

```

public class Trooper extends Soldier{

    private int nbYears;

    public Trooper(String name, String rank, int nbYears)
    {
        super(name,rank);
        this.nbYears = nbYears;
    }

    public Trooper(Trooper t) {
        super(t);
        nbYears = t.nbYears;
    }

    public double calculateAward() {
        return nbYears * 2000;
    }

    public int getNbYears() {

        return nbYears;
    }

}

```

```

public class Platoon {
    private String name;
    private Person arp[];
    private int nb;

    Platoon(String name)
    {
        this.name = name;
        arp = new Person[30];
        nb = 0;
    }

    public void addPrson(Person p)
    {
        if(nb >= arp.length)
            return;

        if(p instanceof Officer)
            arp[nb] = new Officer((Officer)p);
        else
            arp[nb] = new Trooper((Trooper)p);
        nb++;
    }
}

```

```

public int countOfficer(double s)
{
    int count = 0;
    for(int i = 0; i<nb ; i++)
        if(arp[i] instanceof Officer)
            if(arp[i].calculateAward() >= s)
                count++;
    return count;
}

public Trooper[] getTroopers(int n)
{
    Trooper[] t = new Trooper[nb];
    int j = 0;
    for(int i=0; i<nb ; i++)
    {
        if(arp[i] instanceof Trooper)
        {
            Trooper x = (Trooper)arp[i];
            if(x.getNbYears() > n)
            {
                t[j] = x;
                j++;
            }
        }
    }
    return t;
}
}

```

---

Not required classes:

```

public interface Person {

    double calculateAward();
    String getName();

}

```

```

public abstract class Soldier implements Person
{
    private String name;
    private String rank;

    public Soldier(String name, String rank) {
        this.name = name;
        this.rank = rank;
    }

    public Soldier(Soldier s) {
        name = s.name;
        rank = s.rank;
    }
}

```

```
    }

    public String getName() {
        return name;
    }
}

public class Officer extends Soldier{

    public Officer(String name, String rank)
    {
        super(name,rank);
    }
    public Officer(Officer o) {
        super(o);
    }

    public double calculateAward() {
        return 50000;
    }
}
```