```cpp
/*
Matthew Hebert
A04817851
Project 6
*/

#include <iostream>
#include <string>
#include <fstream>
#include <iomanip>

using namespace std;

const int n = 3, numsem = 9, numyear = 3;

struct address
{
    string street;
    string city;
    string state;
    string zip;
};

struct semeval
{
    double eval[9];
};

struct comp_eval
{
    double compeval[3];
    double total;
    double average;
};

struct Employee
{
    string header[2];
    string e_name;
    string s_name;
    string e_id;
    string e_tele;
    address address;
    semeval e_evals;
    comp_eval comp_e_evals;
    double salary;
    double salraise;
    double salraiseindoll;
    double newsal;
    string letter[2];
    string note[2];
};


int inputData(ifstream &, ofstream &, int, Employee []);
int validateData(ofstream &, int, Employee []);
void FWE(int, Employee []);
void TAFWE(int, Employee []);
void SRinPercent(int, Employee []);
void Salary(int, Employee []);
void letter(ofstream &, int, Employee []);
void report(ofstream &, int, Employee []);

int main()
{
    ifstream fin;
```

```
    ofstream fout;
    fin.open("Project6_A04817851_inputfile.txt");
    fout.open("Project6_A04817851_outputfile.txt");
    Employee emp[n];
    int validatenum;

    for (int r = 0; r < n; r++)
    {
        validatenum = inputData(fin, fout, r, emp);
        if (validatenum == 1)
            continue;
        else if (validatenum == 2)
            break;
        FWE(r, emp);
        TAFWE(r, emp);
        SRinPercent(r, emp);
        Salary(r, emp);
        report(fout, r, emp);
    }
    fin.close();
    fout.close();
}


int inputData(ifstream &fin, ofstream &fout, int r, Employee emp[])
{
    int validate;

//get employee string info
    if(r == 0)
    {
        getline(fin, emp[r].letter[0]);
        getline(fin, emp[r].letter[1]);
    }

    getline(fin, emp[r].header[0]);
    getline(fin, emp[r].header[1]);
    getline(fin, emp[r].e_name);
    getline(fin, emp[r].s_name);
    getline(fin, emp[r].e_id);
    getline(fin, emp[r].e_tele);
    getline(fin, emp[r].address.street, ',');
    getline(fin, emp[r].address.city, ',');
    getline(fin, emp[r].address.state, ',');
    getline(fin, emp[r].address.zip);

//get employee evaluations
    for(int e = 0; e < numsem; e++)
    {
        fin >> emp[r].e_evals.eval[e];
    }

//get salary
    fin >> emp[r].salary;

//get discrepancy notes
    fin.ignore();
    getline(fin, emp[r].note[0]);
    getline(fin, emp[r].note[1]);

//validate data
    validate = validateData(fout, r, emp);
    if (validate == 1)
        return 1;
    else if (validate == 2)
        return 2;
```

```
}

int validateData(ofstream &fout, int r, Employee emp[])
{
//test warning length
    if (!(emp[0].letter[0].length() == 39))
    {
        cout << "ERROR. ";
        fout << "ERROR. PLEASE DO NOT EDIT WARNING MESSAGES." << endl;
        return 1;
    }

//test congrats length
    if (!(emp[0].letter[1].length() == 42))
    {
        cout << "ERROR. ";
        fout << "ERROR. PLEASE DO NOT EDIT CONGRATS MESSAGE." << endl;
        return 1;
    }

//test header lengths
    if (!(emp[r].header[0].length() == 34))
    {
        cout << "ERROR. ";
        fout << "ERROR. PLEASE DO NOT EDIT HEADERS." << endl;
        return 1;
    }

    if (!(emp[r].header[1].length() == 66))
    {
        cout << "ERROR. ";
        fout << "ERROR. PLEASE DO NOT EDIT HEADERS." << endl;
        return 1;
    }

//test user info lengths
    if (!(emp[r].e_name.length() > 1 && emp[r].e_name.length() < 50))
    {
        cout << "ERROR. ";
        fout << "ERROR. PLEASE ENTER A VALID EMPLOYEE NAME." << endl;
        return 1;
    }

    if (!(emp[r].s_name.length() > 1 && emp[r].s_name.length() < 50))
    {
        cout << "ERROR. ";
        fout << "ERROR. PLEASE ENTER A VALID SUPERVISOR NAME." << endl;
        return 1;
    }

    if (!(emp[r].e_id.length() > 1 && emp[r].e_id.length() < 20))
    {
        cout << "ERROR. ";
        fout << "ERROR. PLEASE ENTER A VALID EMPLOYEE ID." << endl;
        return 1;
    }

    if (!(emp[r].e_tele.length() > 1 && emp[r].e_tele.length() < 20))
    {
        cout << "ERROR. ";
        fout << "ERROR. PLEASE ENTER A VALID EMPLOYEE TELEPHONE NUMBER." << endl;
        return 1;
    }

    if (!(emp[r].address.street.length() > 1 && emp[r].address.street.length() < 33))
    {
```

```
199         cout << "ERROR. ";
200         fout << "ERROR. PLEASE ENTER A VALID STREET." << endl;
201         return 1;
202     }
203 
204     if (!(emp[r].address.city.length() > 1 && emp[r].address.city.length() < 20))
205     {
206         cout << "ERROR. ";
207         fout << "ERROR. PLEASE ENTER A VALID CITY." << endl;
208         return 1;
209     }
210 
211     if (!(emp[r].address.state.length() > 1 && emp[r].address.state.length() < 15))
212     {
213         cout << "ERROR. ";
214         fout << "ERROR. PLEASE ENTER A VALID STATE." << endl;
215         return 1;
216     }
217 
218     if (!(emp[r].address.zip.length() > 1 && emp[r].address.zip.length() < 10))
219     {
220         cout << "ERROR. ";
221         fout << "ERROR. PLEASE ENTER A VALID ZIP." << endl;
222         return 1;
223     }
224 
225 //test evaluation values
226     for(int g = 0; g < numsem; g++)
227     {
228         if (!(emp[r].e_evals.eval[g] > 0 && emp[r].e_evals.eval[g] < 150))
229         {
230             cout << "ERROR. ";
231             fout << "ERROR. PLEASE ENTER A VALID GRADE FOR SEMESTER " << g+1;
232             return 2;
233         }
234     }
235 
236 //test salary values
237     if(emp[r].salary < 0)
238     {
239         cout << "ERROR. ";
240         fout << "ERROR. PLEASE ENTER A VALID SALARY.";
241         return 2;
242     }
243 
244 //test discrepancy note lengths
245     if(!(emp[r].note[0].length() > 85 && emp[r].note[0].length() < 150))
246     {
247         cout << "ERROR. ";
248         fout << "ERROR. PLEASE VALIDATE THE DISCREPANCY NOTE.";
249         return 1;
250     }
251     if(!(emp[r].note[1].length() > 59 && emp[r].note[1].length() < 125))
252     {
253         cout << "ERROR. ";
254         fout << "ERROR. PLEASE VALIDATE THE DISCREPANCY NOTE.";
255         return 1;
256     }
257 }
258 
259 void FWE(int r, Employee emp[])
260 {
261 // declare weights
262     const double fallw = 0.39, springw = 0.37, summerw = 0.24;
263 
264 //calculate final weighted evaluations
```

```
265     emp[r].comp_e_evals.compeval[0] = (emp[r].e_evals.eval[0]*springw)
266                                       +(emp[r].e_evals.eval[1]*summerw)
267                                       +(emp[r].e_evals.eval[2]*fallw);
268
269     emp[r].comp_e_evals.compeval[1] = (emp[r].e_evals.eval[3]*springw)
270                                       +(emp[r].e_evals.eval[4]*summerw)
271                                       +(emp[r].e_evals.eval[5]*fallw);
272
273     emp[r].comp_e_evals.compeval[2] = (emp[r].e_evals.eval[6]*springw)
274                                       +(emp[r].e_evals.eval[7]*summerw)
275                                       +(emp[r].e_evals.eval[8]*fallw);
276
277
278 }
279
280 void TAFWE(int r, Employee emp[])
281 {
282 //calculate tfwe
283     emp[r].comp_e_evals.total = emp[r].comp_e_evals.compeval[0]
284                                +emp[r].comp_e_evals.compeval[1]
285                                +emp[r].comp_e_evals.compeval[2];
286 //calculate afwe
287     emp[r].comp_e_evals.average = emp[r].comp_e_evals.total/3;
288
289 }
290
291 void SRinPercent(int r, Employee emp[])
292 {
293 //calculate sr in percent
294     if (emp[r].comp_e_evals.average < 75)
295         emp[r].salraise = 0;
296     else if (emp[r].comp_e_evals.average > 75 && emp[r].comp_e_evals.average <= 80)
297         emp[r].salraise = 1;
298     else if (emp[r].comp_e_evals.average > 80 && emp[r].comp_e_evals.average <= 90)
299         emp[r].salraise = 3;
300     else if (emp[r].comp_e_evals.average > 90 && emp[r].comp_e_evals.average <= 100)
301         emp[r].salraise = 5;
302     else
303         emp[r].salraise = 10;
304 }
305
306 void Salary(int r, Employee emp[])
307 {
308 //calculate sr in dollars
309     emp[r].salraiseindoll = (emp[r].salraise * 0.01) * emp[r].salary;
310
311 //calculate salary with raise
312     emp[r].newsal = emp[r].salraiseindoll + emp[r].salary;
313
314 }
315
316 void letter(ofstream &fout, int r, Employee emp[])
317 {
318 //write warning / congrats messages
319     if (emp[r].comp_e_evals.average < 70)
320         fout << endl << emp[0].letter[0] << endl;
321     else if (emp[r].comp_e_evals.average >= 95)
322         fout << endl << emp[0].letter[1] << endl;
323 }
324
325 void report(ofstream &fout, int r, Employee emp[])
326 {
327     fout << setw(50) << emp[r].header[0] << endl;
328     fout << emp[r].header[1] << endl << endl;
329     fout << setw(34) << "Name of the Employee:" << setw(9) << " " << emp[r].e_name << endl;
330     fout << setw(34) << "Name of the Supervisor:" << setw(9) << " " << emp[r].s_name << endl;
```

```
331      fout << setw(34) << "Employee ID:" << setw(9) << " " << emp[r].e_id << endl;
332      fout << setw(34) << "Telephone Number:" << setw(9) << " " << emp[r].e_tele << endl;
333      fout << setw(34) << "Address:" << setw(9) << " " << emp[r].address.street << ", " <<
emp[r].address.city << ", " << emp[r].address.state << ", " << emp[r].address.zip << endl;
334
335  //write evaluations
336      for (int o = 0; o < numyear; o++)
337      {
338          if (o == 0)
339          {
340              fout << setw(34) << "Spring Semester Evaluation, 2011:" << setw(9) << " " << setprecision(2) <<
fixed << emp[r].e_evals.eval[0] << endl;
341              fout << setw(34) << "Summer Semester Evaluation, 2011:" << setw(9) << " " <<
emp[r].e_evals.eval[1] << endl;
342              fout << setw(34) << "Fall Semester Evaluation, 2011:" << setw(9) << " " <<
emp[r].e_evals.eval[2] << endl;
343          }
344          else if (o == 1)
345          {
346              fout << setw(34) << "Spring Semester Evaluation, 2012:" << setw(9) << " " <<
emp[r].e_evals.eval[3] << endl;
347              fout << setw(34) << "Summer Semester Evaluation, 2012:" << setw(9) << " " <<
emp[r].e_evals.eval[4] << endl;
348              fout << setw(34) << "Fall Semester Evaluation, 2012:" << setw(9) << " " <<
emp[r].e_evals.eval[5] << endl;
349          }
350          else if (o == 2)
351          {
352              fout << setw(34) << "Spring Semester Evaluation, 2013:" << setw(9) << " " <<
emp[r].e_evals.eval[6] << endl;
353              fout << setw(34) << "Summer Semester Evaluation, 2013:" << setw(9) << " " <<
emp[r].e_evals.eval[7] << endl;
354              fout << setw(34) << "Fall Semester Evaluation, 2013:" << setw(9) << " " <<
emp[r].e_evals.eval[8] << endl;
355          }
356      }
357
358  //write computed evaluations
359      fout << setw(34) << "Final Weighted Evaluation, 2011:" << setw(9) << " " <<
emp[r].comp_e_evals.compeval[0] << endl;
360      fout << setw(34) << "Final Weighted Evaluation, 2012:" << setw(9) << " " <<
emp[r].comp_e_evals.compeval[1] << endl;
361      fout << setw(34) << "Final Weighted Evaluation, 2013:" << setw(9) << " " <<
emp[r].comp_e_evals.compeval[2] << endl;
362      fout << setw(34) << "Total Final Weighted Evaluation:" << setw(9) << " " << emp[r].comp_e_evals.total
<< endl;
363      fout << setw(34) << "Average Final Weighted Evaluation:" << setw(9) << " " <<
emp[r].comp_e_evals.average << endl;
364      fout << setw(34) << "Current Salary:" << setw(9) << " " << "$" << emp[r].salary << endl;
365      fout << setw(34) << "Salary Raise in %:" << setw(9) << " " << emp[r].salraise << "%" << endl;
366      fout << setw(34) << "Salary Raise in Dollars:" << setw(9) << " " << "$" << emp[r].salraiseindoll <<
endl;
367      fout << setw(34) << "Salary in Dollars with Raise:" << setw(9) << " " << "$" << emp[r].newsal << endl;
368
369  //write warning/congrats message
370      letter(fout, r, emp);
371
372  //write discrepancy note
373      fout << endl << emp[r].note[0] << endl;
374      fout << "\t" << emp[r].note[1] << endl << endl << endl;
375  }
```