TASK-5

#include <iostream>

using namespace std;

class Player

{

private:

string name;

int run;

public:

set\_name(string s)

{

name = s;

}

set\_run(int r)

{

run = r;

}

int get\_run()

{

return run;

}

string get\_name()

{

return name;

}

};

class Team

{

private:

Player p[11];

string name;

public:

set\_name(string s)

{

name = s;

}

get\_name()

{

return name;

}

set\_players(void)

{

for(int i=0;i<11;i++)

{

string s;

int r;

cin>>s>>r;

p[i].set\_name(s);

p[i].set\_run(r);

}

}

string highest\_scorer()

{

int max = 0;

string str;

for(int i=0;i<11;i++)

{

if(p[i].get\_run()>=max)

{

max = p[i].get\_run();

str = p[i].get\_name();

}

}

return str;

}

int highest\_run()

{

int max = 0;

for(int i=0;i<11;i++)

{

if(p[i].get\_run()>=max)

{

max = p[i].get\_run();

}

}

return max;

}

int total()

{

int t = 0;

for(int i=0;i<11;i++)

{

t+=p[i].get\_run();

}

return t;

}

};

class Match

{

private:

Team team[2];

public:

Match()

{

team[0].set\_players();

team[1].set\_players();

}

string winner()

{

if(team[0].total()>team[1].total())

return team[0].get\_name();

else if(team[0].total()<team[1].total())

return team[1].get\_name();

else return "Draw";

}

string man\_of\_match()

{

if(team[0].highest\_run()>team[1].highest\_run())

return team[0].highest\_scorer();

else if(team[0].highest\_run()<team[1].highest\_run())

return team[1].highest\_scorer();

else return "We have two man of the matches here";

}

};