宮城県将棋連盟からの依頼

対戦表を作成するプログラム　仕様

1. どこで使うか
   1. 大会の組み分けに使用する
   2. 参加者は、10～30人に仮定する
2. 現在の利用方法
   1. 参加者は大会会場に到着
   2. 参加者は参加登録を行う
      1. 名前、くじ番号を申請
      2. 遠方から一緒に来たなど、「できるだけ対戦相手にならないようにしてほしい」人たち（チーム）を申請
   3. 参加者が奇数の場合、「不戦敗」となる参加者を作成する
   4. 主催者は1回目の対戦表を作成する
      1. 同じチームの人とは、できるだけ対戦しないようにする
      2. くじの番号の若いほうから順番に対戦表を作成する
   5. 参加者は対戦し、主催者に結果を報告する
      1. 主催者は、勝敗を記録する
   6. 主催者は2回目の対戦表を作成する
      1. 最優先：今日既に対戦している人と対戦しないこと
      2. 優先：同じ勝数の人と対戦すること。同じ勝ち数の人が奇数の場合、勝ち数が1少ない人を1人入れる。
      3. できるだけ：同じチームの人と対戦しないようにしたいが、仕方なければ対戦してもよい
   7. （オ）と（カ）を繰り返す
   8. 点数計算
      1. それぞれの参加者が、何勝したか計算する
      2. それぞれの参加者の、それぞれの試合ごとに、勝った場合は相手の勝数を加算する（強い相手に勝った人の点数を高くする）

C#でプログラムを作成する

1. 提供されるクラスについて
   1. Personクラス
      1. 参加者について登録するクラス
         1. LotNumber・・・くじの番号
         2. PersonGroup・・・グループ（アルファベット大文字1文字）
         3. PersonName・・・参加者名
   2. Matchクラス
      1. 対戦するPersonのくじ番号と、勝敗（または結果が出ていない）を記録
      2. コンストラクタ（完成）
         1. くじ番号を２つ渡す。Result=0（結果が出ていない）を設定する。
      3. MakeMatchメソッド（未完成）
         1. 第1引数：くじ番号をKeyに、PersonをValueに設定したDictionary
         2. 第2引数：今までの対戦が格納されたMatchの配列
         3. 戻り値：対戦の一覧
      4. MergeMatchメソッド（未完成）
         1. 第1引数：今までの対戦が格納されたMatchの配列
         2. 第2引数：今回の対戦が格納されたMatchの配列
         3. 戻り値：第1引数に第2引数を結合させたMatchの配列
   3. Scoreクラス（未定義）
      1. くじ番号と、勝数、最終得点を記録
      2. CalcWinCountメソッド（未定義）
         1. 第1引数：最終戦の結果まで含めたMatchの配列
         2. 戻り値：くじ番号をKeyに、ScoreのインスタンスをValueに設定したDictionary<int,Score>　※最終得点は0に設定する
      3. CalcScoreメソッド（未定義）
         1. 第1引数：最終戦の結果まで含めたMatchの配列
         2. 戻り値：くじ番号をKeyに、ScoreのインスタンスをValueに設定したDictionary<int,Score>　※最終得点も格納する

それ以外に、メソッドの中で必要な処理があれば、それぞれ作成し、テストを追加して構わない。