Assignment 3:

So in this assignment we were given a simple puzzle and we had to implement a program to generate such a puzzle.

I used backtracking in the first part of the assignment as I had to search in all the domains for the correct variable that satisfies the constraint, and if during this evaluation I found something wrong or inconsistent, I back track and change the variable I have used.

The code for such a problem was not a problem and it was easy as the size of the problem was small regarding the number of variables and domains for each of the variables.

Upon running my java code for the first part of this assignment I got:

H | O | S | E | S

---+---+---+---+---

# | # | A | # | T

---+---+---+---+---

# | H | I | K | E

---+---+---+---+---

A | # | L | E | E

---+---+---+---+---

L | A | S | E | R

---+---+---+---+---

E | # | # | L | #

And the time to generate such a cross word was in less than half a second.

Professor suggested that there is no way to generate the Bigger puzzle In Task 2 using the same backtrack algorithm I used in the first task for several reasons. Most importantly the number of words to be placed on the board(Crossword puzzle is way bigger) and also the domain is more than 12000 words. Which is technically impossible and it will take days to execute such a code.

How to deal with CSP problems:

In the lectures there were more than one technique which I tried to implement to solve that problem, Forward checking which is that we will assign variables until we reach a point that one of the variables will have no values to be assigned to and from there we will back track, This technique mixed by starting to explore the words that has the Most Constraints is what I tried to reflect in my implementation.

I tried more than implementation using Java to satisfy these constraint but I found it hectic and did not reach a solution, I tried to implement it in python because there were more than one tutorial online and in fact I reached something but I did not reach the desire output.

Upon running my python code this is what I have reached.

V S R U

POSIT SENNA

ELITE PLEBS

RA ENVOY UP

NUTS O U STEP

SKIP R T ETCH

ELAN SPOT

ERST SPAN

ETERNAL

It kind of assign the variables to words for example word 7 is POSIT and word 11 is ELITE and so on.

I understood quite well what I should do to obtain the solution and that back tracking will never work with such problems, On the contrary I will have to do better AI and start with the variables that have most constraints and try to forward check.