```
// Name - Isala Piyarisi
// IIT ID - 2018421
// UOW ID - w1742118

BEGIN
SET scores;
SET scoresArchive;
SET lastScores;
SET lastScoresArchive;
SET restoredSnapshot = false;
SET option;
OUTPUT Welcome to Springfield Golf Club.
REPEAT
   showMenu();
   INPUT option
   OUTPUT >
   WHILE option is not integer:
      OUTPUT ERROR 406: Invalid Input \nSelect *Only* one of these Option:
      showMenu();
      OUTPUT >
      INPUT option

   SWITCH(OPTION){
      case 1:
         enterScore();
         break;

      case 2:
         findGolfer();
         break;

      case 3:
         displayScoreboard();
         break;

      case 4:
         removeGolfer();
         break;

      case 5:
         restoreGolfer();
         break;

      case 6:
         restoreLastState();
         break;
```

```
        case 7:
            QUIT THE PROGRAM

        default:
            OUTPUT ERROR 406: Invalid Input
            showMenu();
    }

UNTIL option >= 1 && option <= 7




FUNCTION showMenu(){
        OUTPUT Select one of these options:
        OUTPUT 1) Enter Scores
        OUTPUT 2) Find Golfer
        OUTPUT 3) Display Scoreboard
        OUTPUT 4) Remove Golfer
        OUTPUT 5) Restore a removed golfer
        if(isRestoredSnapshot()){
            OUTPUT 6) Redo last action
        }
        else{
            OUTPUT 6) Undo last action
        }
        OUTPUT 7) Exit Program
}

FUNCTION enterScore(){
    SET newScores;
    OUTPUT How many golfers in the group:
    INPUT n_golfers
    WHILE n_golfers is not integer:
        OUTPUT ERROR 406: Invalid Input
        OUTPUT How many golfers in the group:
        INPUT n_golfers
    FOR i IN i FROM n_golfers:
        OUTPUT GOLFER {i}
        OUTPUT Name:
        INPUT name
}
```

```
FUNCTION getScoreFromUser(){
   OUTPUT Result:
   INPUT score
   WHILE score is not integer:
      OUTPUT ERROR 400: Invalid Result,  Make sure your result is between 18 and 108, Enter Again !
      OUTPUT Result:
      INPUT score
   IF score < 18 || score > 108:
      OUTPUT ERROR 400: Invalid Result,  Make sure your result is between 18 and 108, Enter Again !
      score = getScoreFromUser();
   return score
}

FUNCTION findGolfer(){
   OUTPUT Name:
   INPUT name
   IF (checkGolfer(name)) {
      OUTPUT {name} {getScores(name))};
   }
   ELSE{
      OUTPUT ERROR 404: Player not found"
   }
}

FUNCTION removeGolfer() {
    OUTPUT Name:
    INPUT name
    IF (checkGolfer(name)) {
       setScoresArchive(name, getScores(name));
       ADD name TO Scores
       OUTPUT {name} was removed from the data structure
    }
    ELSE {
       OUTPUT ERROR 404: Player not found
    }
}

FUNCTION restoreGolfer(){
   OUTPUT Name:
   INPUT name
   IF (checkGolferArchive(name)) {
      setScores(name, getScoresArchive(name));
      ADD name TO scoresArchive
      OUTPUT {name} was restored to the data structure
   } ELSE {
      OUTPUT ERROR 404: Player not found
    }
}
FUNCTION displayScoreboard(){
    SORT scores
    FOR score in scores:
        OUTPUT {score.key} {score.value}
}

FUNCTION restoreLastState(){
    SET currentScores = scores
    SET currentScoresArchive = scoresArchive
    if(isRestoredSnapshot()) {
        OUTPUT Redoing the last action
        SET restoredSnapshot = false;
    }
    else {
        OUTPUT Undoing the last action
        SET restoredSnapshot = true;
    }
    scores = lastScores
    scoresArchive = lastScoresArchive
    lastScores = currentScores
    lastScoresArchive = currentScoresArchive
}

FUNCTION setScores(name, score){
    snapshotCurrentState();
    ADD score AND name TO scores MAP
}

FUNCTION setMultipleScores(newScores){
    snapshotCurrentState();
    CONCATENATE newScores AND score
}

FUNCTION getScores(name){
    RETURN GET score FROM scores MAP WHERE name IS EQUAL TO KEY
}

FUNCTION checkGolfer(name){
    RETURN WEATHER name IN score MAP
}

FUNCTION setScoresArchive(name, score){
    snapshotCurrentState();
    ADD scoresArchive AND name TO scores MAP
}

FUNCTION getScoresArchive(name){
    RETURN GET scoresArchive FROM scores MAP WHERE name IS EQUAL TO KEY
}
FUNCTION checkGolferArchive(name){
    RETURN WEATHER name IN scoresArchive MAP
}

FUNCTION snapshotCurrentState(){
    lastScores = scores
    lastScoresArchive = scoresArchive
}

FUNCTION isRestoredSnapshot(){
    return restoredSnapshot IS TRUE OR FALSE;
}
```