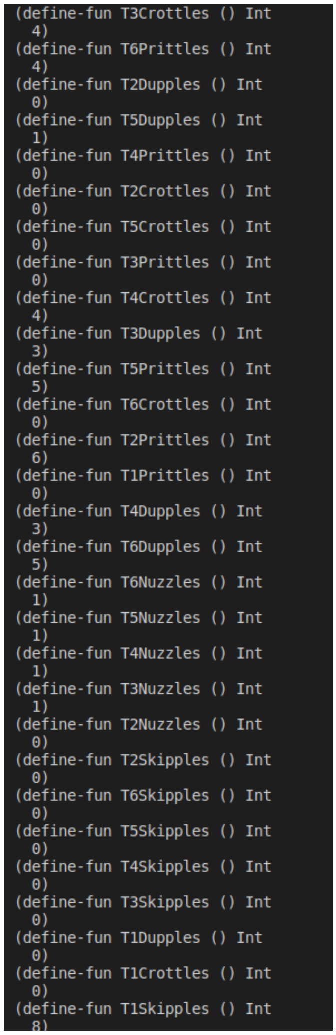
Assignments week 5

Aleksandar Ivanov, Ivaylo Ivanov

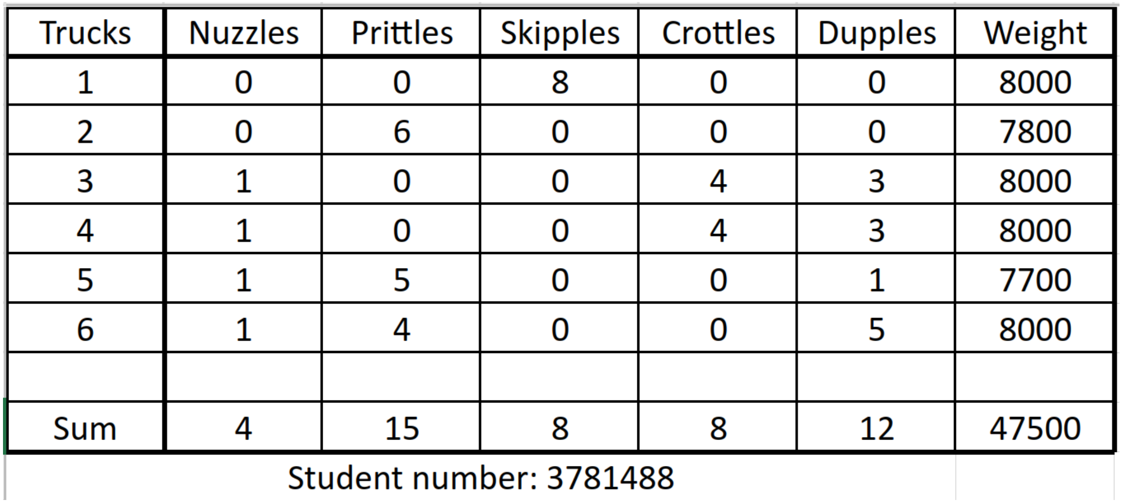
12.01.2021

Assignment 1:

For this assignment we simplified our approach by having different variables for all pallets of all trucks – this made the code way more readable and easy to explain. We added functions that get us the sum of all types of pallets as well as other helper functions for the rules. The last line required the prittles to be maximized, which was the main point of the equation. We used Aleksandar's student number – 3781488, which required no prittles and crottles in the same truck.



```
(define-fun T3Crottles () Int
  4)
(define-fun T6Prittles () Int
  4)
(define-fun T2Dupples () Int
  0)
(define-fun T5Dupples () Int
  1)
(define-fun T4Prittles () Int
  0)
(define-fun T2Crottles () Int
  0)
(define-fun T5Crottles () Int
  0)
(define-fun T3Prittles () Int
  0)
(define-fun T4Crottles () Int
  4)
(define-fun T3Dupples () Int
  3)
(define-fun T5Prittles () Int
  5)
(define-fun T6Crottles () Int
  0)
(define-fun T2Prittles () Int
  6)
(define-fun T1Prittles () Int
  0)
(define-fun T4Dupples () Int
  3)
(define-fun T6Dupples () Int
  5)
(define-fun T6Nuzzles () Int
  1)
(define-fun T5Nuzzles () Int
  1)
(define-fun T4Nuzzles () Int
  1)
(define-fun T3Nuzzles () Int
  1)
(define-fun T2Nuzzles () Int
  0)
(define-fun T2Skipples () Int
  0)
(define-fun T6Skipples () Int
  0)
(define-fun T5Skipples () Int
  0)
(define-fun T4Skipples () Int
  0)
(define-fun T3Skipples () Int
  0)
(define-fun T1Dupples () Int
  0)
(define-fun T1Crottles () Int
  0)
(define-fun T1Skipples () Int
  8)
```

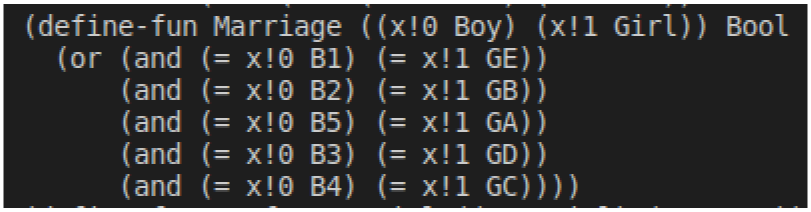


Trucks	Nuzzles	Prittles	Skipples	Crottles	Dupples	Weight
1	0	0	8	0	0	8000
2	0	6	0	0	0	7800
3	1	0	0	4	3	8000
4	1	0	0	4	3	8000
5	1	5	0	0	1	7700
6	1	4	0	0	5	8000
Sum	4	15	8	8	12	47500

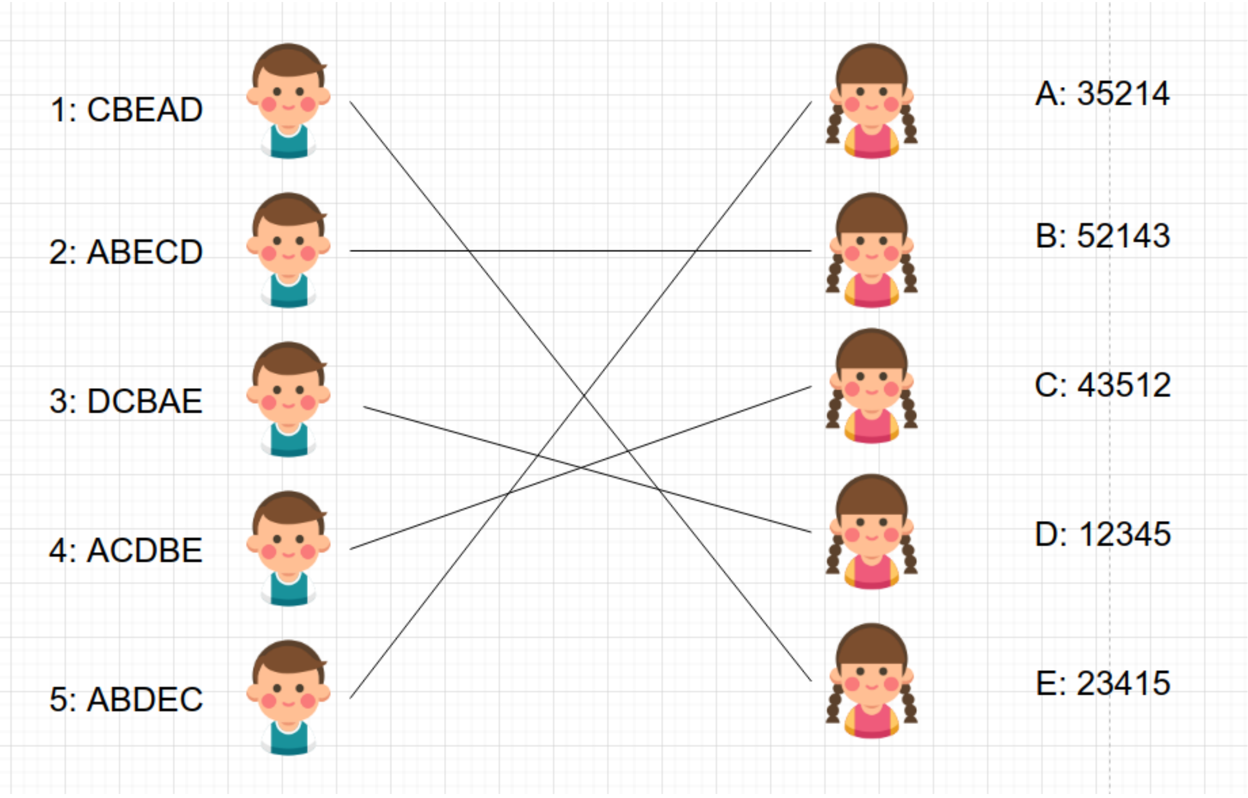
Student number: 3781488

Assignment 2:

For the stable marriages assignment we used the template provided in sharepoint, added a reverse function that gives us someone's preference as an integer, so we can easily use it in equations/inequalities and made sure that when a marriage exists, if any of the partners (X) prefer someone else (Y), then that someone else (Y) is already married to a fourth person (Z) who is preferred over the partner in question (X).



```
(define-fun Marriage ((x!0 Boy) (x!1 Girl)) Bool
  (or (and (= x!0 B1) (= x!1 GE))
      (and (= x!0 B2) (= x!1 GB))
      (and (= x!0 B5) (= x!1 GA))
      (and (= x!0 B3) (= x!1 GD))
      (and (= x!0 B4) (= x!1 GC))))
```


1: CBEAD
2: ABECD
3: DCBAE
4: ACDBE
5: ABDEC
A: 35214
B: 52143
C: 43512
D: 12345
E: 23415