#include <iostream>

using namespace std;

union worker

{

struct Tex

{

int number, experience, salary, prof, e;

char FIO[3];

enum Education

{

SecondaryProf,

Higher

} education;

} T;

struct Ped

{

int number, experience, salary, pro, s, z;

char FIO[3];

enum Stepen

{

Candidate,

Doctor,

No

} stepen;

enum Zvanie

{

Docent,

Professor,

Net

} zvanie;

} P;

};

int main()

{

setlocale(LC\_ALL, "Russian");

string c, speciality;

int rab = -1, zp\_tex = 0, count\_tex = 0, count\_ped = 0, t = 10, p = 20;

const int razmer = 5;

worker massiv[razmer];

while (1)

{

cout << endl << "Введите: \nexit - для выхода из программы \nadd - добавить сведения о сотруднике \nall\_worker - сведения обо всех сотрудниках ВУЗа \ncount\_tex - количество инженерно-технических работников \ncount\_ped - количество научно-педогагических работников \nall\_doctor - сведения обо всех докторах наук \nsalary\_tex - фонд зарплат инженерно-технических работников\n" << endl;

cin >> c;

if (c == "exit")

{

return 0;

}

else if (c == "add")

{

rab += 1;

cout << "Выберите: \ntex - Нучно-технический работник \nped - Научно-педагогический работник\n";

cin >> speciality;

if (speciality == "tex")

{

massiv[rab].T.prof = t;

cout << "Введите табельный номер\n";

cin >> massiv[rab].T.number;

cout << "Введите ФИО (3 буквы без пробела)\n";

cin >> massiv[rab].T.FIO;

cout << "Введите стаж\n";

cin >> massiv[rab].T.experience;

cout << "Введите образование (sr - среднее профессиональное / vis - высшее)\n";

string obrazovanie;

cin >> obrazovanie;

if (obrazovanie == "sr")

{

massiv[rab].T.e = worker::Tex::Education::SecondaryProf;

}

else if (obrazovanie == "vis")

{

massiv[rab].T.e = worker::Tex::Education::Higher;

}

cout << "Введите оклад\n";

cin >> massiv[rab].T.salary;

//для salary\_tex (з/п инж-тех работников)

zp\_tex += massiv[rab].T.salary;

//для count\_tex (кол-во инж-тех работников)

count\_tex += 1;

}

if (speciality == "ped")

{

massiv[rab].P.pro = p;

cout << "Введите табельный номер\n";

cin >> massiv[rab].P.number;

cout << "Введите ФИО (3 буквы без пробела)\n";

cin >> massiv[rab].P.FIO;

cout << "Введите стаж\n";

cin >> massiv[rab].P.experience;

cout << "Введите учёную степень (kan - кандидат наук / doc - доктор наук / no - отсутствует)\n";

string st;

cin >> st;

if (st == "kan")

{

massiv[rab].P.s = worker::Ped::Stepen::Candidate;

}

else if (st == "doc")

{

massiv[rab].P.s = worker::Ped::Stepen::Doctor;

}

else if (st == "no")

{

massiv[rab].P.s = worker::Ped::Stepen::No;

}

cout << "Введите учёное звание ( doz - доцент / pr - профессор / no - отсутствует )\n";

string zv;

cin >> zv;

if (zv == "doz")

{

massiv[rab].P.z = worker::Ped::Zvanie::Docent;

}

else if (zv == "pr")

{

massiv[rab].P.z = worker::Ped::Zvanie::Professor;

}

else if (zv == "no")

{

massiv[rab].P.z = worker::Ped::Zvanie::Net;

}

cout << "Введите оклад\n";

cin >> massiv[rab].P.salary;

//для count\_ped (кол-во науч-пед работников)

count\_ped += 1;

}

}

else if (c == "all\_worker")

{

for (int i = 0; i < rab + 1; i++)

{

if (massiv[i].T.prof == t)

{

cout << massiv[i].T.number << endl << massiv[i].T.FIO << endl << massiv[i].T.experience << endl << massiv[rab].T.e << endl << massiv[i].T.salary << endl << endl;

}

else if (massiv[i].P.pro == p)

{

cout << massiv[i].P.number << endl << massiv[i].P.FIO << endl << massiv[i].P.experience << endl << massiv[i].P.s << endl << massiv[i].P.z << endl << massiv[i].P.salary << endl << endl;

}

}

}

else if (c == "count\_tex")

{

cout << endl << count\_tex << endl;

}

else if (c == "count\_ped")

{

cout << endl << count\_ped << endl;

}

else if (c == "all\_doctor")

{

for (int i = 0; i < rab + 1; i++)

{

if (massiv[i].P.pro == p)

{

if (massiv[i].P.s == worker::Ped::Stepen::Doctor)

{

cout << massiv[i].P.number << endl << massiv[i].P.FIO << endl << massiv[i].P.experience << endl << massiv[i].P.s << endl << massiv[i].P.z << endl << massiv[i].P.salary << endl << endl;

}

}

}

}

else if (c == "salary\_tex")

{

cout << endl << zp\_tex << endl;

}

else

{

cout << endl << "Такой команды нет\n";

}

}

return 0;

}