**DECISION ANALYTICS**

Assignment 1: Constraint Programming

Student Name: Gyanendar Manohar

Student Id: R00207241

TASK#1

*Object domain: person(james,daniel,emily,sophie)*

*predicates: starter, main course, drinks, desert*

|  |
| --- |
| A: Objects, Attributes & predicates line [128-144] |
|  |
| Line [153-185] |
|  |
| B: Explicit constraints line [291-410] |
|  |
|  |
|  |
| C: Implicit Constraint Line [187-290] |
|  |
|  |
|  |
| Solution Printer Line [17-41] |
|  |
| Output: |
| -----------------Question#1 Dinner Puzzel-----------------------  solution 1  ***- James:***  ***- onion\_soup***  ***- filet\_steak***  ***- coke***  ***- apple\_crumble***  ***- Daniel:***  ***- carpaccio***  ***- fried\_chicken***  ***- beer***  ***- chocolate\_cake***  ***- Emily:***  ***- mushroom\_tart***  ***- vegan\_pie***  ***- red\_wine***  ***- ice\_cream***  ***- Sophie:***  ***- prawn\_cocktail***  ***- baked\_mackerel***  ***- white\_wine***  ***- tiramisu***  OPTIMAL |
| **problem\_1() is wrapper for calling all the needed function and displaying the result (optimal solution)**  Line [854-864] |
|  |
| Line [416-427] |
|  |
|  | |

TASK#2 Soduku

|  |
| --- |
| A: Decision variables line[442-455] |
|  |
| B: Implement the constraints that specify the digits, which are already given in the puzzle |
| Line [534-552] |
|  |
| C: Constraint on row, column, and squares |
| Line[457-499] |
|  |
| Line [500-532] |
|  |
| Solution Printer |
| Line [43-73] |
|  |
| Solutions: Obtained total five solutions for the given puzzle |
| -----------------Question#2 Soduku -----------------------------  Solution:#1  [[2 6 1 8 4 7 5 3 9]  [7 4 5 9 2 3 6 8 1]  [8 9 3 1 5 6 4 2 7]  [1 8 7 5 9 4 3 6 2]  [4 5 9 6 3 2 7 1 8]  [3 2 6 7 1 8 9 5 4]  [5 7 8 2 6 9 1 4 3]  [9 1 4 3 8 5 2 7 6]  [6 3 2 4 7 1 8 9 5]]  Solution:#2  [[2 6 1 7 4 8 5 3 9]  [7 4 5 9 2 3 6 8 1]  [8 9 3 1 5 6 4 2 7]  [1 8 7 5 9 4 3 6 2]  [4 5 9 6 3 2 7 1 8]  [3 2 6 8 1 7 9 5 4]  [5 7 8 2 6 9 1 4 3]  [9 1 4 3 8 5 2 7 6]  [6 3 2 4 7 1 8 9 5]]  Solution:#3  [[2 6 1 9 4 8 5 3 7]  [7 4 5 1 2 3 6 8 9]  [8 9 3 7 5 6 4 2 1]  [1 8 7 5 9 4 3 6 2]  [4 5 9 6 3 2 7 1 8]  [3 2 6 8 1 7 9 5 4]  [5 7 8 2 6 9 1 4 3]  [9 1 4 3 8 5 2 7 6]  [6 3 2 4 7 1 8 9 5]]  Solution:#4  [[2 6 8 7 4 9 5 3 1]  [7 4 5 1 2 3 6 8 9]  [1 9 3 5 8 6 4 2 7]  [8 1 7 9 5 4 3 6 2]  [4 5 9 6 3 2 7 1 8]  [3 2 6 8 1 7 9 5 4]  [5 7 4 2 6 8 1 9 3]  [9 8 1 3 7 5 2 4 6]  [6 3 2 4 9 1 8 7 5]]  Solution:#5  [[2 8 6 7 4 9 5 3 1]  [7 4 5 1 2 3 6 8 9]  [1 9 3 5 8 6 4 2 7]  [8 1 7 9 5 4 3 6 2]  [4 5 9 6 3 2 7 1 8]  [3 6 2 8 1 7 9 5 4]  [5 7 4 2 6 8 1 9 3]  [9 2 1 3 7 5 8 4 6]  [6 3 8 4 9 1 2 7 5]]  OPTIMAL |
| **problem\_2() is wrapper for calling all the needed function and displaying the result (all solution)**  Line [866-875] |
|  |
| Line [554-570] |
|  |

TASK#3

|  |
| --- |
| A: Load the excel file **Assignment\_DA\_1\_data.xlsx** and extract all relevant information |
| Line [589-620] |
|  |
| B: Create the decision variables |
| Line [622-731] |
|  |
|  |
|  |
| C: Contractor cannot work on two projects simultaneously |
| Line [734-747] |
|  |
| D: If a project is accepted to be delivered then exactly one contractor per job of the project needs to work on it |
| Line [750-764] |
|  |
| E: If a project is not taken on then no one should be contracted to work on it |
| Line [767-781] |
|  |
| F: project dependency and project conflict constraints |
| Line [783-797] |
|  |
| G: the difference between the value of all delivered projects and the cost of all required subcontractors, is at least €2160 |
| Line [800-826] |
|  |
| Solution Printer |
| Line [76-120] |
|  |
| Obtained total 5 solutions |
| -----------------Question#3 Project planner---------------------  Solution:1:  *Project C->[('M4', 'Job H', 'Contractor H'), ('M5', 'Job E', 'Contractor E'), ('M6', 'Job G', 'Contractor G'), ('M7', 'Job B', 'Contractor E'), ('M8', 'Job E', 'Contractor E')]*  *Project D->[('M2', 'Job D', 'Contractor H'), ('M3', 'Job F', 'Contractor F'), ('M4', 'Job I', 'Contractor G'), ('M5', 'Job H', 'Contractor H')]*  *Project E->[('M8', 'Job J', 'Contractor C'), ('M9', 'Job A', 'Contractor A')]*  *Project H->[('M8', 'Job A', 'Contractor A'), ('M9', 'Job B', 'Contractor E'), ('M10', 'Job D', 'Contractor H'), ('M11', 'Job I', 'Contractor G')]*  *Project I->[('M10', 'Job L', 'Contractor D'), ('M11', 'Job F', 'Contractor F'), ('M12', 'Job K', 'Contractor K')]*  **Profit Margin = 2165.0**  Solution:2:  *Project C->[('M4', 'Job H', 'Contractor H'), ('M5', 'Job E', 'Contractor E'), ('M6', 'Job G', 'Contractor G'), ('M7', 'Job B', 'Contractor E'), ('M8', 'Job E', 'Contractor E')]*  *Project D->[('M2', 'Job D', 'Contractor H'), ('M3', 'Job F', 'Contractor F'), ('M4', 'Job I', 'Contractor G'), ('M5', 'Job H', 'Contractor H')]*  *Project E->[('M8', 'Job J', 'Contractor C'), ('M9', 'Job A', 'Contractor A')]*  *Project H->[('M8', 'Job A', 'Contractor A'), ('M9', 'Job B', 'Contractor E'), ('M10', 'Job D', 'Contractor H'), ('M11', 'Job I', 'Contractor G')]*  *Project I->[('M10', 'Job L', 'Contractor D'), ('M11', 'Job F', 'Contractor F'), ('M12', 'Job K', 'Contractor B')]*  **Profit Margin = 2175.0**  Solution:3:  *Project C->[('M4', 'Job H', 'Contractor H'), ('M5', 'Job E', 'Contractor A'), ('M6', 'Job G', 'Contractor G'), ('M7', 'Job B', 'Contractor E'), ('M8', 'Job E', 'Contractor E')]*  *Project D->[('M2', 'Job D', 'Contractor H'), ('M3', 'Job F', 'Contractor F'), ('M4', 'Job I', 'Contractor G'), ('M5', 'Job H', 'Contractor H')]*  *Project E->[('M8', 'Job J', 'Contractor C'), ('M9', 'Job A', 'Contractor A')]*  *Project H->[('M8', 'Job A', 'Contractor A'), ('M9', 'Job B', 'Contractor E'), ('M10', 'Job D', 'Contractor H'), ('M11', 'Job I', 'Contractor G')]*  *Project I->[('M10', 'Job L', 'Contractor D'), ('M11', 'Job F', 'Contractor F'), ('M12', 'Job K', 'Contractor B')]*  **Profit Margin = 2165.0**  Solution:4:  *Project C->[('M4', 'Job H', 'Contractor H'), ('M5', 'Job E', 'Contractor E'), ('M6', 'Job G', 'Contractor G'), ('M7', 'Job B', 'Contractor E'), ('M8', 'Job E', 'Contractor E')]*  *Project D->[('M2', 'Job D', 'Contractor H'), ('M3', 'Job F', 'Contractor F'), ('M4', 'Job I', 'Contractor G'), ('M5', 'Job H', 'Contractor H')]*  *Project E->[('M8', 'Job J', 'Contractor C'), ('M9', 'Job A', 'Contractor A')]*  *Project H->[('M8', 'Job A', 'Contractor A'), ('M9', 'Job B', 'Contractor E'), ('M10', 'Job D', 'Contractor H'), ('M11', 'Job I', 'Contractor G')]*  *Project I->[('M10', 'Job L', 'Contractor A'), ('M11', 'Job F', 'Contractor F'), ('M12', 'Job K', 'Contractor B')]*  **Profit Margin = 2165.0**  Solution:5:  *Project A->[('M1', 'Job A', 'Contractor A'), ('M2', 'Job B', 'Contractor E'), ('M3', 'Job C', 'Contractor K')]*  *Project C->[('M4', 'Job H', 'Contractor H'), ('M5', 'Job E', 'Contractor E'), ('M6', 'Job G', 'Contractor G'), ('M7', 'Job B', 'Contractor E'), ('M8', 'Job E', 'Contractor E')]*  *Project D->[('M2', 'Job D', 'Contractor H'), ('M3', 'Job F', 'Contractor F'), ('M4', 'Job I', 'Contractor G'), ('M5', 'Job H', 'Contractor H')]*  *Project E->[('M8', 'Job J', 'Contractor C'), ('M9', 'Job A', 'Contractor A')]*  *Project H->[('M8', 'Job A', 'Contractor A'), ('M9', 'Job B', 'Contractor E'), ('M10', 'Job D', 'Contractor H'), ('M11', 'Job I', 'Contractor G')]*  *Project I->[('M10', 'Job L', 'Contractor D'), ('M11', 'Job F', 'Contractor F'), ('M12', 'Job K', 'Contractor B')]*  **Profit Margin = 2165.0**  OPTIMAL |
| **problem\_3() is wrapper for calling all the needed function and displaying the result (all solution)**  Line [877-886] |
|  |
| **Line [843-852]** |
|  |

|  |
| --- |
|  |