IPL DREAM 11

Main Functions:

- batsman
  - batsman runs
  - ball faced by the batsman
  - striker rate
- bowler
  - runs given by bowler
  - wickets taken by bowler
  - economy rate
- wicket keeper
- all rounder
  - players in both batsman and bowler list
- Dream-18
- Dream-11

Additional functions:

- limit(n,dictionary) - returns first n items as a dictionary from the given dictionary.
- reduce(n,player,dictionary) – reduce the players list in a dictionary to n items.

Criteria:

- Batsman - based on top striker rate and top batsman run.
- Bowler - based on low economy rate and more wickets taken.
- Wicket Keeper – based on more no.of wickets (stumped).
- All Rounder – top players in both batsman and bowler list.

PSEUDO CODE:

```
    # def for take top n player
1. begin
2.      def limit(n,dicty):
3.          ct ← 0
4.          Top = {}
5.          for i in dicty do
6.              ct ← ct + 1
7.              if ct < n+1 then
8.                  Top[i]=dicty[i]
9.              else
10.                 break
11.             end
12.         end
13.         return Top
14.     end
15.
        # batsman selection
16.     def batsman():
17.         import csv
18.         open("ipl-stats.csv") in read mode as ipl:
19.             Dict Reader(ipl)
20.             batsman_runs={}
21.             for ball in csvreader:
22.                 if batsman in batsman_runs:
23.                     batsman_runs  ← batsman_runs + runs
24.                 else:
25.                     batsman_runs ← runs
26.                 end
27.             end
28.         end
29.
30.         open("ipl-stats.csv") in read mode as ipl:
31.             Dict Reader(ipl)
32.             balls_faced={}
33.             for ball in csvreader:
34.                 if ball is not a wideball then
35.                     if batsman in balls_faced then
36.                         batsman ← +1
37.                     else
```

```
38.                 batsman ← 1
39.             end
40.         end
41.     end
42.
43.     strike_rate={}
44.     for i in batsman_runs:
45.         i ← (batsman_runs[i] / balls_faced[i]) * 100
46.     end
47.
48.     strike_rate in decreasing order
49.     batsman_runs in decreasing order
50.
        # best batsmans
51.     best={}
52.     for i in strike_rate do
53.         if i in batsman_runs then
54.             best[i] ← (strike_rate[i] * batsman_runs[i])
55.         end
56.     end
57.     arrange dictionary(best) in decreasing order
58.     return best
59. end
60. def bowlers():
61.     import csv
        # runs given by bowlers
62.     open("ipl-stats.csv") as stats_ipl:
63.         DictReader(stats_ipl)
64.         b_runs = {}
65.         for row in stats do
66.             if bowler not in b_runs.keys then
67.                 bowler ← total_runs
68.             else
69.                 bowler  ←  + total_runs
70.             end
71.         end
72.     end
        # balls by bowlers
73.     open("ipl-stats.csv") as stats_ipl
74.         DictReader(stats_ipl)
75.         b_balls = {}
76.         for row in stats do
77.             balls ← 0
```

```
78.                 if ball is not a wide and noball then
79.                     if bowler not in b_balls.keys() then
80.                         bowler ← 1
81.                     else
82.                         bowler ← +1
83.                     end
84.                 end
85.             end
86.         end
            # wickets taken by bowlers
87.         open("ipl-stats.csv") as stats_ipl
88.             DictReader(stats_ipl)
89.             b_wickets = {}
90.             wicket_type= ('bowled','caught','caught and bowled','hit wicket',
                              'lbw','stumped')
91.             for row in stats do
92.                 if dismissal_kind in wicket_type then
93.                     if bowler not in b_wickets.keys() then
94.                         b_wickets["bowler"] ← 1
95.                     else
96.                         b_wickets["bowler"] ← +1
97.                     end
98.                 end
99.             end
100.        end
101.        b_wickets in decreasing order

            # economy rate of bowlers
102.        bowlers_eco = {}
103.        for i in bowler_name do
104.            if b_balls[i] > 500 then
105.                bowlers_eco[i] ← (b_runs[i]/(b_balls[i]/6))
106.            end
107.        end
108.        arrange bowlers_eco in decreasing order

            # eff bowlers
109.        best={}
110.        for i in bowlers_eco do
111.            best[i] = (b_wickets[i] / bowlers_eco[i])
112.        end
113.        arrange best in decreasing order
114.        return best
```

```
115.  end
116.  def wk():
117.      import csv
118.      open("ipl-stats.csv") as ipl
119.          stats ← csv.DictReader(ipl)
120.          wk = {}
121.          for row in stats do
122.              if dismissal_kind == "stumped" then
123.                  if fielder in wk then
124.                      wk[fielder] ← +1
125.                  else
126.                      wk[fielder] ← 1
127.                  end
128.              end
129.          end
130.          arrange wk in decreasing order
131.          return wk
132.      end
133.  end
134.  def all_rounders():
135.      bt ← batsman()
136.      bw ← bowlers()
137.      bt ← bt - limit(6,bt)
138.      bw ← bw - limit(6,bw)
139.
140.      best={}
141.      for i in bt do
142.          if i in bw then
143.              best[i] ← bt[i] * bw [i]
144.          end
145.      end
146.      arrange best in decreasing order
147.      return best
148.  end
149.  def Dream_18():
150.      bt=[x for x in limit(6,batsman())]
151.      bw=[x for x in limit(6,bowlers())]
152.      wt=[x for x in limit(2,wk())]
153.      ar=[x for x in limit(4,all_rounders())]
154.
155.      players={"Batsman":bt,"Bowlers":bw,"Wicket Keepers":wt,
156.                   "All Rounders":ar}
157.      return players
```

```
    end
    def reduce(n,player,p):
        for i in range(n,len(p[player])):
            p[player].pop()
        end
        return p
    end

    def Dream_11():
        p=Dream_18()
        p=reduce(4,"Batsman",p)
        p=reduce(4,"Bowlers",p)
        p=reduce(1,"Wicket Keepers",p)
        p=reduce(2,"All Rounders",p)
        return p
    end
end

```

SQUAD:-

- CHRIS JONATHAN SELWYN ʕ•ᴥ•ʔ
- DEEPAK VASAN (✿◠‿◠)
- MUTHU SRIMAN :-}
- HARI HARA SUDHAN (˘▾˘~)
- BHANUPRATHAP (｡◕‿◕｡)
- DHANUSH (^ᴥ^)