Exercises 2.3

1) Develop a structure to represent the planets in the solar system. Each planet has fields for the planet's name, its distance from the sun (in miles), and the number of moons it has. Place items in each of the fields for the planets: Earth and Venus.

```
typedef struct {
        char name[30];
        double sun_dist ;   // in miles.
        int moon_no ;
    } planet ;
```

```
Planet Earth, Venus ;
Earth.name= "Earth";
Earth.sun_dist = 93x10^6;
Earth. moon_no = 1;
Venus.name= "Venus";
Venus.sun_dist= 67x10^6;
Venus.moon_no = 0
```

2) Modify the humanBeing structure so that we can include different information based on marital status. Marital status should be an enumerated type with fields: single, married, widowed, divorced. Use a union to include different information based on marital status as follows:

- Single : No information needed
- Married : Include a married date field
- Widowed : Include marriage date and death of spouse date fields
- Divorced : Include divorce date and no. of divorces field.

Assign values to the fields for some person of type humanBeing.

```
typedef struct {
    char name[10]; int age; float salary; date dob; sexType sexInfo;
  enum tagField {Single, Married, Widowed, Divorced}
  union {
      date marriage; struct{ date marriage; date spouse_death; }widow;
  struct {date divorce; int divorce_no} divorced; }u; } humanBeing;
```