr>
<tr>
<td></td>
<td>RUS</td>
<td>Svetlana Kuznetsova</td>
</tr>
<tr>
<td></td>
<td>RUS</td>
<td>Vera Zvonareva</td>
</tr>
<tr>
<td></td>
<td>RUS</td>
<td>Yevgeny Kafelnikov</td>
</tr>
<tr>
<td></td>
<td>SER</td>
<td>Jelena Janković</td>
</tr>
<tr>
<td></td>
<td>SRB</td>
<td>Ana Ivanovic</td>
</tr>
<tr>
<td></td>
<td>SRB</td>
<td>Novak Djokovic</td>
</tr>
<tr>
<td></td>
<td>SUI</td>
<td>Martina Hingis</td>
</tr>
<tr>
<td></td>
<td>SUI</td>
<td>Roger Federer</td>
</tr>
<tr>
<td></td>
<td>SUI</td>
<td>Stanislas Wawrinka</td>
</tr>
<tr>
<td></td>
<td>SVK</td>
<td>Dominika Cibulková</td>
</tr>
<tr>
<td></td>
<td>SWE</td>
<td>Magnus Norman</td>
</tr>
<tr>
<td></td>
<td>SWE</td>
<td>Robin Söderling</td>
</tr>
</tbody>
</table>
### Question 1.1

<table>
<thead>
<tr>
<th>Country</th>
<th>FullName</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWE</td>
<td>Stefan Edberg</td>
</tr>
<tr>
<td>SWE</td>
<td>Thomas Enqvist</td>
</tr>
<tr>
<td>SWE</td>
<td>Thomas Johansson</td>
</tr>
<tr>
<td>TCH</td>
<td>Helena Suková</td>
</tr>
<tr>
<td>TCH</td>
<td>Ivan Lendl</td>
</tr>
<tr>
<td>UYK</td>
<td>Andrei Medvedev</td>
</tr>
<tr>
<td>USA</td>
<td>Andre Agassi</td>
</tr>
<tr>
<td>USA</td>
<td>Andy Roddick</td>
</tr>
<tr>
<td>USA</td>
<td>Jennifer Capriati</td>
</tr>
<tr>
<td>USA</td>
<td>Jim Courier</td>
</tr>
<tr>
<td>USA</td>
<td>Lindsay Davenport</td>
</tr>
</tbody>
</table>

### Question 1.2

<table>
<thead>
<tr>
<th>TournamentID</th>
<th>FinalYear</th>
<th>FinalName</th>
<th>WinnerID</th>
<th>RunnerUpID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1990</td>
<td>Women's Singles</td>
<td>16</td>
<td>70</td>
</tr>
<tr>
<td>1</td>
<td>1990</td>
<td>Men's Singles</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1990</td>
<td>Men's Singles</td>
<td>30</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>1990</td>
<td>Women's Singles</td>
<td>17</td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td>1990</td>
<td>Women's Singles</td>
<td>50</td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td>1990</td>
<td>Men's Singles</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>1990</td>
<td>Women's Singles</td>
<td>15</td>
<td>96</td>
</tr>
<tr>
<td>4</td>
<td>1990</td>
<td>Men's Singles</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>1991</td>
<td>Men's Singles</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1991</td>
<td>Women's Singles</td>
<td>17</td>
<td>57</td>
</tr>
<tr>
<td>2</td>
<td>1991</td>
<td>Men's Singles</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>1991</td>
<td>Women's Singles</td>
<td>17</td>
<td>38</td>
</tr>
<tr>
<td>3</td>
<td>1991</td>
<td>Men's Singles</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>1991</td>
<td>Women's Singles</td>
<td>17</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>1991</td>
<td>Men's Singles</td>
<td>53</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>1991</td>
<td>Women's Singles</td>
<td>16</td>
<td>50</td>
</tr>
<tr>
<td>1</td>
<td>1992</td>
<td>Women's Singles</td>
<td>17</td>
<td>70</td>
</tr>
<tr>
<td>1</td>
<td>1992</td>
<td>Men's Singles</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1992</td>
<td>Men's Singles</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>1992</td>
<td>Women's Singles</td>
<td>17</td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td>1992</td>
<td>Men's Singles</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>1992</td>
<td>Women's Singles</td>
<td>17</td>
<td>38</td>
</tr>
<tr>
<td>4</td>
<td>1992</td>
<td>Men's Singles</td>
<td>6</td>
<td>55</td>
</tr>
<tr>
<td>4</td>
<td>1992</td>
<td>Women's Singles</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>1</td>
<td>1993</td>
<td>Men's Singles</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1993</td>
<td>Women's Singles</td>
<td>17</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>1993</td>
<td>Women's Singles</td>
<td>16</td>
<td>70</td>
</tr>
<tr>
<td>2</td>
<td>1993</td>
<td>Men's Singles</td>
<td>31</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>1993</td>
<td>Women's Singles</td>
<td>16</td>
<td>69</td>
</tr>
<tr>
<td>TournamentID</td>
<td>FinalYear</td>
<td>FinalName</td>
<td>WinnerID</td>
<td>RunnerUpID</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>----------------</td>
<td>----------</td>
<td>------------</td>
</tr>
<tr>
<td>3</td>
<td>1993</td>
<td>Men's Singles</td>
<td>5</td>
<td>86</td>
</tr>
<tr>
<td>4</td>
<td>1993</td>
<td>Men's Singles</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>1993</td>
<td>Women's Singles</td>
<td>16</td>
<td>57</td>
</tr>
<tr>
<td>1</td>
<td>1994</td>
<td>Men's Singles</td>
<td>5</td>
<td>60</td>
</tr>
<tr>
<td>1</td>
<td>1994</td>
<td>Women's Singles</td>
<td>16</td>
<td>38</td>
</tr>
<tr>
<td>2</td>
<td>1994</td>
<td>Women's Singles</td>
<td>38</td>
<td>18</td>
</tr>
<tr>
<td>2</td>
<td>1994</td>
<td>Men's Singles</td>
<td>31</td>
<td>74</td>
</tr>
<tr>
<td>3</td>
<td>1994</td>
<td>Women's Singles</td>
<td>38</td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td>1994</td>
<td>Men's Singles</td>
<td>6</td>
<td>53</td>
</tr>
<tr>
<td>4</td>
<td>1994</td>
<td>Women's Singles</td>
<td>5</td>
<td>55</td>
</tr>
<tr>
<td>4</td>
<td>1994</td>
<td>Women's Singles</td>
<td>56</td>
<td>15</td>
</tr>
<tr>
<td>1</td>
<td>1995</td>
<td>Men's Singles</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>1995</td>
<td>Women's Singles</td>
<td>18</td>
<td>38</td>
</tr>
<tr>
<td>2</td>
<td>1995</td>
<td>Women's Singles</td>
<td>16</td>
<td>38</td>
</tr>
<tr>
<td>2</td>
<td>1995</td>
<td>Men's Singles</td>
<td>32</td>
<td>29</td>
</tr>
<tr>
<td>3</td>
<td>1995</td>
<td>Women's Singles</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>1995</td>
<td>Men's Singles</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>1995</td>
<td>Men's Singles</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>1995</td>
<td>Women's Singles</td>
<td>16</td>
<td>38</td>
</tr>
<tr>
<td>1</td>
<td>1996</td>
<td>Women's Singles</td>
<td>17</td>
<td>71</td>
</tr>
<tr>
<td>1</td>
<td>1996</td>
<td>Men's Singles</td>
<td>3</td>
<td>29</td>
</tr>
<tr>
<td>2</td>
<td>1996</td>
<td>Men's Singles</td>
<td>8</td>
<td>53</td>
</tr>
<tr>
<td>2</td>
<td>1996</td>
<td>Women's Singles</td>
<td>16</td>
<td>38</td>
</tr>
<tr>
<td>3</td>
<td>1996</td>
<td>Women's Singles</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>1996</td>
<td>Men's Singles</td>
<td>5</td>
<td>29</td>
</tr>
<tr>
<td>4</td>
<td>1996</td>
<td>Men's Singles</td>
<td>54</td>
<td>93</td>
</tr>
<tr>
<td>4</td>
<td>1996</td>
<td>Women's Singles</td>
<td>16</td>
<td>38</td>
</tr>
<tr>
<td>1</td>
<td>1997</td>
<td>Women's Singles</td>
<td>19</td>
<td>18</td>
</tr>
<tr>
<td>1</td>
<td>1997</td>
<td>Men's Singles</td>
<td>5</td>
<td>61</td>
</tr>
<tr>
<td>2</td>
<td>1997</td>
<td>Women's Singles</td>
<td>39</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>1997</td>
<td>Men's Singles</td>
<td>33</td>
<td>31</td>
</tr>
<tr>
<td>3</td>
<td>1997</td>
<td>Men's Singles</td>
<td>44</td>
<td>87</td>
</tr>
<tr>
<td>3</td>
<td>1997</td>
<td>Women's Singles</td>
<td>19</td>
<td>51</td>
</tr>
<tr>
<td>4</td>
<td>1997</td>
<td>Men's Singles</td>
<td>5</td>
<td>86</td>
</tr>
<tr>
<td>4</td>
<td>1997</td>
<td>Women's Singles</td>
<td>19</td>
<td>57</td>
</tr>
<tr>
<td>1</td>
<td>1998</td>
<td>Men's Singles</td>
<td>7</td>
<td>62</td>
</tr>
<tr>
<td>1</td>
<td>1998</td>
<td>Women's Singles</td>
<td>19</td>
<td>56</td>
</tr>
<tr>
<td>2</td>
<td>1998</td>
<td>Men's Singles</td>
<td>34</td>
<td>75</td>
</tr>
<tr>
<td>2</td>
<td>1998</td>
<td>Women's Singles</td>
<td>38</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>1998</td>
<td>Women's Singles</td>
<td>20</td>
<td>19</td>
</tr>
<tr>
<td>3</td>
<td>1998</td>
<td>Men's Singles</td>
<td>44</td>
<td>88</td>
</tr>
<tr>
<td>4</td>
<td>1998</td>
<td>Men's Singles</td>
<td>5</td>
<td>55</td>
</tr>
<tr>
<td>4</td>
<td>1998</td>
<td>Women's Singles</td>
<td>57</td>
<td>97</td>
</tr>
<tr>
<td>1</td>
<td>1999</td>
<td>Men's Singles</td>
<td>8</td>
<td>63</td>
</tr>
<tr>
<td>1</td>
<td>1999</td>
<td>Women's Singles</td>
<td>19</td>
<td>24</td>
</tr>
</tbody>
</table>
**Question 1.2**

<table>
<thead>
<tr>
<th>TournamentID</th>
<th>FinalYear</th>
<th>FinalName</th>
<th>WinnerID</th>
<th>RunnerUpID</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1999</td>
<td>Women's Singles</td>
<td>16</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>1999</td>
<td>Men's Singles</td>
<td>6</td>
<td>76</td>
</tr>
<tr>
<td>3</td>
<td>1999</td>
<td>Women's Singles</td>
<td>22</td>
<td>19</td>
</tr>
<tr>
<td>3</td>
<td>1999</td>
<td>Men's Singles</td>
<td>6</td>
<td>60</td>
</tr>
<tr>
<td>4</td>
<td>1999</td>
<td>Men's Singles</td>
<td>5</td>
<td>60</td>
</tr>
<tr>
<td>4</td>
<td>1999</td>
<td>Women's Singles</td>
<td>20</td>
<td>16</td>
</tr>
</tbody>
</table>

**Question 1.3**

*(No output)*

**Question 1.4**

*(No output)*

**Question 1.5**

<table>
<thead>
<tr>
<th>BirthMonth</th>
<th>NumBirthdays</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>13</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td>9</td>
</tr>
</tbody>
</table>

**Question 1.6**

<table>
<thead>
<tr>
<th>Country</th>
<th>TotalPlayers</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>14</td>
</tr>
<tr>
<td>ESP</td>
<td>11</td>
</tr>
</tbody>
</table>

**Question 1.7**

<table>
<thead>
<tr>
<th>TournamentName</th>
</tr>
</thead>
<tbody>
<tr>
<td>Australian Open</td>
</tr>
<tr>
<td>Wimbledon</td>
</tr>
</tbody>
</table>
### QUESTION 1.8

<table>
<thead>
<tr>
<th>FullName</th>
<th>TournamentName</th>
<th>FinalYear</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monica Seles</td>
<td>Australian Open</td>
<td>1996</td>
</tr>
<tr>
<td>Boris Becker</td>
<td>Australian Open</td>
<td>1996</td>
</tr>
<tr>
<td>Yevgeny Kafelnikov</td>
<td>French Open</td>
<td>1996</td>
</tr>
<tr>
<td>Steffi Graf</td>
<td>French Open</td>
<td>1996</td>
</tr>
<tr>
<td>Steffi Graf</td>
<td>Wimbledon</td>
<td>1996</td>
</tr>
<tr>
<td>Pete Sampras</td>
<td>Wimbledon</td>
<td>1996</td>
</tr>
<tr>
<td>Richard Krajicek</td>
<td>US Open</td>
<td>1996</td>
</tr>
<tr>
<td>Steffi Graf</td>
<td>US Open</td>
<td>1996</td>
</tr>
<tr>
<td>Justine Henin</td>
<td>Australian Open</td>
<td>2004</td>
</tr>
<tr>
<td>Roger Federer</td>
<td>Australian Open</td>
<td>2004</td>
</tr>
<tr>
<td>Anastasia Myskina</td>
<td>French Open</td>
<td>2004</td>
</tr>
<tr>
<td>Gastón Gaudio</td>
<td>French Open</td>
<td>2004</td>
</tr>
<tr>
<td>Svetlana Kuznetsova</td>
<td>Wimbledon</td>
<td>2004</td>
</tr>
<tr>
<td>Roger Federer</td>
<td>Wimbledon</td>
<td>2004</td>
</tr>
<tr>
<td>Roger Federer</td>
<td>US Open</td>
<td>2004</td>
</tr>
<tr>
<td>Maria Sharapova</td>
<td>US Open</td>
<td>2004</td>
</tr>
<tr>
<td>Maria Sharapova</td>
<td>Australian Open</td>
<td>2008</td>
</tr>
<tr>
<td>Novak Djokovic</td>
<td>Australian Open</td>
<td>2008</td>
</tr>
<tr>
<td>Ana Ivanovic</td>
<td>French Open</td>
<td>2008</td>
</tr>
<tr>
<td>Rafael Nadal</td>
<td>French Open</td>
<td>2008</td>
</tr>
<tr>
<td>Serena Williams</td>
<td>Wimbledon</td>
<td>2008</td>
</tr>
<tr>
<td>Roger Federer</td>
<td>Wimbledon</td>
<td>2008</td>
</tr>
<tr>
<td>Venus Williams</td>
<td>US Open</td>
<td>2008</td>
</tr>
<tr>
<td>Rafael Nadal</td>
<td>US Open</td>
<td>2008</td>
</tr>
<tr>
<td>Victoria Azarenka</td>
<td>Australian Open</td>
<td>2012</td>
</tr>
<tr>
<td>Novak Djokovic</td>
<td>Australian Open</td>
<td>2012</td>
</tr>
<tr>
<td>Rafael Nadal</td>
<td>French Open</td>
<td>2012</td>
</tr>
<tr>
<td>Maria Sharapova</td>
<td>French Open</td>
<td>2012</td>
</tr>
<tr>
<td>Andy Murray</td>
<td>Wimbledon</td>
<td>2012</td>
</tr>
<tr>
<td>Serena Williams</td>
<td>Wimbledon</td>
<td>2012</td>
</tr>
<tr>
<td>Serena Williams</td>
<td>US Open</td>
<td>2012</td>
</tr>
<tr>
<td>Roger Federer</td>
<td>US Open</td>
<td>2012</td>
</tr>
</tbody>
</table>

### QUESTION 1.9

(No output)
SECTION B  OBJECT-ORIENTED PROGRAMMING

FINAL OUTPUT:

All Players
---------------
Colby Noble (HKO)
Craig Hamilton (MYA)
Dante Franklin (SKO)
Chadwick Baird (IND)
Wallace Forbes (CAM)
Tanek Jimenez (ITA)
Jared Flynn (BUR)
Yasir Rosario (TAI)
Levi Klein (BOT)
Kibo Ford (ASA)
Lamar Carney (CAF)
Dieter Reese (BRU)
Brady Serrano (RWA)
Theodore Snyder (TNE)
Flynn Flynn (MOL)
Alec Sellers (UST)
Oscar Price (THA)
Neville Ball (LES)
Nero Barr (GER)
Yardley Craig (UST)
Daquan Morales (SPA)
Stuart Barrett (MIS)
Hector Hinton (DJI)
Dillon Maddox (LIB)
Nash Powers (AND)
Harper Hubbard (HKO)
Stuart Williams (GIB)
Barrett Roberson (PER)
Knox Olson (SOM)
Kasper Greene (SCO)
Emery Kramer (SAF)
Jason Underwood (POR)

All Matches
--------------
M132 Colby Noble (HKO) vs. Jason Underwood (POR) Not Yet Played
M231 Craig Hamilton (MYA) vs. Emery Kramer (SAF) Not Yet Played
M330 Dante Franklin (SKO) vs. Kasper Greene (SCO) Not Yet Played
M429 Chadwick Baird (IND) vs. Knox Olson (SOM) Not Yet Played
M528 Wallace Forbes (CAM) vs. Barrett Roberson (PER) Not Yet Played
M627 Tanek Jimenez (ITA) vs. Stuart Williams (GIB) Not Yet Played
M726 Jared Flynn (BUR) vs. Harper Hubbard (HKO) Not Yet Played
M825 Yasir Rosario (TAI) vs. Nash Powers (AND) Not Yet Played
M924 Levi Klein (BOT) vs. Dillon Maddox (LIB) Not Yet Played
M1023 Kibo Ford (ASA) vs. Hector Hinton (DJI) Not Yet Played
M1122 Lamar Carney (CAF) vs. Stuart Barrett (MIS) Not Yet Played
M1221 Dieter Reese (BRU) vs. Daquan Morales (SPA) Not Yet Played
M1320 Brady Serrano (RWA) vs. Yardley Craig (UST) Not Yet Played
M1419 Theodore Snyder (TNE) vs. Nero Barr (GER) Not Yet Played
M1518 Flynn Flynn (MOL) vs. Neville Ball (LES) Not Yet Played
M1617 Alec Sellers (UST) vs. Oscar Price (THA) Not Yet Played

Match Result :M429 Chadwick Baird (IND) vs. Knox Olson (SOM) 6-0 6-8 6-4 6-3 Player 1 wins