Anthony Tran
Axt220037
CS 4337-503

Assignment 4

Code is in corresponding Q folders. (some too long to submit screenshots)

1. Query results. Siblings defined as having the same 2 parents. Uncles and aunt inlaws are not considered.

   a. Kathleen

   > *mother*(X, maisy).

   X = kathleen

   Next | 10 | 100 | 1,000 | Stop

   ```
   ?-   mother(X, maisy).
   ```

   b. Naomi and Finnegan

   > *sister*(X, maisy)

   X = naomi
   X = finnegan

   Next | 10 | 100 | 1,000 | Stop

   ```
   ?-   sister(X, maisy)
   ```

   c. No

   > *brother*(X, maisy)

   false

   ```
   ?-   brother(X, maisy)
   ```

   d. No brothers

   > *brother*(X, maisy)

   false

   ```
   ?-   brother(X, maisy)
   ```

e. Neila

> grandmother(X, maisy)

X = neila
false

?- `grandmother(X, maisy).`

f. Joe

> grandfather(X, maisy).

X = joe

[Next] [10] [100] [1,000] [Stop]

?- `grandfather(X, maisy).`

g. Ashley and naomi2

> aunt(X, maisy).

X = ashley
X = naomi2

[Next] [10] [100] [1,000] [Stop]

?- `aunt(X, maisy).`

h. Beau

> uncle(X, maisy).

X = beau

[Next] [10] [100] [1,000] [Stop]

?- `uncle(X, maisy).`

i. joe_sr

> greatgrandfather(X, maisy).

X = joe_sr

[Next] [10] [100] [1,000] [Stop]

?- `greatgrandfather(X, maisy).`

greatgreatgrandmother(X, maisy).

false

?-   greatgreatgrandmother(**X**, maisy).

i.

greatgreatgrandfather(X, maisy).

false

Predicate defined in line 84

?-   greatgreatgrandfather(**X**, maisy).

? -   reversea ([a , b , c , d], Q).   append ([d , c , b] , [a] , Q) → $Q = [d , c , b , a]$

   ↳ H = a , T = [b , c , d]

    reversea([b , c , d] , Q).   append ([d , c] , [b] , R) → R = [d , c , b]

    ↳ H = b , T = [c , d]

     reverse a([c , d] , Q).   append ([d] , [c] , n) → R = [d , c]

     ↳ H = c , T = d

      reversea([d] , Q).   append ([ ] , [d] , R) → R = [d]

      H = d   T =

      ↳ reversea ([ ] , Q).

       reverse a ([ ] , [ ]). → [ ]   base case

2.

reversea([a,b,c,d],Q).

Q = [d, c, b, a]

?-   reversea([a,b,c,d],**Q**).

nelements([1,2,3,4,5,6,7], Q).

Q = 12

| Next | 10 | 100 | 1,000 | Stop |

?-   nelements([1,2,3,4,5,6,7], **Q**).

3.

> **delements**([a, [b,c], d, e, f], Q)

Q = [d, e, f]

```
?-   delements([a, [b,c], d, e, f], Q)
```

4.

5. Not sure

6. Query results.

   a. 35 and over

> **smallestFirstYearGroup**(Age)

Age = 35

```
?-   smallestFirstYearGroup(Age)
```

   b. 15-19 has largest number of 1st years

> **largestFirstYearGroup**(Age)

Age = 1519

| Next | 10 | 100 | 1,000 | Stop |

```
?-   largestFirstYearGroup(Age)
```

   c. White_alone_non_hispanic

> **largest_race**(Race, Count)

Count = 6923,
Race = white_alone_non_hispanic

```
?-   largest_race(Race, Count)
```

   d. 2793 full time employed students

> **full_time_employ_count**(Total)

Total = 2793

```
?-   full_time_employ_count(Total)
```

e. 13278 total students

🌞 *total_students*(Total)　　　　　　　　　　　　　　　　　　⊕ ▬ ⊗

Total = 13278

?- `total_students(`**`Total`**`)`

---

f. 4128 students unemployed

🌞 *first_and_second_unemployed*(Total)　　　　　　　　　　　⊕ ▬ ⊗

Total = 4128

?- `first_and_second_unemployed(`**`Total`**`)`

---

g. 9393 students

🌞 *older_than_twenty*(Total)　　　　　　　　　　　　　　　　⊕ ▬ ⊗

Total = 9393

?- `older_than_twenty(`**`Total`**`)`

---

h. 8237 students

🌞 *older_than_twenty_but_under_thirtyfive*(Total)　　　　　⊕ ▬ ⊗

Total = 8237

?- `older_than_twenty_but_under_thirtyfive(`**`Total`**`)`

---

i. 438 disabilities

🌞 *disability_count_no_four_years*(Total)　　　　　　　　　　⊕ ▬ ⊗

Total = 438

?- `disability_count_no_four_years(`**`Total`**`)`

j. 990 asians

> asian_count(Total)
>
> **Total** = 990

```
?-  asian_count(Total)
```

k. 3912 asian or hispanic

> asian_and_hispanic(Total)
>
> **Total** = 3913

```
?-  asian_and_hispanic(Total)
```

l. 12288 non asians

> not_asian(Total)
>
> **Total** = 12288

```
?-  not_asian(Total)
```

m. Only have statistics on asian-alone so don't know if any asians are hispanic