

SRM

PPS MINI PROJECT

BY:

TARUN RAMKUMAR (RA2111009010117)

SASI KUMAR.K(RA2111009010115)

CRICKET SCORE BOARD (BATTING AVG)

```

#include<stdio.h>

#include<conio.h>

#include <string.h>

struct cricket
{
    char name [100];
    char team_name[100];
    int batting_average;
};

void main()
{
    struct cricket player[200];
    int i,n;
    char ch,team[100];
    printf("HOW MANY PLAYERS\n");
    scanf("%d",&n);
    for (i=0;i<n;i++)
    {
        printf("\n INPUT THE NAME OF THE PLAYER %d : ",i+1);
        scanf("%s",player[i].name);
        printf("\n INPUT THE TEAM NAME OF THE PLAYER %d :",i+1);
        scanf("%s",player[i].team_name);
        printf("\n INPUT THE BATTING AVERAGE OF THE PLAYER %d:",i+1);
        scanf("%d",&player[i].batting_average);
    }

    printf("=====\n");
    printf(" PLAYER'S NAME COUNTRY BATTING AVERAGE\n");

```

```

printf("=====\n");
for(i=0;i<=n;i++)
printf(" %s      %s      %d\n",player[i].name, player[i].team_name,player[i].batting_average);
for(i=0;i<n;i++)
{
printf("\n INPUT THE NAME OF THE PLAYER %d : ",i+1);
scanf("%s",player[i].name);
printf("\n INPUT THE TEAM NAME OF THE PLAYER %d :",i+1);
scanf("%s",player[i].team_name);
printf("\n INPUT THE BATTING AVERAGE OF THE PLAYER %d:",i+1);
scanf("%d",&player[i].batting_average);
}

```

```

printf("=====\n");
printf(" PLAYER'S NAME COUNTRY BATTINGAVERAGE\n");
printf("=====\n");
for(i=0;i<=n;i++)
printf(" %20s %20s%d\n",player[i].name, player[i].team_name,player[i].batting_average);
for(i=0;i<n;i++)
{
printf("\n INPUT THE NAME OF THE PLAYER %d : ",i+1);
scanf("%s",player[i].name);
printf("\n INPUT THE TEAM NAME OF THE PLAYER %d :",i+1);
scanf("%s",player[i].team_name);
printf("\n INPUT THE BATTING AVERAGE OF THE PLAYER %d:",i+1);
scanf("%d",&player[i].batting_average);
}

```

```

printf("=====\n");
printf(" PLAYER'S NAME COUNTRY BATTING AVERAGE\n");
printf("=====\n");

```

```

for(i=0;i<=n;i++)
printf("%20s%d\n",player[i].name,player[i].team_name,player[i].batting_average);
read:
printf("\n\n INPUT FOR WHICH TEAM YOU WANT TO LIST : ");
scanf("%s",team);
printf("\n %s \n",team);
printf("=====\n");
printf(" PLAYER'S NAME BATTING AVERAGE \n");
printf("=====\n");
for(i=0;i<=n;i++)

printf("%20s%20s%d\n",player[i].name,player[i].team_name,player[i].batting_average);
printf(" \n\n DO YOU WANT TO LIST ANY OTHER TEAM ? (Y/N) : ");
ch=getch();
if (ch == 'Y' || ch == 'y')
goto read;
getch();
}

```

ALGORITHM :

Step1:START

Step2:Declare structure cricket . Name as string. Team name as string . Batting average as integer .

Step3:Declare i,n and team name

Step4:Print “how many players” and get the value of ‘n’

Step5:Get the names, team names and the batting averages

Step6:Print the names, team names and batting averages

Step7:Form a team using the data

Step8:Print the team players and average

Step9:Stop .

Syntax :

while(condition) { Statements; Increment For loop: For Loop in C is a statement which allows code to be repeatedly executed.

For loop contains 3 parts Initialization, Condition and Increment or Decrements.

/decrements (++ or --); } syntax for (assign value; decision statement; increment operator (or)decrement

operator) do-while : 8 A do-while Loop in C is similar to a while loop, except that a do-while loop is executed at least one time.

A do while loop is a control flow statement that executes a block of code at least once, and then repeatedly executes the block, or not, depending on a given condition at the end of the block (in while).

syntax `do { Statements; Increment/decrement (++ or --) } while();` When use do..while Loop : When we need to repeat the statement block at least 1 time then we use do-while loop.

9 Nested loop : In Nested loop one loop is placed within another loop body. When we need to repeat loop body itself n number of times use nested loops.

Nested loops can be designed up to 255 blocks. String : A string is a sequence of characters stored in a character array.

A string is a text enclosed in double quotation marks.

A character such as 'd' is not a string and it is indicated by single quotation marks. 'C' provides standard library functions to manipulate strings in a program. `Strcmp()` : The `strcmp()` function is used to compare two strings two strings `str1` and `str2` .

If two strings are same then strcmp() returns 0 , otherwise, it returns a non-zero value. This function compares strings character by character using ASCII value of the characters. Goto : goto is a jumping statement in c language, which transfer the program's control from one statement to another statement (where label is defined).

GITHUB link:

<https://github.com/tk6300/trk/commit/7686ea34c6ca0def10d63410709b3996bada2fa5>

out put:

The screenshot shows the OnlineGDB interface. The code in the editor is as follows:

```
main.c
41 printf(" PLAYER'S NAME COUNTRY BATTING AVERAGE\n");
42 printf("===== \n");
43 for(i=0;i<n;i++)
44 printf(" %20s %20s\n",player[i].name, player[i].team_name,player[i].batting_average);
45 for(i=0;i<n;i++)
```

The output window shows the following text:

```
main.c:59:8: warning: too many arguments for format [-Wformat-extra-args]
59 | printf("%20s\n",player[i].name,player[i].team_name,player[i].batting_average);
    | ^~~~~~
HOW MANY PLAYERS
2
INPUT THE NAME OF THE PLAYER 1 : t
INPUT THE TEAM NAME OF THE PLAYER 1 : iu
INPUT THE BATTING AVERAGE OF THE PLAYER 1:98
INPUT THE NAME OF THE PLAYER 2 : ee
INPUT THE TEAM NAME OF THE PLAYER 2 : ikjj
INPUT THE BATTING AVERAGE OF THE PLAYER 2:65
=====
PLAYER'S NAME COUNTRY BATTING AVERAGE
t          iu          98
ee         ikjj         65
```

Strcmp() : The strcmp() function is used to compare two strings two strings str1 and str2 . If two strings are same then strcmp() returns 0 , otherwise, it returns a non-zero value. This function compares strings character by character using ASCII value of the characters.

Goto : goto is a jumping statement in c language, which transfer the program's control from one statement to another statement (where label is defined).

