Welcome

Cricket Scorecard Management Program

Presented by,

Md Arafat Hossain Himel

IT23038

Supervised by,

DR.MR.Ziaur Rahman

Associate Professor

Dept.Of ICT,MBSTU

Overview

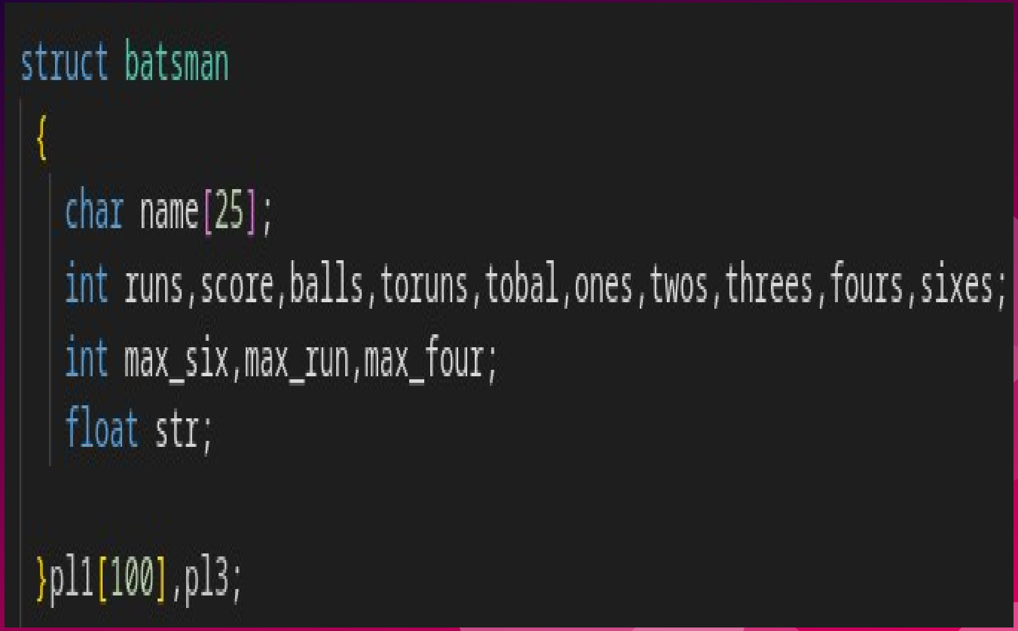
This program collects, processes, and displays cricket match data, focusing on::

- **Batsman Performance: Runs, balls faced, and strike rate.**
- **Bowler Performance: Overs, runs conceded, wickets taken, and economy rate.**
- **Allows users to view individual player statistics and match summaries.**

Structure Definition: Batsman

Explanation:

- The batsman structure holds player data, including individual runs, balls, boundaries (fours and sixes), and strike rate calculation fields.
- Arrays pl1[] store multiple batsman details, while pl3 holds aggregated or max values.



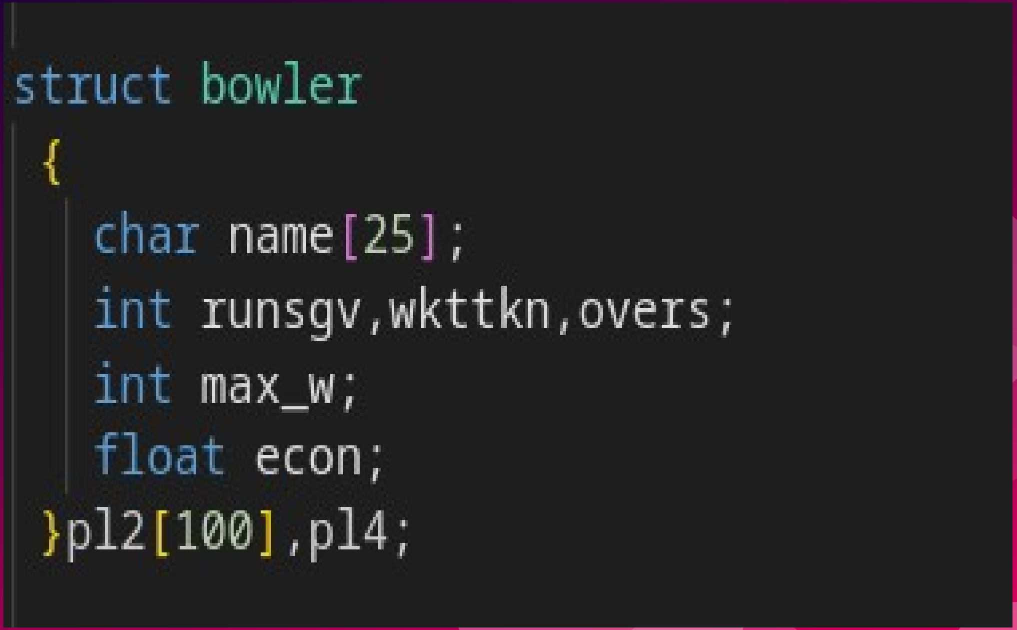
```
struct batsman
 {
   char name[25];
   int runs,score,balls,toruns,tobal,ones,twos,threes,fours,sixes;
   int max_six,max_run,max_four;
   float str;

 }pl1[100],pl3;
```

Structure Definition: Bowler

Explanation:

- The bowler structure holds data on overs bowled, runs conceded, wickets taken, and economy rate.

- Arrays pl2[] store individual bowlers' data, while pl4 aggregates and tracks maximum performance metrics.

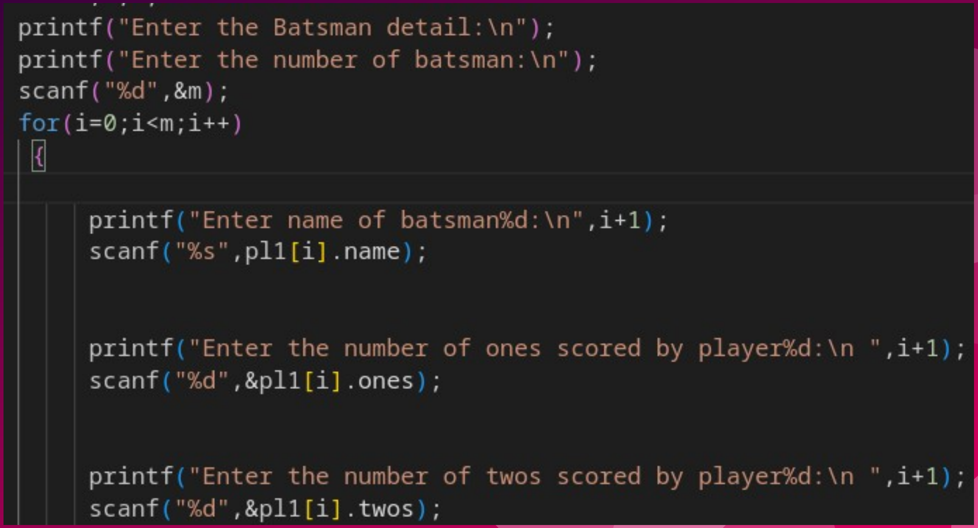


```c
struct bowler
 {
   char name[25];
   int runsgv,wkttkn,overs;
   int max_w;
   float econ;
 }pl2[100],pl4;
```

Input Section: Batsman Data Collection

Explanation:

- The user enters the number of batsmen (m), followed by individual details like name, boundaries, and balls played.
- Data is stored in the pl1[] array.
- This portion of the code also take additional inputs for threes,fours,sixes and balls played.

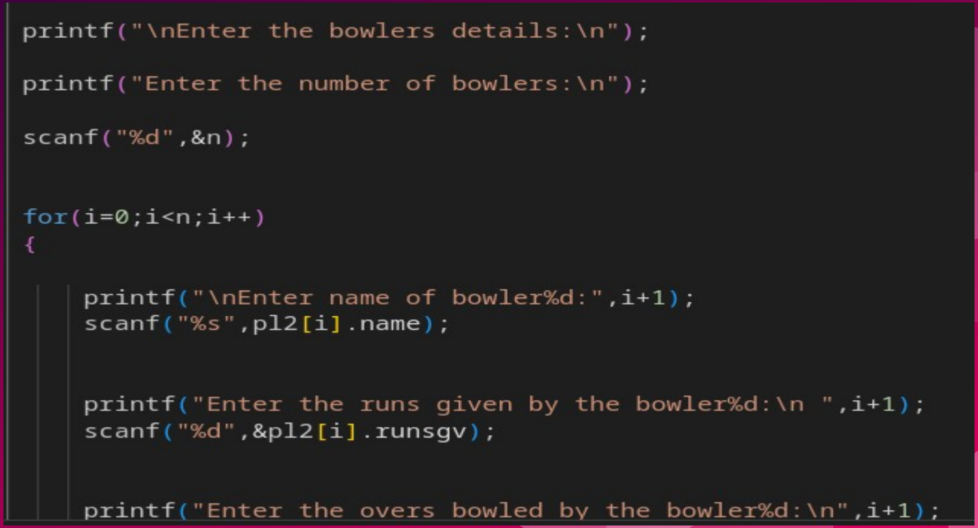


```
printf("Enter the Batsman detail:\n");
printf("Enter the number of batsman:\n");
scanf("%d",&m);
for(i=0;i<m;i++)
 {

    printf("Enter name of batsman%d:\n",i+1);
    scanf("%s",pl1[i].name);


    printf("Enter the number of ones scored by player%d:\n ",i+1);
    scanf("%d",&pl1[i].ones);


    printf("Enter the number of twos scored by player%d:\n ",i+1);
    scanf("%d",&pl1[i].twos);
```

Input Section: Bowler Data Collection

Explanation:

- User inputs bowler details like runs conceded, overs bowled, and wickets taken.
- Data is stored in the pl2[] array for each bowler.
- It also takes additional inputs for wickets taken.



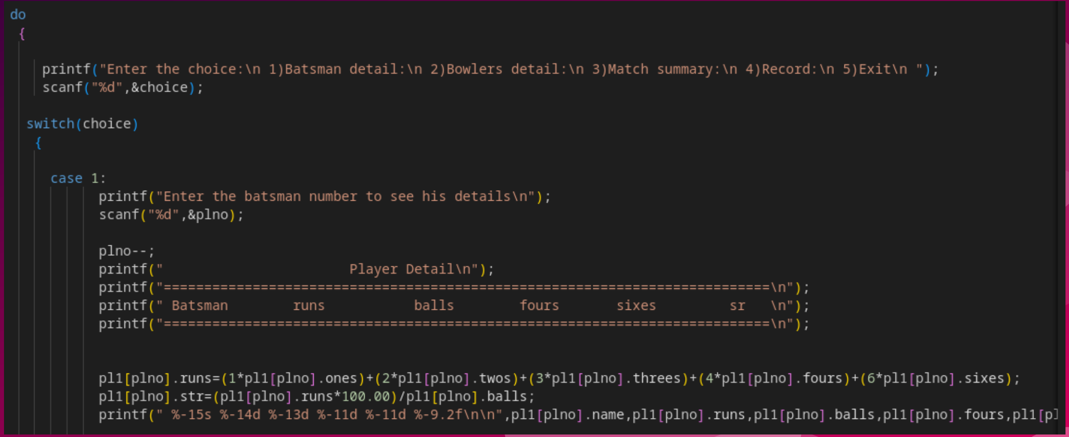
```
printf("\nEnter the bowlers details:\n");

printf("Enter the number of bowlers:\n");

scanf("%d",&n);


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

    printf("\nEnter name of bowler%d:",i+1);
    scanf("%s",pl2[i].name);


    printf("Enter the runs given by the bowler%d:\n ",i+1);
    scanf("%d",&pl2[i].runsgv);


    printf("Enter the overs bowled by the bowler%d:\n",i+1);
```

Main Menu and Functionality:

Explanation:

Users can choose different functionalities:

- Option 1: Display a batsman’s individual performance.
- Option 2: Display a bowler’s statistics.
- Option 3: Show a summary of the match.
- Option 4: Find and display maximum records for runs, wickets, etc.

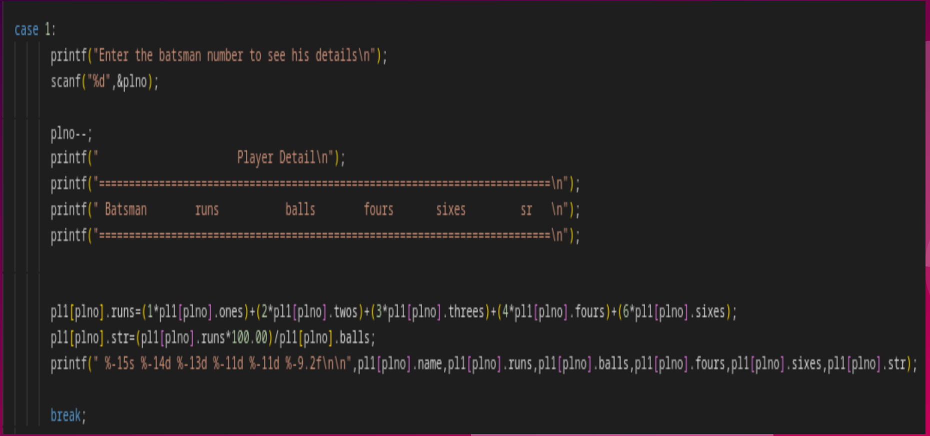


```c
do
 {

   printf("Enter the choice:\n 1)Batsman detail:\n 2)Bowlers detail:\n 3)Match summary:\n 4)Record:\n 5)Exit\n ");
   scanf("%d",&choice);

  switch(choice)
   {

    case 1:
          printf("Enter the batsman number to see his details\n");
          scanf("%d",&plno);

          plno--;
          printf("                       Player Detail\n");
          printf("===========================================================================\n");
          printf(" Batsman        runs           balls        fours       sixes         sr   \n");
          printf("===========================================================================\n");


          pl1[plno].runs=(1*pl1[plno].ones)+(2*pl1[plno].twos)+(3*pl1[plno].threes)+(4*pl1[plno].fours)+(6*pl1[plno].sixes);
          pl1[plno].str=(pl1[plno].runs*100.00)/pl1[plno].balls;
          printf(" %-15s %-14d %-13d %-11d %-11d %-9.2f\n\n",pl1[plno].name,pl1[plno].runs,pl1[plno].balls,pl1[plno].fours,pl1[p
```

Case 1: Batsman Details Display

Explanation:

- The program calculates total runs and strike rate for a selected batsman and displays them.

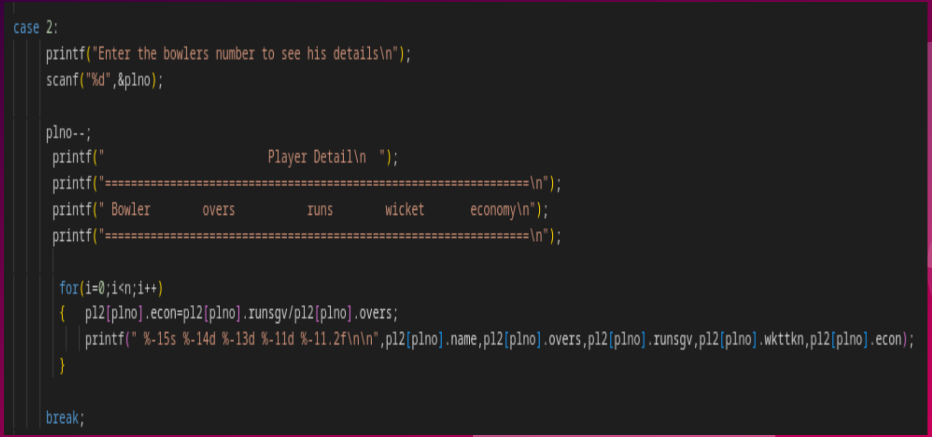


```
case 1:
        printf("Enter the batsman number to see his details\n");
        scanf("%d",&plno);

        plno--;
        printf("                       Player Detail\n");
        printf("==========================================================================\n");
        printf(" Batsman        runs           balls        fours      sixes         sr   \n");
        printf("==========================================================================\n");


        pl1[plno].runs=(1*pl1[plno].ones)+(2*pl1[plno].twos)+(3*pl1[plno].threes)+(4*pl1[plno].fours)+(6*pl1[plno].sixes);
        pl1[plno].str=(pl1[plno].runs*100.00)/pl1[plno].balls;
        printf(" %-15s %-14d %-13d %-11d %-11d %-9.2f\n\n",pl1[plno].name,pl1[plno].runs,pl1[plno].balls,pl1[plno].fours,pl1[plno].sixes,pl1[plno].str);

        break;
```

Case 2: Bowler Details Display

Explanation:

- The program calculates total runs and strike rate for a selected batsman and displays them.

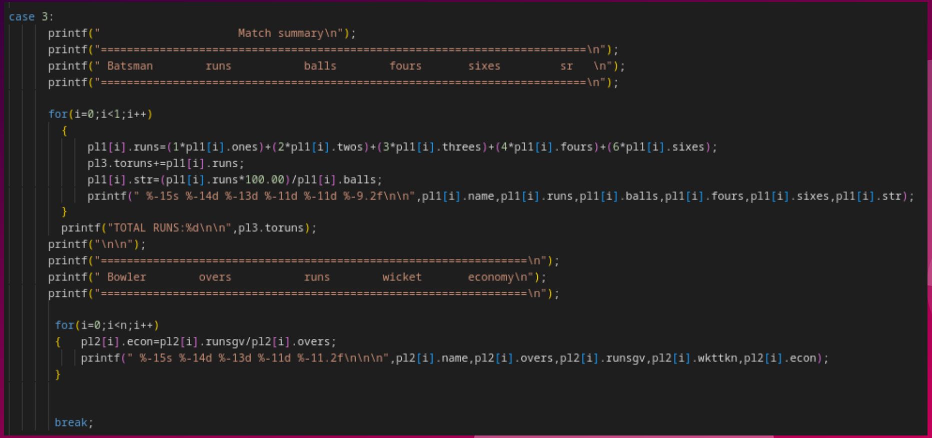


```c
case 2:
    printf("Enter the bowlers number to see his details\n");
    scanf("%d",&plno);

    plno--;
     printf("                         Player Detail\n  ");
     printf("=================================================================\n");
     printf(" Bowler       overs           runs        wicket       economy\n");
     printf("=================================================================\n");

      for(i=0;i<n;i++)
      {   pl2[plno].econ=pl2[plno].runsgv/pl2[plno].overs;
          printf(" %-15s %-14d %-13d %-11d %-11.2f\n\n",pl2[plno].name,pl2[plno].overs,pl2[plno].runsgv,pl2[plno].wkttkn,pl2[plno].econ);
      }

    break;
```

Case 3: Match Summary

Explanation:

- Calculates and displays total runs scored by a batsman in the match.

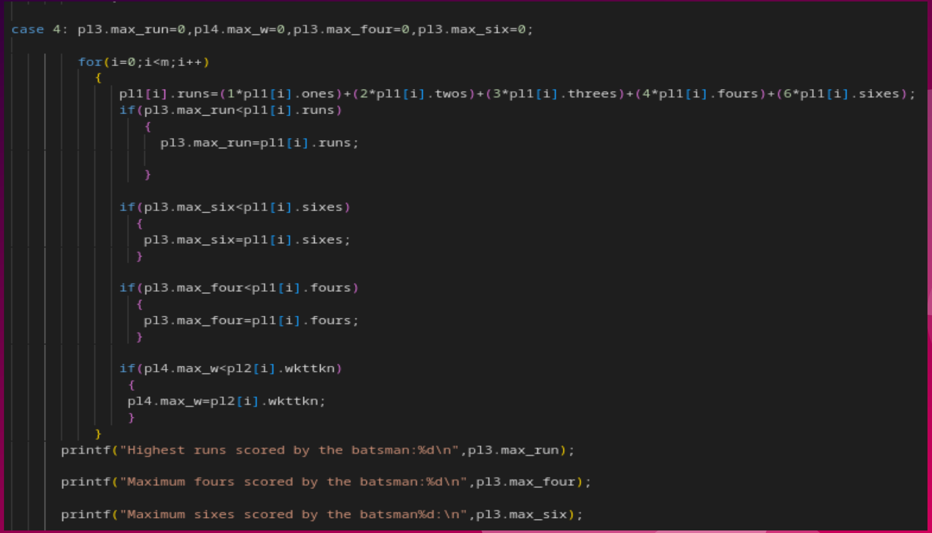


```c
case 3:
        printf("                    Match summary\n");
        printf("=========================================================================\n");
        printf(" Batsman        runs            balls        fours       sixes         sr   \n");
        printf("=========================================================================\n");

        for(i=0;i<1;i++)
          {
              pl1[i].runs=(1*pl1[i].ones)+(2*pl1[i].twos)+(3*pl1[i].threes)+(4*pl1[i].fours)+(6*pl1[i].sixes);
              pl3.toruns+=pl1[i].runs;
              pl1[i].str=(pl1[i].runs*100.00)/pl1[i].balls;
              printf(" %-15s %-14d %-13d %-11d %-11d %-9.2f\n\n",pl1[i].name,pl1[i].runs,pl1[i].balls,pl1[i].fours,pl1[i].sixes,pl1[i].str);
          }
          printf("TOTAL RUNS:%d\n\n",pl3.toruns);
        printf("\n\n");
        printf("================================================================\n");
        printf(" Bowler        overs            runs        wicket       economy\n");
        printf("================================================================\n");

         for(i=0;i<n;i++)
         {   pl2[i].econ=pl2[i].runsgv/pl2[i].overs;
             printf(" %-15s %-14d %-13d %-11d %-11.2f\n\n\n",pl2[i].name,pl2[i].overs,pl2[i].runsgv,pl2[i].wkttkn,pl2[i].econ);
         }


         break;
```

Case 4: Maximum Records

Explanation:

- Finds and displays the maximum runs scored by a batsman and the maximum wickets taken by a bowler.
- The program also finds and displays maximum fours hit by the batsman,maximum six hit by the batsman.



```c
case 4: pl3.max_run=0,pl4.max_w=0,pl3.max_four=0,pl3.max_six=0;

        for(i=0;i<m;i++)
          {
            pl1[i].runs=(1*pl1[i].ones)+(2*pl1[i].twos)+(3*pl1[i].threes)+(4*pl1[i].fours)+(6*pl1[i].sixes);
            if(pl3.max_run<pl1[i].runs)
              {
                pl3.max_run=pl1[i].runs;

              }

            if(pl3.max_six<pl1[i].sixes)
             {
              pl3.max_six=pl1[i].sixes;
             }

            if(pl3.max_four<pl1[i].fours)
             {
              pl3.max_four=pl1[i].fours;
             }

            if(pl4.max_w<pl2[i].wkttkn)
             {
             pl4.max_w=pl2[i].wkttkn;
             }
          }
      printf("Highest runs scored by the batsman:%d\n",pl3.max_run);

      printf("Maximum fours scored by the batsman:%d\n",pl3.max_four);

      printf("Maximum sixes scored by the batsman%d:\n",pl3.max_six);
```

Conclusion:

Features:

- **Player Data Handling:** The program collects detailed data for both batsmen (e.g., name, runs, boundaries, balls faced) and bowlers (e.g., runs given, wickets taken, overs bowled).
- **Dynamic Menu Options:** The menu allows users to view different statistics like individual player details, match summaries, and record-breaking performances.
- **Strike Rate and Economy Calculation:** The program calculates key performance indicators like strike rate for batsmen and economy rate for bowlers automatically.
- **Match Summary Display**: Summarizes match data, displaying total runs scored by batsmen and performance statistics for all bowlers.

Limitations:

- **Limited Error Handling**: There’s no error handling for incorrect inputs or out-of-bound array indices (e.g., selecting a non-existent player number).
- **Fixed Data Size:** The arrays (pl1[100], pl2[100]) limit the maximum number of players to 100, which is hard-coded and may not suit all use cases. Dynamic memory allocation could be better.
- **Basic User Interface:** The program only uses basic text-based input/output. It could be enhanced with a graphical interface (GUI) or formatted tables.
- **Limited to One Match**: The code structure doesn’t allow for managing or comparing statistics across multiple matches or players over time.