```cpp
#include "Dancer.h"
#include <iostream>
using namespace std;
Dancer::Dancer()
{
    id = 0;
    hours = 0;
    rate = 0;
}

void Dancer::read_val(ifstream &inputfile)
{
    inputfile >> id >> hours >> rate;
}

void Dancer::print_val(ofstream& outputfile)
{
        outputfile << id << " " << hours << " " << rate << endl;
}

void Dancer::sort_by_id(Dancer d[], int k)
{
    int temp;
    float tempr;
    for (int i = 0; i < 12; i++)
    {
        for (int j = i + 1; j < 12; j++)
        {
            if (d[i].id > d[j].id)
            {

                temp = d[i].id;
                d[i].id = d[j].id;
                d[j].id = temp;
                temp = d[i].hours;
                d[i].hours = d[j].hours;
                d[j].hours = temp;
                tempr = d[i].rate;
                d[i].rate = d[j].rate;
                d[j].rate = tempr;
            }
        }
    }
    for (int i = 0; i < 12; i++)
    {
        for (int j = i + 1; j < 12; j++)
        {
            if (d[i].id == d[j].id)
            {
```

```
50                  if (d[i].hours < d[j].hours)
51                  {
52                      temp = d[j].hours;
53                      d[j].hours = d[i].hours;
54                      d[i].hours = temp;
55                  }
56              }
57          }
58      }
59  }
60
61
62  void Dancer::sort_by_rate(Dancer d[], int k)
63  {
64      float tempr;
65      for (int i = 0; i < 12; i++)
66      {
67          for (int j = i + 1; j < 12; j++)
68          {
69              if (d[i].id == d[j].id)
70              {
71                  if (d[i].rate > d[j].rate)
72                  {
73                      tempr = d[i].rate;
74                      d[i].rate = d[j].rate;
75                      d[j].rate = tempr;
76                  }
77              }
78          }
79      }
80  }
81
82  int Dancer::couple_counter(Dancer d[], int k)
83  {
84      int couples = 0;
85      for (int i = 0; i < 12; i++)
86      {
87          couples++;
88          if (d[i].id == d[i + 1].id)
89              couples--;
90      }
91      return couples;
92  }
93
94  void Dancer::u_id(Dancer d[], ofstream& outputfile)
95  {
96      int sponsers = 1;
97      float total = 0;
98      for (int i = 0; i < 12; i++)
```

```
 99      {
100              if (d[i].id == d[i+1].id)
101              {
102                  sponsers += 1;
103                  total += d[i].get_earnings(d, k:i);
104              }
105              else if (d[i].id != d[i + 1].id)
106              {
107                  total += d[i].get_earnings(d, k:i);
108                  outputfile << d[i].id << ": " << sponsers << " Sponsers $  ↵
                       " << total << endl;
109                  sponsers = 1;
110                  total = 0;
111              }
112      }
113  }
114
115  float Dancer::get_earnings(Dancer d[], int k)
116  {
117      return d[k].hours * d[k].rate;
118  }
119
120  int Dancer::get_id()
121  {
122      return id;
123  }
124
125  int Dancer::get_hrs()
126  {
127      return hours;
128  }
129
130  float Dancer::get_rate()
131  {
132      return rate;
133  }
134
135  float Dancer::total_raised(Dancer d[])
136  {
137      float total = 0;
138      for (int i = 0; i < 12; i++)
139      {
140          total += d[i].hours * d[i].rate;
141      }
142       return total;
143  }
144
145  void Dancer::swap(int k, int j, Dancer d[])
146  {
```

```
147
148 }
149
150 void Dancer::best_couple(Dancer d[], ofstream& outputfile)
151 {
152     float best = 0;
153     int bestsponsers;
154     int sponsers = 1;
155     int coupleid;
156     float total = 0;
157
158     for (int i = 0; i < 12; i++)
159     {
160         if (d[i].id == d[i + 1].id)
161         {
162             sponsers += 1;
163             total += d[i].get_earnings(d, k:i);
164         }
165         else if (d[i].id != d[i + 1].id)
166         {
167             total += d[i].get_earnings(d, k:i);
168             if (total > best)
169             {
170                 best = total;
171                 coupleid = d[i].id;
172                 bestsponsers = sponsers;
173                 sponsers = 1;
174                 total = 0;
175             }
176             if (!(total > best))
177             {
178                 sponsers = 1;
179                 total = 0;
180             }
181         }
182     }
183
184     outputfile << "Couple with the most money is : " << coupleid << " With ↵
        " << bestsponsers << " sponsers, earning $" << best << endl;
185 }
186
```