**Basketball game strategy**

A basketball coach wants to implement a strategy to keep his oldest player fresh. His strategy is based on replacing the oldest player on court from the Bench after he plays 10% minutes of his age.

The numbers of players are from 1 to 12. A 5 will be selected randomly to play on court. Every time the time is up for the oldest, the coach will select the second bench player alternatively from left side or right side of the bench. In the same way the times up player will be added alternatively to the left or to the right of the bench.

At the end of the game the coach needs a list of all the players with their numbers sorted by the minutes they played, and anther list sorted by age, number and minutes played.



**Classes needed:**

Class Player with number, age and minutes

Class courtPlayer and Class BenchPlayer which inherit from Player add a status ‘C’ or ‘B’

**Data Structures needed:**

1. The locker room where the players are waiting is an array of 12 players.
2. A sorted circular linked list of 5 players on court sorted by player number from the center
3. An unsorted doubly linked list of 7 players on the bench

**The game(main program)**

The game consists of 4 quarters of 12 minutes each. The coach starts the first quarter by creating randomly the linkedlist of courtplayers and eventually the doubly linked list of benchplayer. The second quarter should start with the end of the first quarter players.

***The deliverables*** (*Hint: the list of players should be merged* )

: ***Report 1***

Number Minutesplayed

2 40

7 38

3 35

And ***Report 2***

Age Number Minutesplayed

26 3 35

29 2 40

39 7 38

**Advanced Option:**

Print ***Report 3***

Replacement Player Replacing replacement time in the quarter or in total

3 7 Q1: 4

2 3 Q2:2 or 14 minutes from beginning