#include<bits/stdc++.h>

using namespace std;

class Player {

public:

Player();

std::string name;

int id;

int runsScored;

int ballsPlayed;

int ballsBowled;

int runsGiven;

int wicketsTaken;

};

Player::Player() {

runsScored = 0;

ballsPlayed = 0;

ballsBowled = 0;

runsGiven = 0;

wicketsTaken = 0;

}

class Team {

public:

Team();

std::string name;

int totalRunsScored;

int wicketsLost;

int totalBallsBowled;

std::vector<Player> players;

};

Team::Team() {

totalRunsScored = 0;

wicketsLost = 0;

totalBallsBowled = 0;

}

class Game {

public:

Game();

int playersPerTeam;

int maxBalls;

int totalPlayers;

std::string players[11];

bool isFirstInnings;

Team teamA, teamB;

Team \*battingTeam, \*bowlingTeam;

Player \*batsman, \*bowler;

void welcome();

void showAllPlayers();

int takeIntegerInput();

void selectPlayers();

bool validateSelectedPlayer(int);

void showTeamPlayers();

void toss();

void tossChoice(Team);

void startFirstInnings();

void initializePlayers();

void playInnings();

void bat();

bool validateInningsScore();

void showGameScorecard();

void startSecondInnings();

void showMatchSummary();

};

Game::Game() {

playersPerTeam = 4;

maxBalls = 6;

totalPlayers = 11;

players[0] = "Virat";

players[1] = "Rohit";

players[2] = "Dhawan";

players[3] = "Pant";

players[4] = "Karthik";

players[5] = "KLRahul";

players[6] = "Jadeja";

players[7] = "Hardik";

players[8] = "Bumrah";

players[9] = "BKumar";

players[10] = "Ishant";

isFirstInnings = false;

teamA.name = "Team-A";

teamB.name = "Team-B";

}

void Game::welcome() {

cout << "---------------------------------------" << endl;

cout << "|============== CRIC-IN ==============|" << endl;

cout << "| |" << endl;

cout << "| Welcome to Virtual Cricket Game |" << endl;

cout << "---------------------------------------" << endl;

cout << endl << endl;

cout << "----------------------------------------------------" << endl;

cout << "|================== Instructions ==================|" << endl;

cout << "----------------------------------------------------" << endl;

cout << "| |" << endl;

cout << "| 1. Create 2 teams (Team-A and Team-B with 4 |" << endl;

cout << "| players each) from a given pool of 11 players.|" << endl;

cout << "| 2. Lead the toss and decide the choice of play. |" << endl;

cout << "| 3. Each innings will be of 6 balls. |" << endl;

cout << "----------------------------------------------------" << endl;

}

void Game::showAllPlayers() {

cout << endl;

cout << "---------------------------------------" << endl;

cout << "|========== Pool of Players ==========|" << endl;

cout << "---------------------------------------" << endl;

cout << endl;

for (int i = 0; i < totalPlayers; i++) {

cout << "\t\t[" << i << "] " << players[i] << endl;

}

}

int Game::takeIntegerInput() {

int n;

while (!(cin >> n)) {

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

cout << "Invalid input! Please try again with valid input: ";

}

return n;

}

bool Game::validateSelectedPlayer(int index) {

int n;

vector<Player> players;

players = teamA.players;

n = players.size();

for (int i = 0; i < n; i++) {

if (players[i].id == index) {

return false;

}

}

players = teamB.players;

n = players.size();

for (int i = 0; i < n; i++) {

if (players[i].id == index) {

return false;

}

}

return true;

}

void Game::selectPlayers() {

cout << endl;

cout << "------------------------------------------------" << endl;

cout << "|========== Create Team-A and Team-B ==========|" << endl;

cout << "------------------------------------------------" << endl;

for (int i = 0; i < playersPerTeam; i++) {

// Add player to team A

teamASelection:

cout << endl << "Select player " << i + 1 << " of Team A - ";

int playerIndexTeamA = takeIntegerInput();

if (playerIndexTeamA < 0 || playerIndexTeamA > 10) {

cout << endl << "Please select from the given players" << endl;

goto teamASelection;

} else if (!validateSelectedPlayer(playerIndexTeamA)) {

cout << endl << "Player has been already selected. Please select other player" << endl;

goto teamASelection;

} else {

Player teamAPlayer;

teamAPlayer.id = playerIndexTeamA;

teamAPlayer.name = players[playerIndexTeamA];

teamA.players.push\_back(teamAPlayer);

}

// Add player to team B

teamBSelection:

cout << endl << "Select player " << i + 1 << " of Team B - ";

int playerIndexTeamB = takeIntegerInput();

if (playerIndexTeamB < 0 || playerIndexTeamB > 10) {

cout << endl << "Please select from the given players" << endl;

goto teamBSelection;

} else if (!validateSelectedPlayer(playerIndexTeamB)) {

cout << endl << "Player has been already selected. Please select other player" << endl;

goto teamBSelection;

} else {

Player teamBPlayer;

teamBPlayer.name = players[playerIndexTeamB];

teamB.players.push\_back(teamBPlayer);

}

}

}

void Game::showTeamPlayers() {

vector<Player> teamAPlayers = teamA.players;

vector<Player> teamBPlayers = teamB.players;

cout << endl << endl;

cout << "--------------------------\t\t--------------------------" << endl;

cout << "|======= Team-A =======|\t\t|======= Team-B =======|" << endl;

cout << "--------------------------\t\t--------------------------" << endl;

for (int i = 0; i < playersPerTeam; i++) {

cout << "|\t" << "[" << i << "] " << teamAPlayers[i].name << "\t |"

<< "\t\t"

<< "|\t" << "[" << i << "] " << teamBPlayers[i].name << "\t |" << endl;

}

cout << "--------------------------\t\t--------------------------" << endl << endl;

}

void Game::toss() {

cout << endl;

cout << "-----------------------------------" << endl;

cout << "|========== Let's Toss ==========|" << endl;

cout << "-----------------------------------" << endl << endl;

cout << "Tossing the coin..." << endl << endl;

srand(time(NULL));

int randomValue = rand() % 2; // 0 or 1

switch (randomValue) {

case 0:

cout << "Team-A won the toss" << endl << endl;

tossChoice(teamA);

break;

case 1:

cout << "Team-B won the toss" << endl << endl;

tossChoice(teamB);

break;

}

}

void Game::tossChoice(Team tossWinnerTeam) {

cout << "Enter 1 to bat or 2 to bowl first. " << endl

<< "1. Bat" << endl

<< "2. Bowl " << endl;

int tossInput = takeIntegerInput();

cin.ignore(numeric\_limits<streamsize>::max(),'\n');

switch (tossInput) {

case 1:

cout << endl << tossWinnerTeam.name << " won the toss and elected to bat first" << endl << endl;

if (tossWinnerTeam.name.compare("Team-A") == 0) { // if Team-A is the toss winner

battingTeam = &teamA;

bowlingTeam = &teamB;

} else { // else Team-B is the toss winner

battingTeam = &teamB;

bowlingTeam = &teamA;

}

break;

case 2:

cout << endl << tossWinnerTeam.name << " won the toss and choose to bowl first" << endl << endl;

if (tossWinnerTeam.name.compare("Team-A") == 0) { // if Team-A is the toss winner

bowlingTeam = &teamA;

battingTeam = &teamB;

} else { // else Team-B is the toss winner

bowlingTeam = &teamB;

battingTeam = &teamA;

}

break;

default:

cout << endl << "Invalid input. Please try again:" << endl << endl;

tossChoice(tossWinnerTeam);

break;

}

}

void Game::startFirstInnings() {

cout << "\t\t ||| FIRST INNINGS STARTS ||| " << endl << endl;

isFirstInnings = true;

initializePlayers();

playInnings();

}

void Game::startSecondInnings() {

cout << "\t\t ||| SECOND INNINGS STARTS ||| " << endl << endl;

isFirstInnings = false;

// Swap battingTeam and bowlingTeam pointers

Team tempTeam = \*battingTeam;

\*battingTeam = \*bowlingTeam;

\*bowlingTeam = tempTeam;

// Select the batsman and the bowler for 2nd Innings

initializePlayers();

playInnings();

}

void Game::initializePlayers() {

// Choose batsman and bowler: Initialize \*batsman and \*bowler

batsman = &battingTeam->players[0];

bowler = &bowlingTeam->players[0];

cout << battingTeam->name << " - " << batsman->name << " is batting " << endl;

cout << bowlingTeam->name << " - " << bowler->name << " is bowling " << endl << endl;

}

void Game::playInnings() {

for (int i = 0; i < maxBalls; i++) {

cout << "Press Enter to bowl...";

getchar();

cout << "Bowling..." << endl;

bat();

if (!validateInningsScore()) {

break;

}

}

}

void Game::bat() {

srand(time(NULL));

int runsScored = rand() % 6;

// Update batting team and batsman score

batsman->runsScored = batsman->runsScored + runsScored;

battingTeam->totalRunsScored = battingTeam->totalRunsScored + runsScored;

batsman->ballsPlayed = batsman->ballsPlayed + 1;

// Update bowling team and bowler score

bowler->ballsBowled = bowler->ballsBowled + 1;

bowlingTeam->totalBallsBowled = bowlingTeam->totalBallsBowled + 1;

bowler->runsGiven = bowler->runsGiven + runsScored;

if (runsScored != 0) { // if runsScored = 1, 2, 3, 4, 5, or 6

cout << endl << bowler->name << " to " << batsman->name << " " << runsScored << " runs!" << endl << endl;

showGameScorecard();

} else { // else runScored = 0 and the batsman is â€˜OUTâ€™

cout << endl << bowler->name << " to " << batsman->name << " OUT!" << endl << endl;

battingTeam->wicketsLost = battingTeam->wicketsLost + 1;

bowler->wicketsTaken = bowler->wicketsTaken + 1;

showGameScorecard();

int nextPlayerIndex = battingTeam->wicketsLost;

batsman = &battingTeam->players[nextPlayerIndex];

}

}

bool Game::validateInningsScore() {

if (isFirstInnings) {

if (battingTeam->wicketsLost == playersPerTeam || bowlingTeam->totalBallsBowled == maxBalls) {

cout << "\t\t ||| FIRST INNINGS ENDS ||| " << endl << endl;

cout << battingTeam->name << " " << battingTeam->totalRunsScored << " - "

<< battingTeam->wicketsLost << " (" << bowlingTeam->totalBallsBowled

<< ")" << endl;

cout << bowlingTeam->name << " needs " << battingTeam->totalRunsScored + 1

<< " runs to win the match" << endl << endl;

return false;

}

} else { // Else 2nd innings

if (battingTeam->totalRunsScored > bowlingTeam->totalRunsScored) { //Case1: If batting team score > bowling team score

cout << battingTeam->name << " WON THE MATCH" << endl << endl;

return false;

//Case2: Batting team is all OUT OR Bowling team is done bowling

} else if (battingTeam->wicketsLost == playersPerTeam || bowlingTeam->totalBallsBowled == maxBalls) {

if (battingTeam->totalRunsScored < bowlingTeam->totalRunsScored) {

cout << bowlingTeam->name << " WON THE MATCH" << endl << endl;

} else {

cout << "MATCH DRAW" << endl << endl;

}

return false;

}

}

return true;

}

void Game::showGameScorecard() {

cout << "--------------------------------------------------------------------------" << endl;

cout << "\t" << battingTeam->name << " " << battingTeam->totalRunsScored << " - "

<< battingTeam->wicketsLost << " (" << bowlingTeam->totalBallsBowled << ") | " << batsman->name

<< " " << batsman->runsScored << " (" << batsman->ballsPlayed << ") \t" << bowler->name << " "

<< bowler->ballsBowled << " - " << bowler->runsGiven << " - " << bowler->wicketsTaken << "\t" << endl;

cout << "--------------------------------------------------------------------------" << endl << endl;

}

void Game::showMatchSummary() {

cout << "\t\t ||| MATCH ENDS ||| " << endl << endl;

cout << battingTeam->name << " " << battingTeam->totalRunsScored << "-" << battingTeam->wicketsLost << " (" << bowlingTeam->totalBallsBowled << ")" << endl;

cout << "==========================================" << endl;

cout << "| PLAYER \t BATTING \t BOWLING |" << endl;

for (int j = 0; j < playersPerTeam; j++) {

Player player = battingTeam->players[j];

cout << "|----------------------------------------|" << endl;

cout << "| " << "[" << j << "] " << player.name << " \t "

<< player.runsScored << "(" << player.ballsPlayed << ") \t\t "

<< player.ballsBowled << "-" << player.runsGiven << "-"

<< player.wicketsTaken << "\t |" << endl;

}

cout << "==========================================" << endl << endl;

cout << bowlingTeam->name << " " << bowlingTeam->totalRunsScored << "-" << bowlingTeam->wicketsLost << " (" << battingTeam->totalBallsBowled << ")" << endl;

cout << "==========================================" << endl;

cout << "| PLAYER \t BATTING \t BOWLING |" << endl;

for (int i = 0; i < playersPerTeam; i++) {

Player player = bowlingTeam->players[i];

cout << "|----------------------------------------|" << endl;

cout << "| " << "[" << i << "] " << player.name << " \t "

<< player.runsScored << "(" << player.ballsPlayed << ") \t\t "

<< player.ballsBowled << "-" << player.runsGiven << "-"

<< player.wicketsTaken << "\t |" << endl;

}

cout << "==========================================" << endl << endl;

}

int main() {

Game game;

game.welcome();

cout << "\nPress Enter to continue";

getchar();

game.showAllPlayers();

cout << "\nPress Enter to continue";

getchar();

game.selectPlayers();

game.showTeamPlayers();

cin.ignore(numeric\_limits<streamsize>::max(),'\n');

cout << "\nPress Enter to toss";

getchar();

game.toss();

game.startFirstInnings();

game.startSecondInnings();

game.showMatchSummary();

return 0;

}