Question 3

1. Would not unify since Food has a capital F (syntax error)
2. Would unify by assigning the atom **soup** to **Bread**.
3. Would unify and assign **Soup** to **Bread**.
4. Would not unify because **X** would need to take two values at once.
5. Would unify by assigning manager(**X**) **Y**.
6. Would unify and assign healthyFood(bread) to **X** and drink(milk) to **Y**.
7. Would not unify because it is trying to see if the list equals the predicate meal.
8. Would unify and assign an empty list to **Z**, eat([ ]) **X** and drink(milk) to **Y**.
9. Would unify and assign l to **X,** and b to **Z.**
10. Would unify and assign french(jean) to **A** and scottish(joe) to **B**.
11. Would unify and assign healthyFood(bread) to **X** and drink(water) to **Y**.
12. Would unify and assign a to **H** and [b, c] to **T**.
13. Would not unify because it is comparing a list of length 2 to a list of length 3.
14. Would not unify because it would try to assign bread and milk to **T**.
15. Would unify and assign jack to **X**, cook(egg, oil) to **Y** and **Evening** to **Time**.
16. Would unify and assign s(g) to **X** and t(k) to **Y**.
17. Would not unify because it would try to assign **D** to the atom x and 17.
18. Would not unify because it is trying to equate the atom b to the list [**H**|**T**].

Question 4

1. Ground query, will unify $\mathrm{field}(\mathrm{hit\_transfer}, \mathrm{engineering})$ to $\mathrm{field}(\mathbf{X}, \mathbf{Y})$. It will then resolve into $\mathrm{course}(\mathrm{hit\_transfer}, \mathbf{Z}), field(\mathbf{Z}, \boldsymbol{engineering})$ and find that $\mathbf{Z}$ is mechanical by unifying $course(hit\_transfer, \mathbf{Z})$ to $course(hit\_transfer, mechanical)$ and $field(\mathbf{Z}, \boldsymbol{engineering})$ **to** $\boldsymbol{field}(\boldsymbol{mechanical}, \boldsymbol{engineering})$, and return true.

2. Non-ground, will unify $lab\_number(\mathrm{fine\_arts}, \mathbf{X})$ with $\mathrm{lab\_number}(\mathbf{fine\_arts}, 10)$, and return $\mathbf{X} = \mathbf{10}$.

3. Ground query, will return false.

4. Non-ground query, will unify and return all given course pairs, so for $\mathbf{X}$ and $\mathbf{Y}$ respectively, we have $(hit\_transfer, mechanical), (web\_design, computer), (design\_methods, fine - arts), (poetry, literature)$,
$(leadership, management)$ and $(biology, medicin)$.

5. Ground query, will unify $student(adrian)$ with $student(\mathbf{X})$, resolve to $student(adrian, \_)$, unifying that with $student(adrian, web\_design)$ and return true.

6. Ground query, will unify $student(anna, engineering)$ with $student(\mathbf{X}, \mathbf{Y})$, resolving to $field(\mathbf{Z}, \mathrm{engineering}), student(anna, \mathbf{Z})$, and unifying $field(\mathbf{Z}, \mathrm{engineering})$ to $field(mechanical, enigneering)$ and $student(anna, hit\_transfer)$, returning true.

7. Non-ground, will unify $student(\mathbf{X}, \mathbf{engineering})$ with $student(\mathbf{X}, \mathbf{Y})$, resolving to $field(\mathbf{Z}, \mathbf{engineering}), \mathbf{student}(\mathbf{X}, \mathbf{Z})$. The chain of unifications is long, so I will only give one example.

   $\mathbf{student}(\mathbf{X}, \mathbf{Z})$ will unify with $student(anna, hit\_transfer)$, assigning $hit\_transfer$ to $\mathbf{Z}$. Then, $field(hit\_transfer, engineering)$ will unify with $field(\mathbf{X}, \mathbf{Y})$, resolving into $course(hit\_transfer, \mathbf{Z}), \boldsymbol{field}(\mathbf{Z}, \boldsymbol{engineering})$. $\boldsymbol{course}(\boldsymbol{hit\_transfer}, \mathbf{Z})$ will unify with $\boldsymbol{course}(\boldsymbol{hit\_transfer}, \boldsymbol{mechanical})$ and $\boldsymbol{field}(\mathbf{Z}, \mathbf{engineering})$ will unify with $field(mechanical, engineering)$, thus proving that the query returns true when $\mathbf{X} = \mathbf{anna.}$ Similar process is done for $daniel$ and $adrian$, making similar unifications and returning $anna, daniel, adrian.$

8. Non-ground, will return false because there is no possible value for $\mathbf{Y}$ for which $course(fine\_arts, \mathbf{Y})$ is true, nor is there a value for $\mathbf{X}$ for which $student(\mathbf{X}, \mathbf{fine} - \mathbf{arts})$ is true.

9. Non-ground, will return the following values for $\mathbf{X}$: $\mathbf{engineering}, \mathbf{engineering}, \mathbf{art}, \mathbf{social}, \mathbf{business}, \mathbf{engineering}, \mathbf{engineering}, \mathbf{art}, \mathbf{social}, \mathbf{business}$. It does so by unifying all the $field$ statements, and also unifying all the $course$ statements, except for $course(biology, medicin)$ because it does not have a defined field.

10. Non-ground, will return false because $\mathbf{X}$ will be assigned to a number.
11. Non-ground, will unify $lab\_number(\mathbf{X}, \mathbf{15})$ to $lab\_number(mechanical, 15)$, and then $field(mechanical, \mathbf{Y})$ to $field(mechanical, engineering)$, returning $\mathbf{X\ =\ mechanical}$ and $\mathbf{Y\ =\ engineering.}$

12. Non-ground, will resolve $student(\mathbf{X})$ to $student(\mathbf{X}, \_)$ which unifies to $student(anna, hit\_transfer)$. The same process happens with $student(\mathbf{X}, \_)$, but since there is the cut operator (!), it only resolves the pair once and returns that both $\mathbf{X}$s are $anna$.

13. Non-ground, will resolve similarly to 12, but because the cut is at the end, it will only return one $\mathbf{X}$, being $anna$.

14. Non-ground. Will resolve $course(\mathbf{X}, \_)$ to $course(biology, medicin)$ and will fail to resolve $student(\_, \mathbf{X})$ with anything, resulting in returning $\mathbf{X} = \mathbf{biology.}$