**Member names:**

**20181019 Gahyeon Kim,**

**20181058 Juhyun Kim**

**20181173 Kyuhwan Lee**

**20181315 Keonmo Lee**

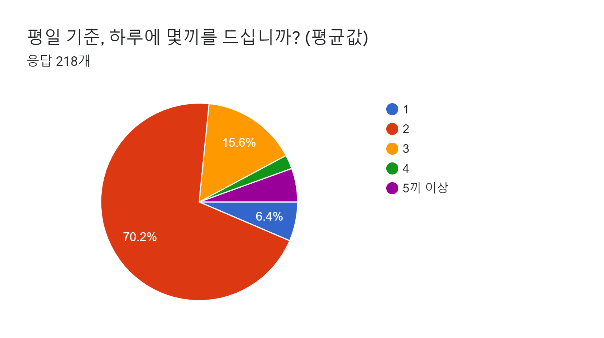
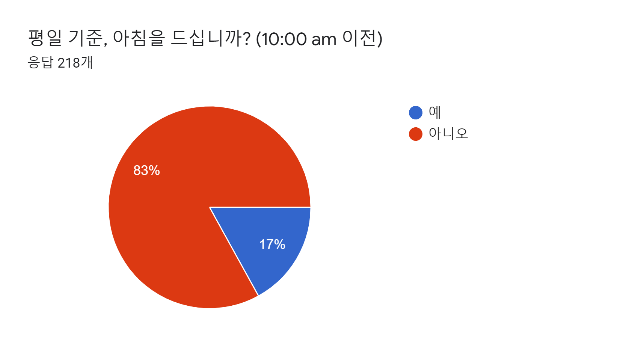
**20181381 Sunghoon Choi**

# Executive Summary

This study was launched to solve the problem that within the UNIST, there is a lack of optimized diet programs. Research will consider the point that students rarely eat breakfast and that they should take the necessary nutrients from limited menus considering limited food expense. This study will try to find a proper menu that can prevent over-intaking of nutrients while getting enough nutrients, during the weekday. Also, this optimized menu can overcome limits for UNIST menu-decision problem like accessibility and restaurant facilities. Through this project, this project is going to set object function and constraints about methods learned in OR that can satisfy our constraints.

# Introduction & Literature Review

In campus life, there are many students considering about keeping their health with balanced nutrients and what to eat. Especially UNIST student has trouble about that. Unfortunate for UNIST students, there aren’t much place for them to get proper meal. And most of them are uncertain that they are having sufficient daily nutrition. We also feel the same way, finally want to find out not only optimized methodology but also alternatives.



(Q1 means, on weekdays, do you eat breakfast before 10am?(red:N/blue:Y))

(Q2 means, in average on weekdays, how often do you usually eat a day?(blue:1/red:2/orange:3/green:4/purple:over 5))

The total amount of nutrients you get per day is the sum of the nutrients in each diet. Although many samples for the ideal diet have already been proposed, we feel the need to be optimized for the current situation, for example, usually skipping breakfast. Therefore, we conducted a survey to find out how often most UNIST students usually eat a day. As a result, they usually tend to skip breakfast and eat two meals for lunch and dinner, so we need a two-meal diet that can meet the nutrients and calories we need per day.

The paper(Maillot, Ferguson, Drewnowski, & Darmon, 2008) is about optimization of nutritional quality for their price. For optimizing this, they use Linear Programming and set variables, such as energy, proteins, carbohydrates, lipids, SFA, Added sugars, Sodium etc. And apply constraints with recommended intake. It gives our team rough solutions such as LP solution and selecting nutrient. But our team think it would be better if it supplies solutions for university students. Because, it is not easy for students to meet these constraints.

Furthermore, the paper (정효지 et al., 2009) states analysis of nutrient should focus on the carbohydrate, protein, and fat. Other constituent besides carbohydrate, protein and fat can be supplied within artificial nutritional supplements or synthesized inside our body.

# Problem statement

Main problem of this case study is to find optimized meal for UNIST student. Optimized meal means that the setting satisfies the two conditions; minimized cost and fulfilling recommended nutrient intake. Most of the students live in dormitories without cooking facilities, so they should pay for meal. Also, UNIST has low accessibility of getting foods outside campus. Therefore, we only consider the foods inside the UNIST. However, it is possible to cause their nutritional imbalances since they rely on the restrictive menus and food delivery. Thus, this study will do research about minimizing the cost of meal in UNIST while satisfying the constraints such as minimal nutrients.

To minimize students' spending on food, the data consist of diet information from nine restaurants in the school. This data consists of prices, calories, and essential nutrition ingredients for each diet. Some cafeteria offer different menus on a day-by-day basis, or may not. With this in mind, the variables were set to account for the day of the week and the diet. That is, each variable represents the number of times that the diet is eaten on a given day of the week. These parameters will be used to solve the purpose function of minimizing costs and serve as a tool to explain solutions.

# Solution procedure

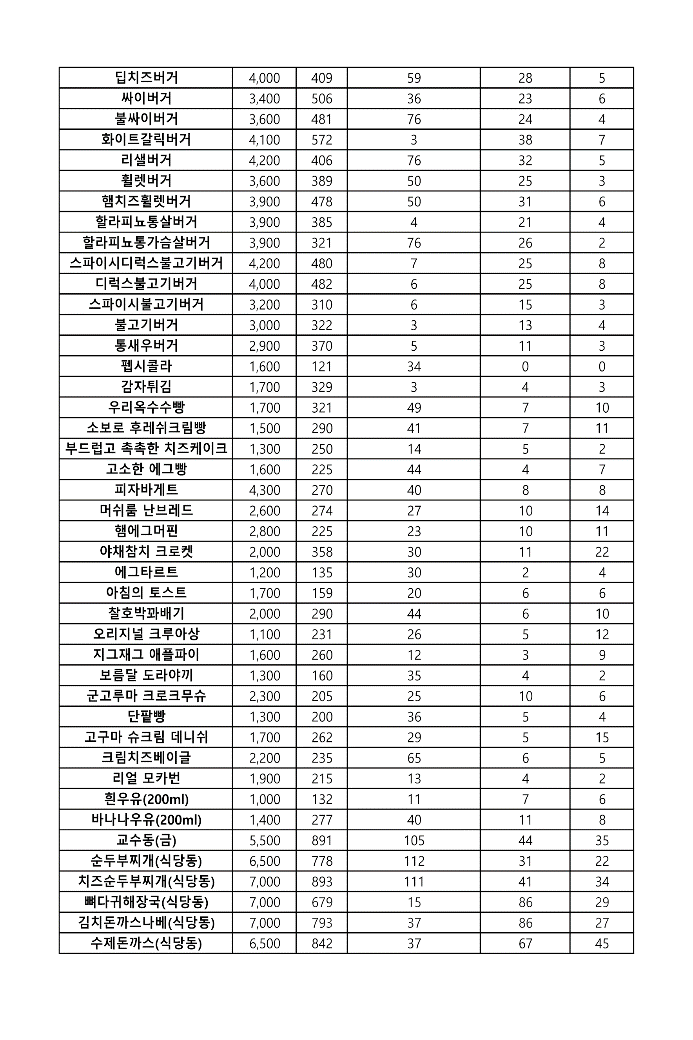
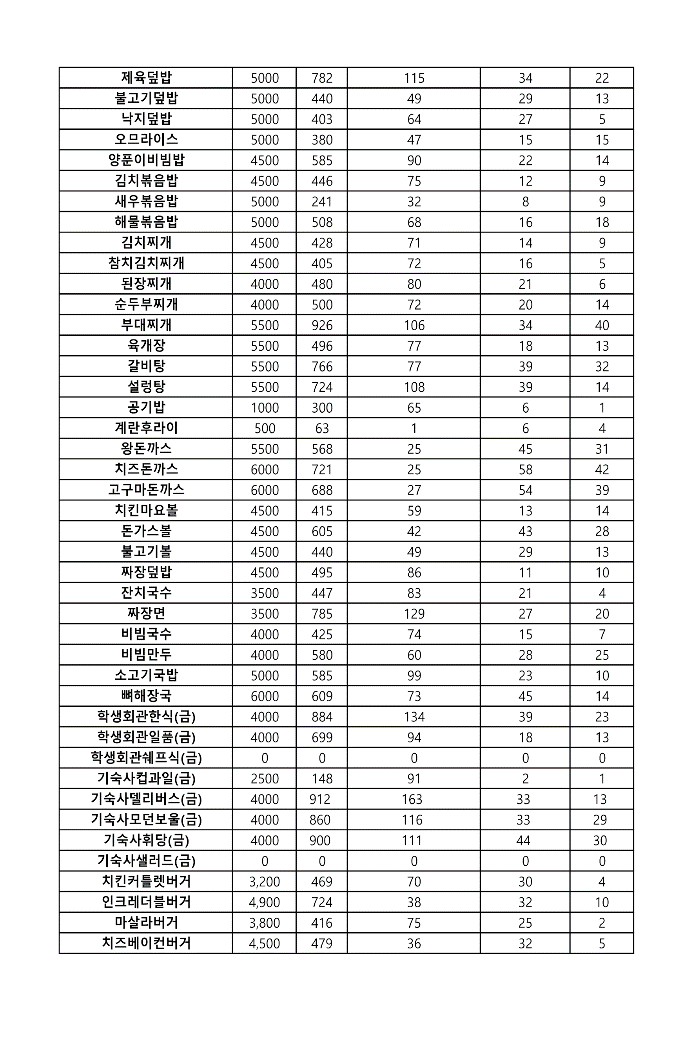
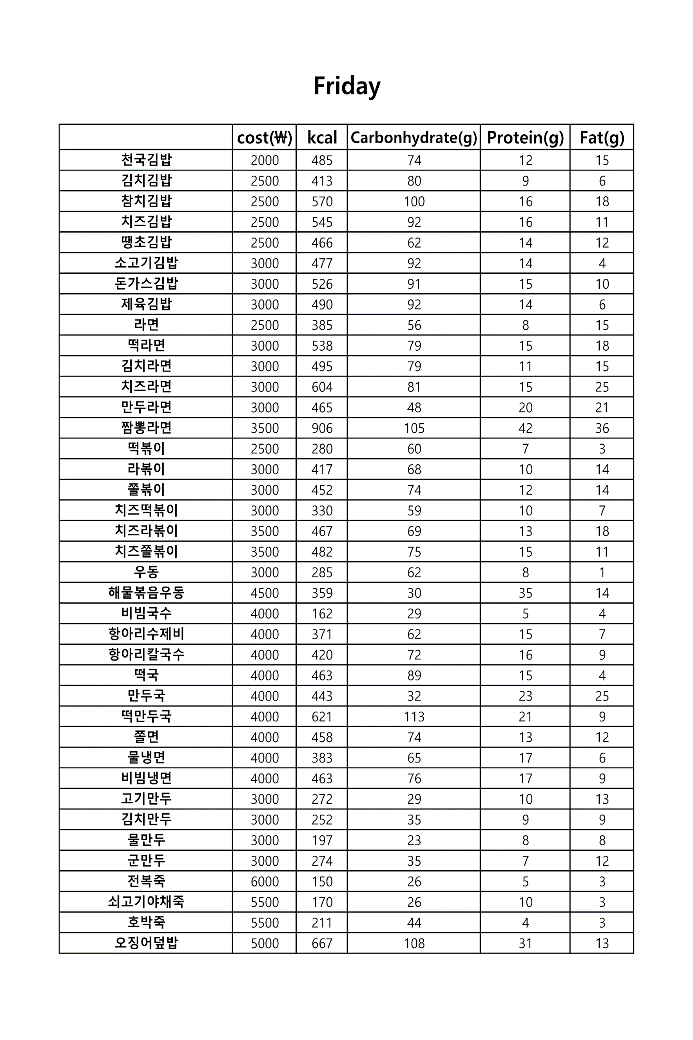
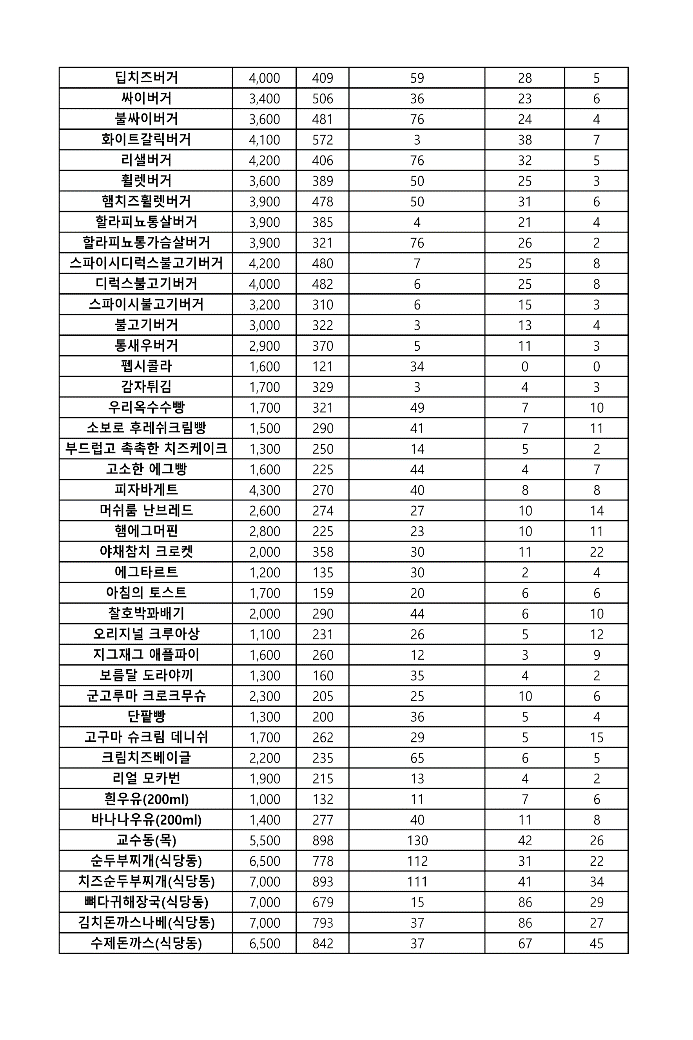
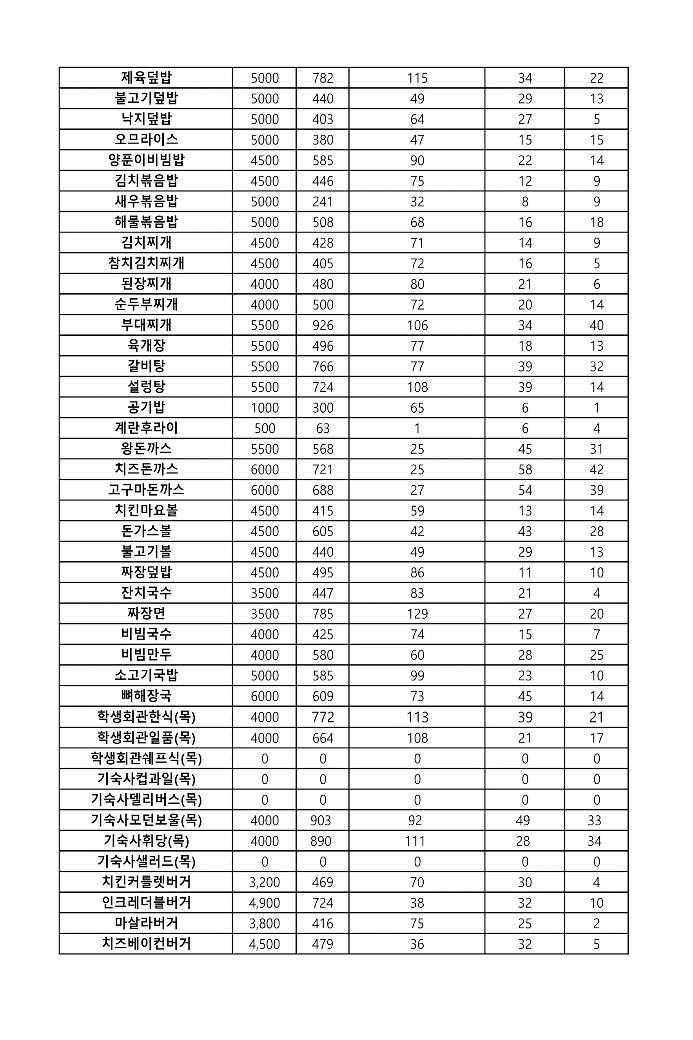
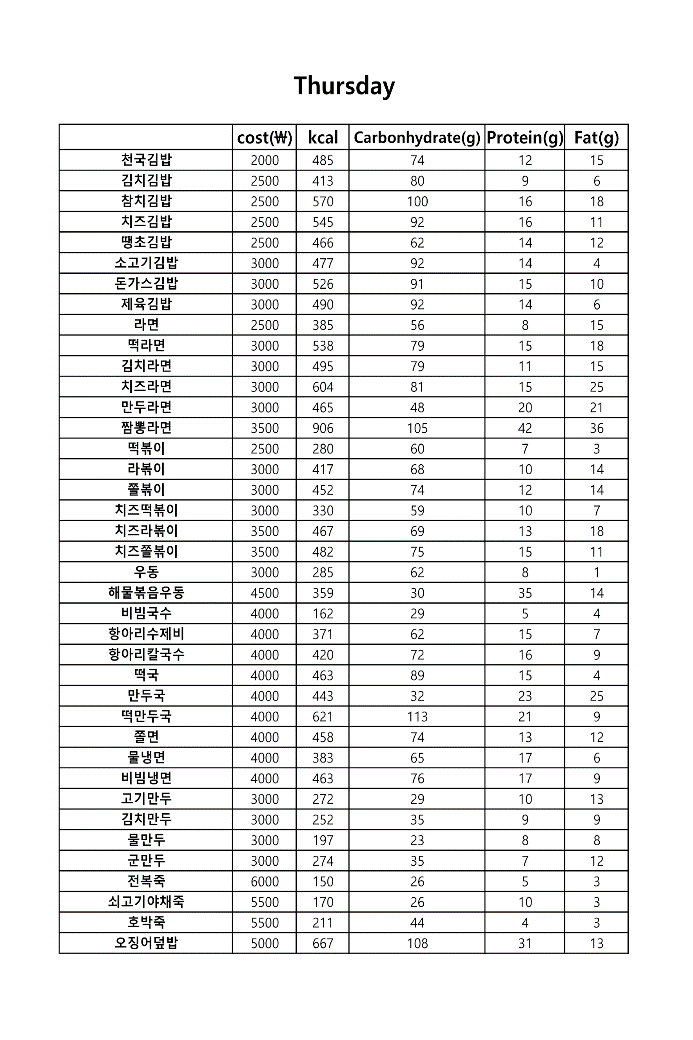
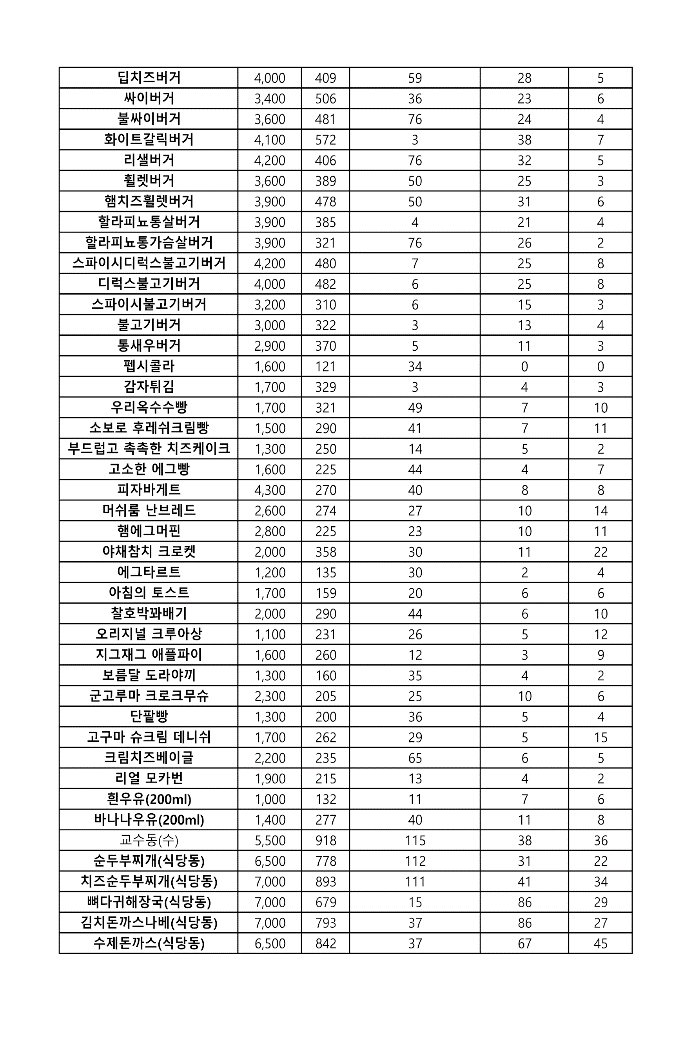
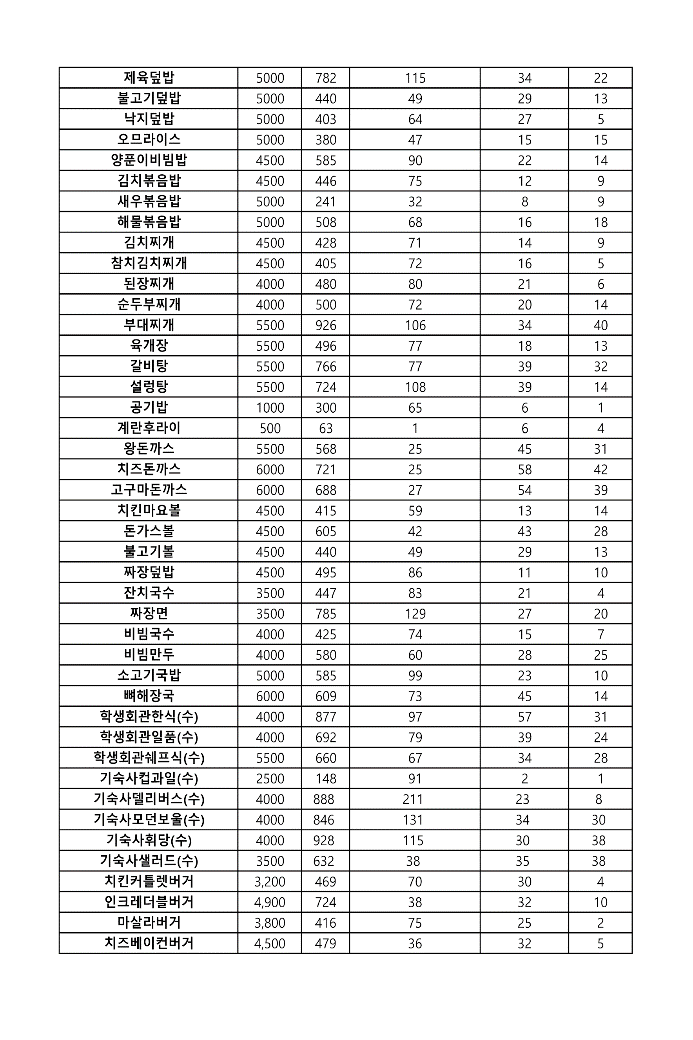
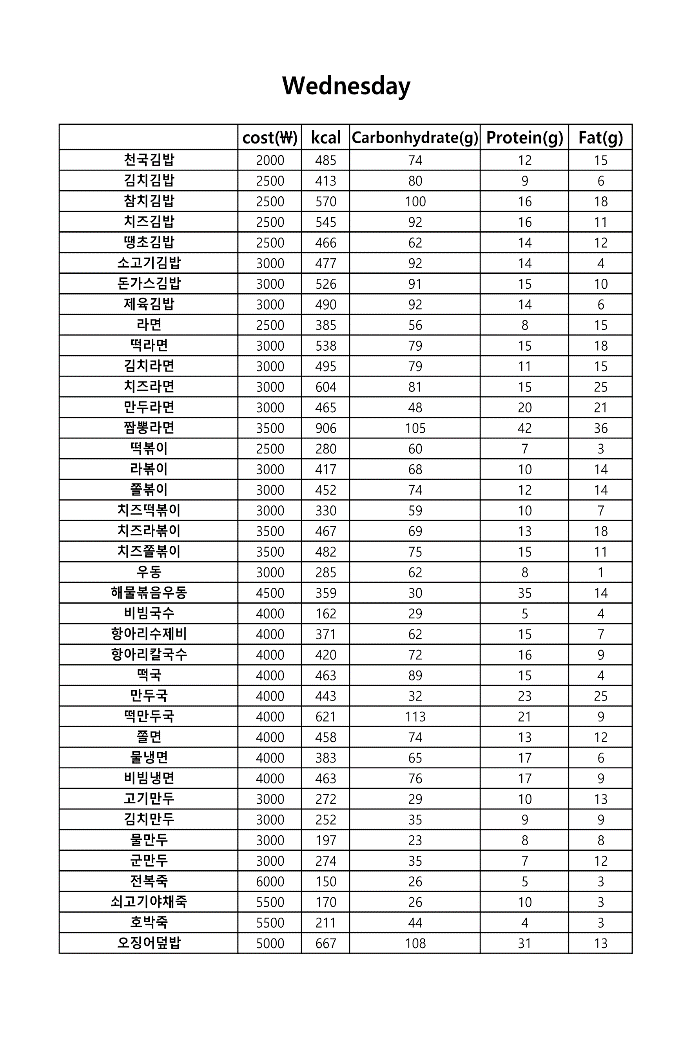
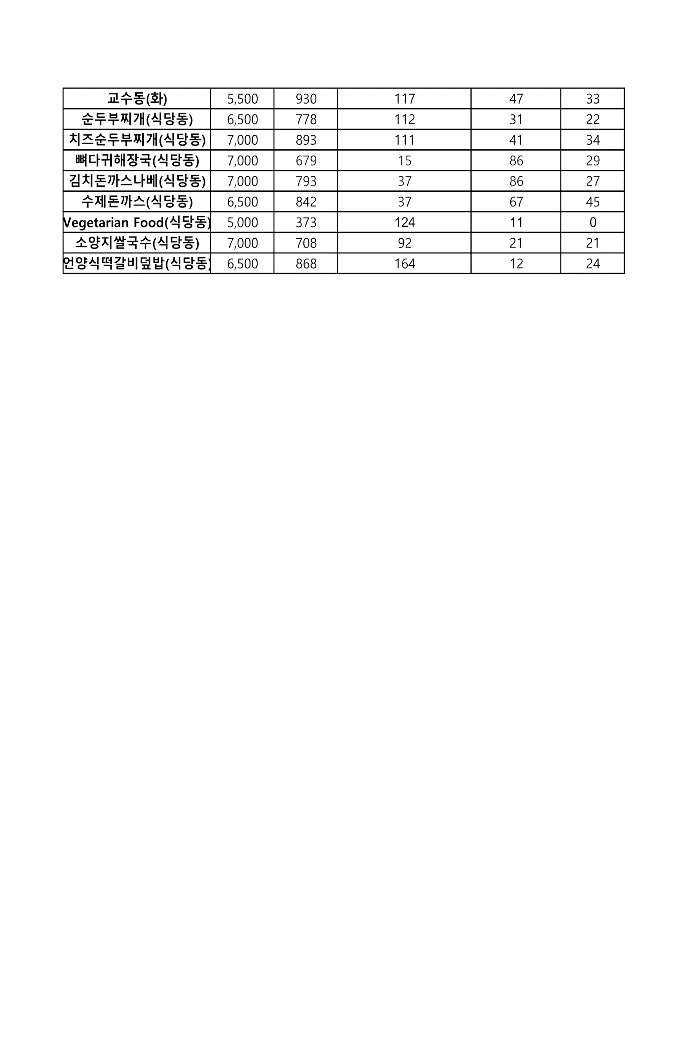
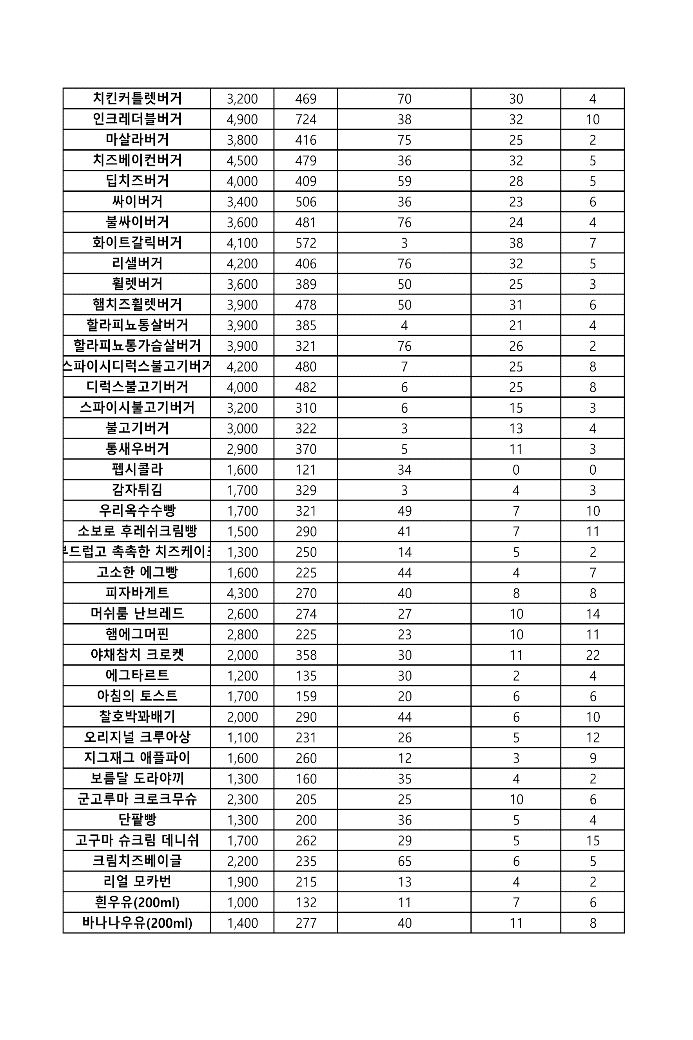
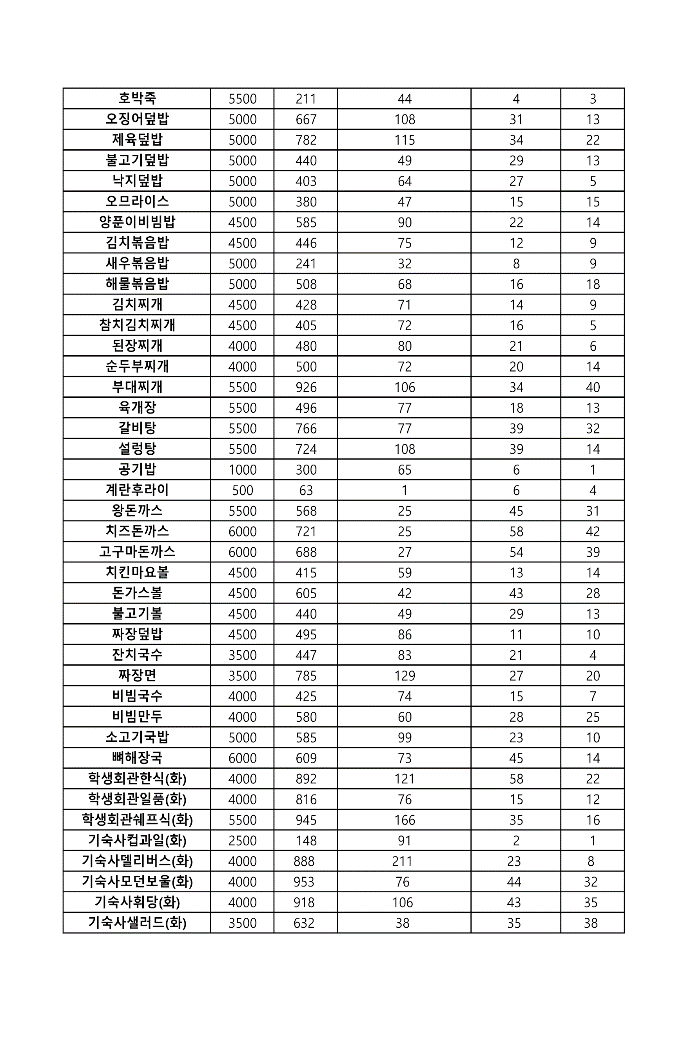
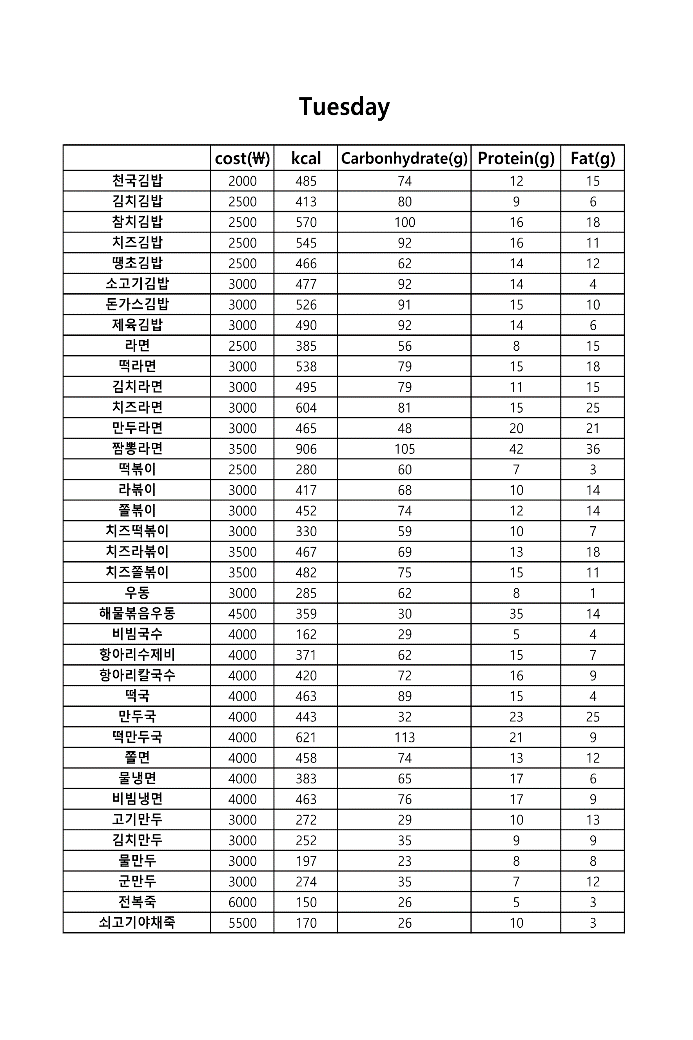
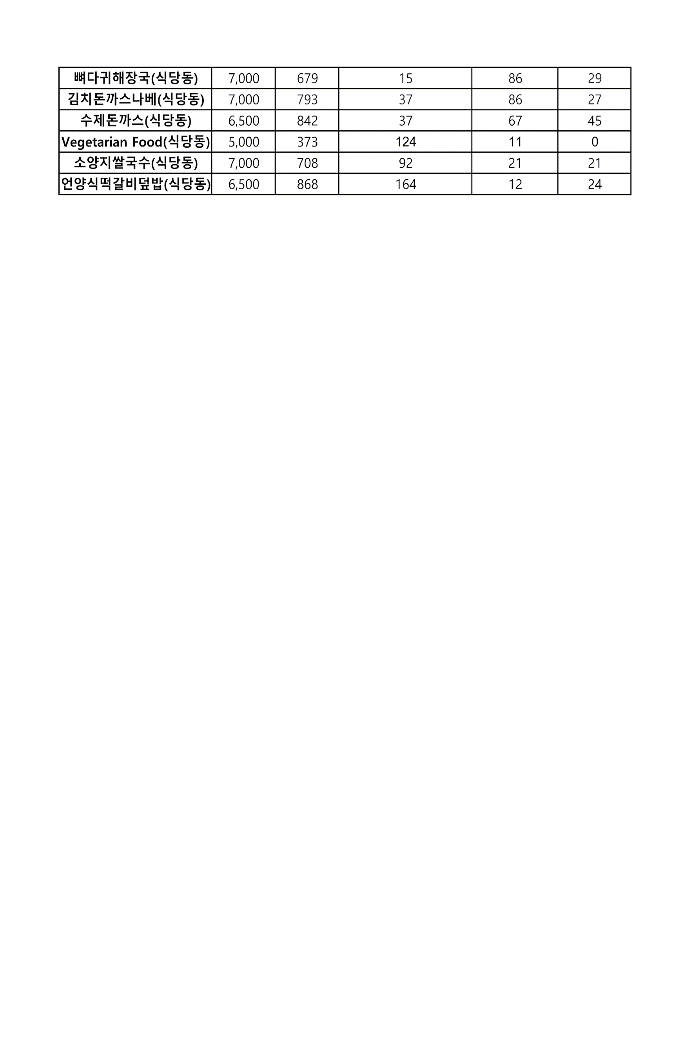
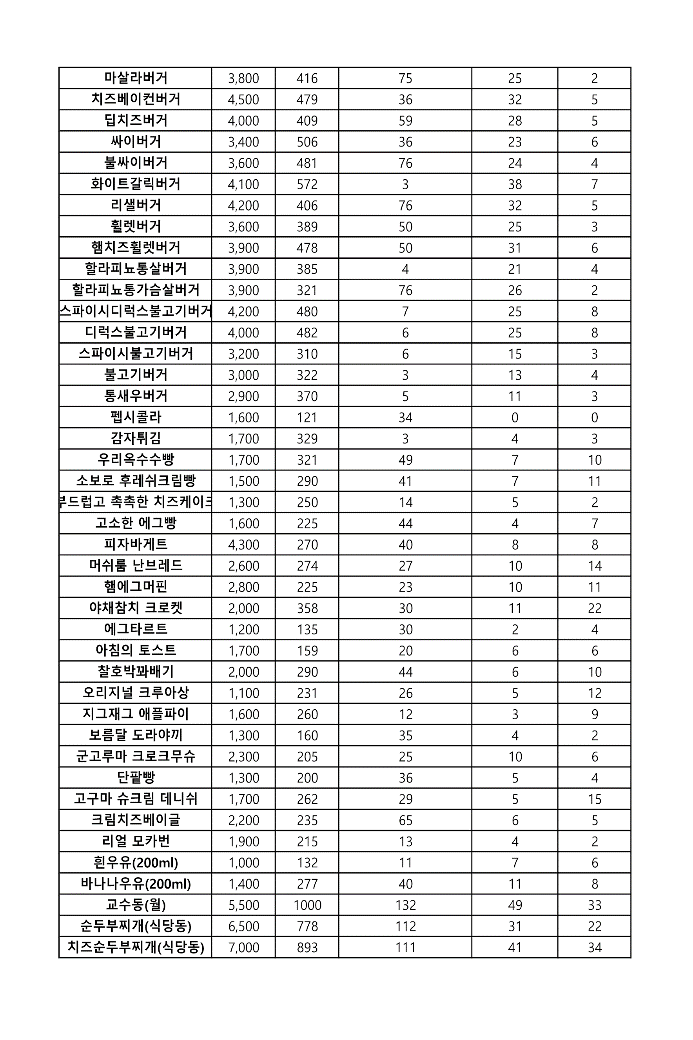
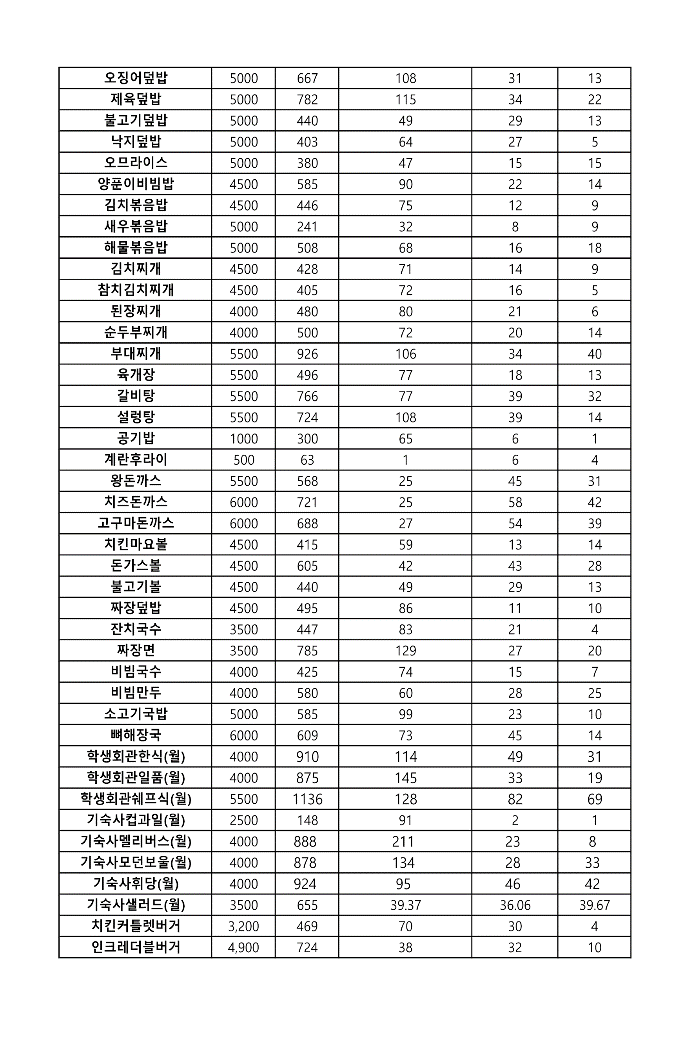
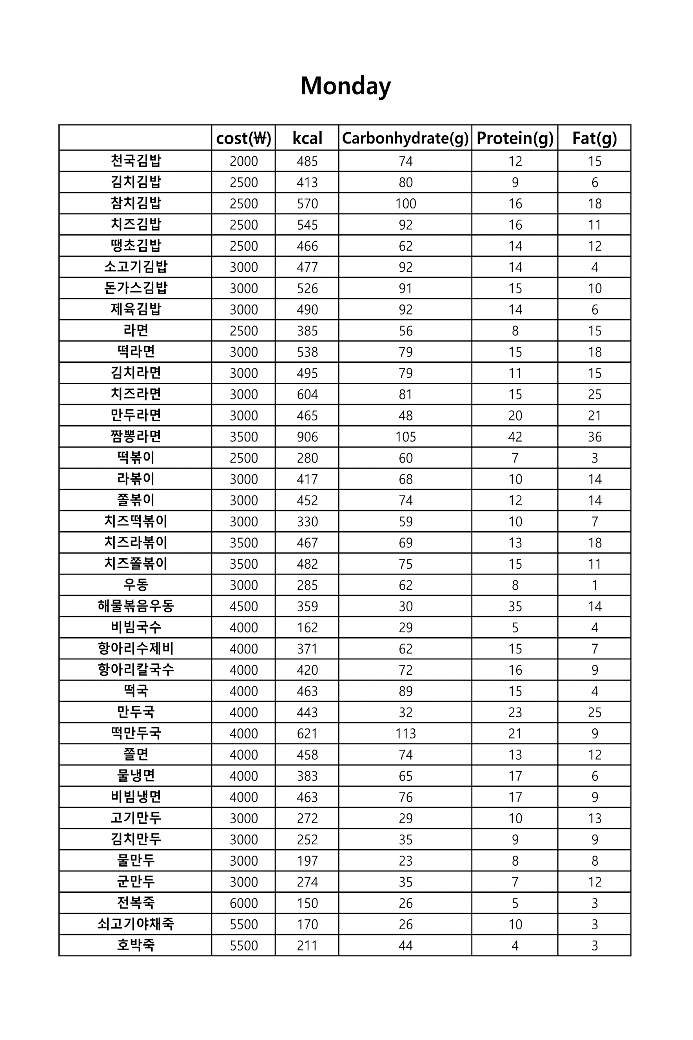
Our team decided to use Linear Programming to create an optimized diet in the UNIST. In order to get Decision Variables, we examined the carbohydrates, protein, fat, calories, and prices of the menu items of UNIST's restaurants. The restaurants we surveyed are student union restaurant, Dormitory restaurant, main building restaurant, Mom’s touch, Bread & Co, Cafeteria, Kimbap Heaven, Dodream in UNIST.

In the case of student union restaurants, Dormitory restaurant, and main-building restaurants, the nutritionist assumes that the menu changes every day of the week, so the average price is paid for lunch and dinner.

In the case of Mom’s touch, we investigated single products. In the case of set products, the extra cost varies from set to set, so it was adjusted to get the single products combined.

In the case of Bread & Co, Kimbap heaven, Dodream, and Cafeteria, the variables are the same depending on the day of the week.

The next step is to set the object function. Since our purpose is to adjust the proper calories and nutrients and minimize the price, we set the proper calories and nutrients as constraint and set the price as object function. In other words, the object function minimizes the price.



Decision Variables-

Ai: Amount of i-th menu consumed per day on Monday

Bi: Amount of i-th menu consumed per day on Tuesday

Ci: Amount of i-th menu consumed per day on Wednesday

Di: Amount of i-th menu consumed per day on Thursday

Ei: Amount of i-th menu consumed per day on Friday

1. Only data on only calories, carbohydrates, protein and fat is considered since vitamin and mineral deficiency can be solved through various supplementation, but hyper ingestion of nutrient cannot be solved without proper menu selection.(Ha & Kim, 2019)
2. We refer to nutrients containing generic(Food name, cooking or FDA) based on <https://www.myfitnesspal.com/ko/food/search>. If there is no relevant information, we refer to https://www.fatsecret.kr/ as the next best thing.
3. All menus were measured based on one serving.
4. Round to first digit if there is a decimal point
5. i from 1 to 60 is the menu of Kimbap Heaven

For foods containing cheese, calculated adding kcal=50, carbohydrates = 1g, protein = 3g, fat = 4g per slice of cheese.

1 roll of Kimbap = 300g

1. i from 61 to 70 is the menu of Dodream

Calculated by adding 1 serving of rice to 1 serving of each soup, since the soup is served with rice.

Except for the expensive menus serving the same food from different restaurant if there are overlapping menus, since our goal is to minimize costs.

Except for selling more than 2 servings since we deal with a diet for 1 person

1. i from 71 to 73 is the menu of student union restaurant.

Using expected nutrition information like kcal because variable system does not divide menu of lunch and dinner

1. i from 74 to 78 is the menu of dormitory restaurant.

Using expected nutrition information like kcal because variable system does not divide menu of lunch and dinner

1. i from 79 to 98 is the menu of Mom’s touch

Calculated based on a single item.

Coke and french fries are also a variable but those will not be considered as meal, but supplement for Mom’s touch menu.

1. i from 99 to 119 is the menu of Bread & Co,

Milk menus are also considered as variables, but those will not be considered as meal. Rather, those will be kind of supplement for Bread & Co menu.

1. 120th i is the menu of main building restaurant.

Using expected nutrition information like kcal because variable system does not divide menu of lunch and dinner

1. i from 121 to 128 is the menu of cafeteria.

Objective function-

Minimize z,



Constraints,

It focused on developing a meal plan to create a nutritionally balanced diet for Koreans. The paper informs that while Koreans are deficient in vitamins and minerals, they are getting fat and energy more than daily nutrient standard, over 20% on standard. (Ha & Kim, 2019)

Therefore, this paper categorized the food group of carbohydrate, fat and protein and proceeds with an analysis that can match the intake of three major nutrients to an appropriate level. Meanwhile, it would be brilliant approach to organize a Unist-only diet to keep the proper proportions of carbohydrates, proteins, and fats: 55 to 65 (%), 15 to 25 (%), and 15 to 25 (%). Focusing on creating a diet that takes into account the intake and proportion of all three major nutrients would be meaningful.

Non-negativity



# Analysis

**i) For men [390 65 65 => 273‬ 45.5‬ 45.5]**

* **About the result 1:**

{controlled by proper proportions of carbohydrates, proteins, and fats : each 55 to 65 (%), 15 to 25 (%), and 15 to 25 (%) and by limits of amounts of carbohydrates, proteins, and fats}

Monday : ['학생회관쉐프식(월)', '언양식떡갈비덮밥(식당동)'] [cost : 4000 + 6500]

=> Diet having Minimum cost : ['학생회관쉐프식(월)', '언양식떡갈비덮밥(식당동)'], 4000 + 6500 = 10500

Tuesday : No diet

Wednesday : No diet

Thursday : No diet

Friday : No diet

|  |
| --- |
| import pandas as pd  # Monday  df = pd.read\_csv("dite\_Mon.csv", encoding='euc\_kr')  list2 = []  list21 = [[1,1,110000000000]]  list22 = []  for i in range(len(df)):  for j in range(i, len(df)):  if df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate'] >= 273 and df.loc[i]['Protein'] + df.loc[j]['Protein'] >= 45.5 and df.loc[i]['Fat'] + df.loc[j]['Fat']>= 45.5:  carb = df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate']  protein = df.loc[i]['Protein'] +df.loc[j]['Protein']  fat = df.loc[i]['Fat'] + df.loc[j]['Fat']  under = carb + protein + fat  ratio\_carb = carb/under  ratio\_protein = protein/under  ratio\_fat = protein/under  if 0.55 <= ratio\_carb <= 0.65:  if 0.15 <= ratio\_protein <= 0.25:  if 0.15 <= ratio\_fat <= 0.25:  list2.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list21[0][2] > int(df.loc[i]['cost'])+ int(df.loc[j]['cost']):  del list21[:]  del list22[:]  list21.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list22.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  elif list21[0][2] == int(df.loc[i]['cost']) + int(df.loc[j]['cost']):  list21.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list22.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  else:  continue  else:  continue  if list2 == []:  list2.append({'Monday': 'No diet'})  print(list2)  print(list21)  print(list22)  print('=============================')  # Tuesday  df = pd.read\_csv("dite\_Tues.csv", encoding='euc\_kr')  list3 = []  list31 = [[1,1,10000000000]]  list32 = []  for i in range(len(df)):  for j in range(i, len(df)):  if df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate'] >= 273 and df.loc[i]['Protein'] + df.loc[j]['Protein'] >= 45.5 and df.loc[i]['Fat'] + df.loc[j]['Fat']>= 45.5:  carb = df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate']  protein = df.loc[i]['Protein'] +df.loc[j]['Protein']  fat = df.loc[i]['Fat'] + df.loc[j]['Fat']  under = carb + protein + fat  ratio\_carb = carb/under  ratio\_protein = protein/under  ratio\_fat = protein/under  if 0.55 <= ratio\_carb <= 0.65:  if 0.15 <= ratio\_protein <= 0.25:  if 0.15 <= ratio\_fat <= 0.25:  list3.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list31[0][2] > int(df.loc[i]['cost'])+ int(df.loc[j]['cost']):  del list31[:]  del list32[:]  list31.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list32.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  elif list31[0][2] == int(df.loc[i]['cost']) + int(df.loc[j]['cost']):  list31.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list32.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  else:  continue  else:  continue  if list3 == []:  list3.append({'Tuesday': 'No diet'})  print(list3)  print(list31)  print(list32)  print('=============================')  # Wednesday  df = pd.read\_csv("dite\_Wed.csv", encoding='euc\_kr')  list4 = []  list41 = [[1,1,110000000000]]  list42 = []  for i in range(len(df)):  for j in range(i, len(df)):  if df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate'] >= 273 and df.loc[i]['Protein'] + df.loc[j]['Protein'] >= 45.5 and df.loc[i]['Fat'] + df.loc[j]['Fat']>= 45.5:  carb = df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate']  protein = df.loc[i]['Protein'] + df.loc[j]['Protein']  fat = df.loc[i]['Fat'] + df.loc[j]['Fat']  under = carb + protein + fat  ratio\_carb = carb/under  ratio\_protein = protein/under  ratio\_fat = protein/under  if 0.55 <= ratio\_carb <= 0.65:  if 0.15 <= ratio\_protein <= 0.25:  if 0.15 <= ratio\_fat <= 0.25:  list4.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list41[0][2] > int(df.loc[i]['cost'])+ int(df.loc[j]['cost']):  del list41[:]  del list42[:]  list41.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list42.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  elif list41[0][2] == int(df.loc[i]['cost']) + int(df.loc[j]['cost']):  list41.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list42.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  else:  continue  else:  continue  if list4 == []:  list4.append({'Wednesday':'No diet'})  print(list4)  print(list41)  print(list42)  print('=============================')  # Thursday  df = pd.read\_csv("dite\_Thurs.csv", encoding='euc\_kr')  list5 = []  list51 = [[1,1,110000000000]]  list52 =[]  for i in range(len(df)):  for j in range(i, len(df)):  if df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate'] >= 273 and df.loc[i]['Protein'] + df.loc[j]['Protein'] >= 45.5 and df.loc[i]['Fat'] + df.loc[j]['Fat']>= 45.5:  carb = df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate']  protein = df.loc[i]['Protein'] +df.loc[j]['Protein']  fat = df.loc[i]['Fat'] + df.loc[j]['Fat']  under = carb + protein + fat  ratio\_carb = carb/under  ratio\_protein = protein/under  ratio\_fat = protein/under  if 0.55 <= ratio\_carb <= 0.65:  if 0.15 <= ratio\_protein <= 0.25:  if 0.15 <= ratio\_fat <= 0.25:  list5.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list51[0][2] > int(df.loc[i]['cost']) + int(df.loc[j]['cost']):  del list51[:]  del list52[:]  list51.append(  [df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list52.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  elif list51[0][2] == int(df.loc[i]['cost']) + int(df.loc[j]['cost']):  list51.append(  [df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list52.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  else:  continue  else:  continue    if list5 == []:  list5.append({'Thursday':'No diet'})  print(list5)  print(list51)  print(list52)  print('=============================')  # Friday  df = pd.read\_csv("dite\_Fri.csv", encoding='euc\_kr')  list6 = []  list61 = [[1,1,110000000000]]  list62 = []  for i in range(len(df)):  for j in range(i, len(df)):  if df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate'] >= 273 and df.loc[i]['Protein'] + df.loc[j]['Protein'] >= 45.5 and df.loc[i]['Fat'] + df.loc[j]['Fat']>= 45.5:  carb = df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate']  protein = df.loc[i]['Protein'] +df.loc[j]['Protein']  fat = df.loc[i]['Fat'] + df.loc[j]['Fat']  under = carb + protein + fat  ratio\_carb = carb/under  ratio\_protein = protein/under  ratio\_fat = protein/under  if 0.55 <= ratio\_carb <= 0.65:  if 0.15 <= ratio\_protein <= 0.25:  if 0.15 <= ratio\_fat <= 0.25:  list6.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list61[0][2] > int(df.loc[i]['cost'])+ int(df.loc[j]['cost']):  del list61[:]  del list62[:]  list61.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list62.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  elif list61[0][2] == int(df.loc[i]['cost']) + int(df.loc[j]['cost']):  list61.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list62.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  else:  continue  else:  continue  if list6 == []:  list6.append({'Friday':'No diet'})  print(list6)  print(list61)  print(list62) |

* **About the result 2:**

{only controlled by limits of amounts of carbohydrates, proteins, and fats}

Monday :

['제육덮밥', '언양식떡갈비덮밥(식당동)'], ['부대찌개', '기숙사델리버스(월)'], ['학생회관한식(월)', '언양식떡갈비덮밥(식당동)'], ['학생회관일품(월)', '학생회관쉐프식(월)'], ['학생회관일품(월)', '기숙사모던보울(월)'], ['학생회관일품(월)', '교수동(월)'], ['학생회관쉐프식(월)', '기숙사델리버스(월)'], ['학생회관쉐프식(월)', '언양식떡갈비덮밥(식당동)'], ['기숙사델리버스(월)', '기숙사휘당(월)'], ['교수동(월)', '언양식떡갈비덮밥(식당동)'], ['치즈순두부찌개(식당동)', '언양식떡갈비덮밥(식당동)']

=> Diet having Minimum cost : ['학생회관일품(월)', '기숙사모던보울(월)'], ['기숙사델리버스(월)', '기숙사휘당(월)'], 4000 + 4000 = 8000

Tuesday :

['제육덮밥', '언양식떡갈비덮밥(식당동)'], ['부대찌개', '기숙사델리버스(화)'], ['학생회관한식(화)', '언양식떡갈비덮밥(식당동)'], ['학생회관쉐프식(화)', '교수동(화)'], ['학생회관쉐프식(화)', '치즈순두부찌개(식당동)'], ['교수동(화)', '언양식떡갈비덮밥(식당동)'], ['치즈순두부찌개(식당동)', '언양식떡갈비덮밥(식당동)']

=> Diet having Minimum cost : ['부대찌개', '기숙사델리버스(화)'], 5500+ 4000 =9500

Wednesday :

['제육덮밥', '언양식떡갈비덮밥(식당동)'], ['부대찌개', '기숙사델리버스(수)'], ['기숙사델리버스(수)', '기숙사휘당(수)'], ['기숙사모던보울(수)', '언양식떡갈비덮밥(식당동)'], ['교수동(수)', '언양식떡갈비덮밥(식당동)'], ['치즈순두부찌개(식당동)', '언양식떡갈비덮밥(식당동)']

=> Diet having Minimum cost : ['기숙사델리버스(수)', '기숙사휘당(수)'], 4000 + 4000 = 8000

Thursday :

['제육덮밥', '언양식떡갈비덮밥(식당동)'], ['교수동(목)', '언양식떡갈비덮밥(식당동)'], ['치즈순두부찌개(식당동)', '언양식떡갈비덮밥(식당동)']

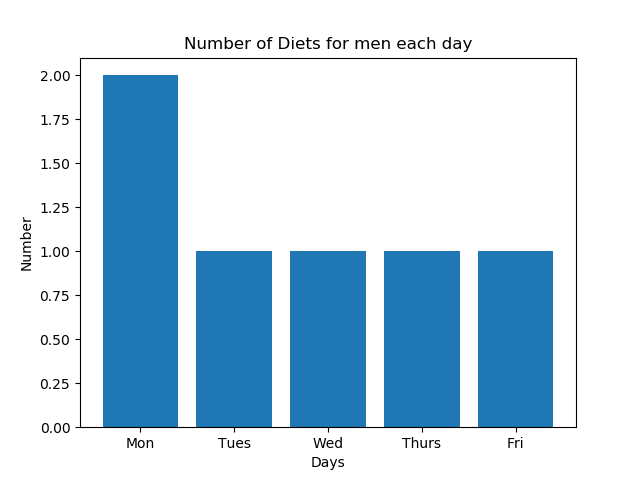
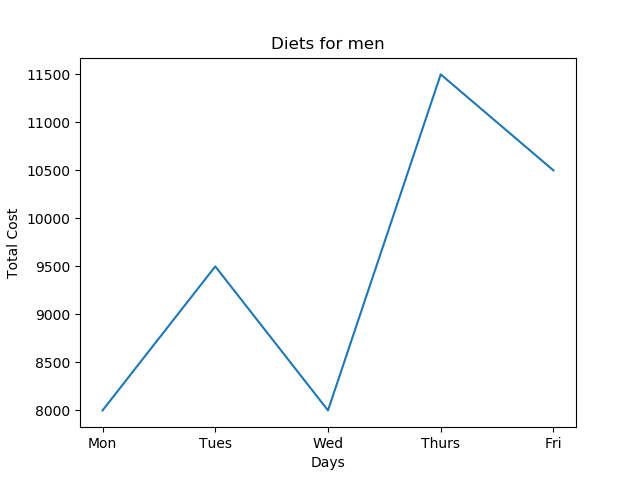
=> Diet having Minimum cost : ['제육덮밥', '언양식떡갈비덮밥(식당동)'], 5000 + 6500 = 11500

Friday :

['제육덮밥', '언양식떡갈비덮밥(식당동)'], ['학생회관한식(금)', '언양식떡갈비덮밥(식당동)'], ['기숙사델리버스(금)', '치즈순두부찌개(식당동)'], ['기숙사휘당(금)', '언양식떡갈비덮밥(식당동)'], ['치즈순두부찌개(식당동)', '언양식떡갈비덮밥(식당동)']

=> Diet having Minimum cost : ['학생회관한식(금)', '언양식떡갈비덮밥(식당동)'], 4000 + 6500 = 10500

|  |
| --- |
| import pandas as pd  # Monday  df = pd.read\_csv("dite\_Mon.csv", encoding='euc\_kr')  list2 = []  list21 = [[1,1,110000000000]]  list22 = []  for i in range(len(df)):  for j in range(i, len(df)):  if df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate'] >= 273 and df.loc[i]['Protein'] + df.loc[j]['Protein'] >= 45.5 and df.loc[i]['Fat'] + df.loc[j]['Fat']>= 45.5:  carb = df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate']  protein = df.loc[i]['Protein'] +df.loc[j]['Protein']  fat = df.loc[i]['Fat'] + df.loc[j]['Fat']  under = carb + protein + fat  ratio\_carb = carb/under  ratio\_protein = protein/under  ratio\_fat = protein/under  list2.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list21[0][2] > int(df.loc[i]['cost'])+ int(df.loc[j]['cost']):  del list21[:]  del list22[:]  list21.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list22.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  elif list21[0][2] == int(df.loc[i]['cost']) + int(df.loc[j]['cost']):  list21.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list22.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list2 == []:  list2.append({'Monday': 'No diet'})  print(list2)  print(list21)  print(list22)  print('=============================')  # Tuesday  df = pd.read\_csv("dite\_Tues.csv", encoding='euc\_kr')  list3 = []  list31 = [[1,1,10000000000]]  list32 = []  for i in range(len(df)):  for j in range(i, len(df)):  if df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate'] >= 273 and df.loc[i]['Protein'] + df.loc[j]['Protein'] >= 45.5 and df.loc[i]['Fat'] + df.loc[j]['Fat']>= 45.5:  carb = df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate']  protein = df.loc[i]['Protein'] +df.loc[j]['Protein']  fat = df.loc[i]['Fat'] + df.loc[j]['Fat']  under = carb + protein + fat  ratio\_carb = carb/under  ratio\_protein = protein/under  ratio\_fat = protein/under  list3.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list31[0][2] > int(df.loc[i]['cost'])+ int(df.loc[j]['cost']):  del list31[:]  del list32[:]  list31.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list32.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  elif list31[0][2] == int(df.loc[i]['cost']) + int(df.loc[j]['cost']):  list31.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list32.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list3 == []:  list3.append({'Tuesday': 'No diet'})  print(list3)  print(list31)  print(list32)  print('=============================')  # Wednesday  df = pd.read\_csv("dite\_Wed.csv", encoding='euc\_kr')  list4 = []  list41 = [[1,1,110000000000]]  list42 = []  for i in range(len(df)):  for j in range(i, len(df)):  if df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate'] >= 273 and df.loc[i]['Protein'] + df.loc[j]['Protein'] >= 45.5 and df.loc[i]['Fat'] + df.loc[j]['Fat']>= 45.5:  carb = df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate']  protein = df.loc[i]['Protein'] + df.loc[j]['Protein']  fat = df.loc[i]['Fat'] + df.loc[j]['Fat']  under = carb + protein + fat  ratio\_carb = carb/under  ratio\_protein = protein/under  ratio\_fat = protein/under  list4.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list41[0][2] > int(df.loc[i]['cost'])+ int(df.loc[j]['cost']):  del list41[:]  del list42[:]  list41.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list42.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  elif list41[0][2] == int(df.loc[i]['cost']) + int(df.loc[j]['cost']):  list41.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list42.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list4 == []:  list4.append({'Wednesday':'No diet'})  print(list4)  print(list41)  print(list42)  print('=============================')  # Thursday  df = pd.read\_csv("dite\_Thurs.csv", encoding='euc\_kr')  list5 = []  list51 = [[1,1,110000000000]]  list52 =[]  for i in range(len(df)):  for j in range(i, len(df)):  if df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate'] >= 273 and df.loc[i]['Protein'] + df.loc[j]['Protein'] >= 45.5 and df.loc[i]['Fat'] + df.loc[j]['Fat']>= 45.5:  carb = df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate']  protein = df.loc[i]['Protein'] +df.loc[j]['Protein']  fat = df.loc[i]['Fat'] + df.loc[j]['Fat']  under = carb + protein + fat  ratio\_carb = carb/under  ratio\_protein = protein/under  ratio\_fat = protein/under  list5.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list51[0][2] > int(df.loc[i]['cost'])+ int(df.loc[j]['cost']):  del list51[:]  del list52[:]  list51.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list52.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  elif list51[0][2] == int(df.loc[i]['cost']) + int(df.loc[j]['cost']):  list51.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list52.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list5 == []:  list5.append({'Thursday':'No diet'})  print(list5)  print(list51)  print(list52)  print('=============================')  # Friday  df = pd.read\_csv("dite\_Fri.csv", encoding='euc\_kr')  list6 = []  list61 = [[1,1,110000000000]]  list62 = []  for i in range(len(df)):  for j in range(i, len(df)):  if df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate'] >= 273 and df.loc[i]['Protein'] + df.loc[j]['Protein'] >= 45.5 and df.loc[i]['Fat'] + df.loc[j]['Fat']>= 45.5:  carb = df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate']  protein = df.loc[i]['Protein'] +df.loc[j]['Protein']  fat = df.loc[i]['Fat'] + df.loc[j]['Fat']  under = carb + protein + fat  ratio\_carb = carb/under  ratio\_protein = protein/under  ratio\_fat = protein/under  list6.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list61[0][2] > int(df.loc[i]['cost'])+ int(df.loc[j]['cost']):  del list61[:]  del list62[:]  list61.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list62.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  elif list61[0][2] == int(df.loc[i]['cost']) + int(df.loc[j]['cost']):  list61.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list62.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list6 == []:  list6.append({'Friday':'No diet'})  print(list6)  print(list61)  print(list62) |



**ii) For women [315 55 52.5 => 220.5‬ 38.5‬ 36.75]**

**About the result 3:**

{controlled by proper proportions of carbohydrates, proteins, and fats : each 55 to 65 (%), 15 to 25 (%), and 15 to 25 (%) and by limits of amounts of carbohydrates, proteins, and fats}

Monday :

['짬뽕라면', '기숙사모던보울(월)'], ['제육덮밥', '부대찌개'], ['부대찌개', '기숙사모던보울(월)'], ['학생회관한식(월)', '기숙사모던보울(월)'], ['학생회관쉐프식(월)', '언양식떡갈비덮밥(식당동)'], ['기숙사모던보울(월)', '기숙사휘당(월)'], ['기숙사모던보울(월)', '치즈순두부찌개(식당동)']

=> Diet having Minimum cost : ['짬뽕라면', '기숙사모던보울(월)'], 3500 + 4000 =7500

Tuesday :

['제육덮밥', '부대찌개']

=> Diet having Minimum cost : ['제육덮밥', '부대찌개'], 5000 + 5500 = 10500

Wednesday :

['제육덮밥', '부대찌개'], ['제육덮밥', '기숙사휘당(수)'], ['제육덮밥', '교수동(수)'], ['부대찌개', '기숙사모던보울(수)'], ['부대찌개', '기숙사휘당(수)'], ['부대찌개', '교수동(수)'], ['기숙사모던보울(수)', '교수동(수)'], ['기숙사모던보울(수)', '치즈순두부찌개(식당동)'], ['기숙사휘당(수)', '기숙사휘당(수)'], ['기숙사휘당(수)', '교수동(수)'], ['기숙사휘당(수)', '치즈순두부찌개(식당동)'], ['교수동(수)', '순두부찌개(식당동)']

=> Diet having Minimum cost : ['기숙사휘당(수)', '기숙사휘당(수)'], 4000 + 4000 = 8000

Thursday :

['제육덮밥', '부대찌개'], ['학생회관한식(목)', '기숙사휘당(목)'], ['기숙사휘당(목)', '기숙사휘당(목)'], ['기숙사휘당(목)', '교수동(목)'], ['기숙사휘당(목)', '치즈순두부찌개(식당동)']

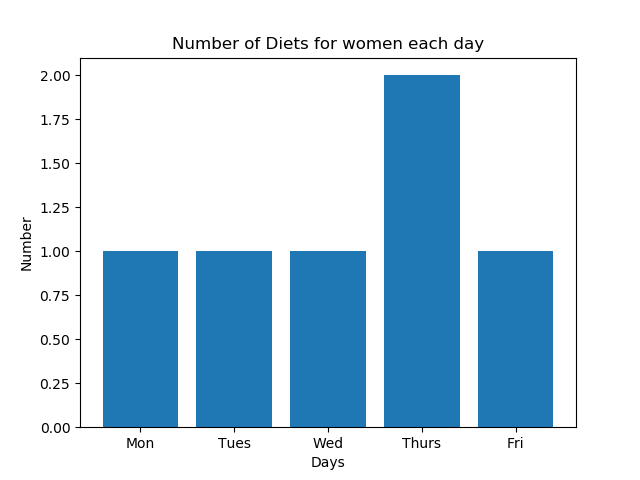
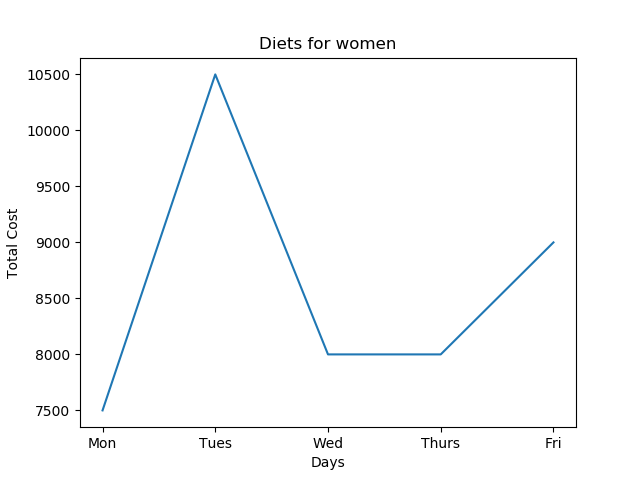
=> Diet having Minimum cost : ['학생회관한식(목)', '기숙사휘당(목)'], ['기숙사휘당(목)', '기숙사휘당(목)'], both 4000 + 4000 = 8000

Friday :

['제육덮밥', '부대찌개'], ['부대찌개', '학생회관한식(금)'], ['부대찌개', '기숙사모던보울(금)'], ['짜장면', '교수동(금)']

=> Diet having Minimum cost : ['짜장면', '교수동(금)'], 3500 + 5500 = 9000

|  |
| --- |
| import pandas as pd  # Monday  df = pd.read\_csv("dite\_Mon.csv", encoding='euc\_kr')  list2 = []  list21 = [[1,1,110000000000]]  list22 = []  for i in range(len(df)):  for j in range(i, len(df)):  if df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate'] >= 220.5 and df.loc[i]['Protein'] + df.loc[j]['Protein'] >= 38.5 and df.loc[i]['Fat'] + df.loc[j]['Fat']>= 36.75:  carb = df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate']  protein = df.loc[i]['Protein'] +df.loc[j]['Protein']  fat = df.loc[i]['Fat'] + df.loc[j]['Fat']  under = carb + protein + fat  ratio\_carb = carb/under  ratio\_protein = protein/under  ratio\_fat = protein/under  if 0.55 <= ratio\_carb <= 0.65:  if 0.15 <= ratio\_protein <= 0.25:  if 0.15 <= ratio\_fat <= 0.25:  list2.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list21[0][2] > int(df.loc[i]['cost'])+ int(df.loc[j]['cost']):  del list21[:]  del list22[:]  list21.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list22.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  elif list21[0][2] == int(df.loc[i]['cost']) + int(df.loc[j]['cost']):  list21.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list22.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  else:  continue  else:  continue  if list2 == []:  list2.append({'Monday': 'No diet'})  print(list2)  print(list21)  print(list22)  print('=============================')  # Tuesday  df = pd.read\_csv("dite\_Tues.csv", encoding='euc\_kr')  list3 = []  list31 = [[1,1,10000000000]]  list32 = []  for i in range(len(df)):  for j in range(i, len(df)):  if df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate'] >= 220.5 and df.loc[i]['Protein'] + df.loc[j]['Protein'] >= 38.5 and df.loc[i]['Fat'] + df.loc[j]['Fat']>= 36.75:  carb = df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate']  protein = df.loc[i]['Protein'] +df.loc[j]['Protein']  fat = df.loc[i]['Fat'] + df.loc[j]['Fat']  under = carb + protein + fat  ratio\_carb = carb/under  ratio\_protein = protein/under  ratio\_fat = protein/under  if 0.55 <= ratio\_carb <= 0.65:  if 0.15 <= ratio\_protein <= 0.25:  if 0.15 <= ratio\_fat <= 0.25:  list3.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list31[0][2] > int(df.loc[i]['cost'])+ int(df.loc[j]['cost']):  del list31[:]  del list32[:]  list31.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list32.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  elif list31[0][2] == int(df.loc[i]['cost']) + int(df.loc[j]['cost']):  list31.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list32.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  else:  continue  else:  continue  if list3 == []:  list3.append({'Tuesday': 'No diet'})  print(list3)  print(list31)  print(list32)  print('=============================')  # Wednesday  df = pd.read\_csv("dite\_Wed.csv", encoding='euc\_kr')  list4 = []  list41 = [[1,1,110000000000]]  list42 = []  for i in range(len(df)):  for j in range(i, len(df)):  if df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate'] >= 220.5 and df.loc[i]['Protein'] + df.loc[j]['Protein'] >= 38.5 and df.loc[i]['Fat'] + df.loc[j]['Fat']>= 36.75:  carb = df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate']  protein = df.loc[i]['Protein'] + df.loc[j]['Protein']  fat = df.loc[i]['Fat'] + df.loc[j]['Fat']  under = carb + protein + fat  ratio\_carb = carb/under  ratio\_protein = protein/under  ratio\_fat = protein/under  if 0.55 <= ratio\_carb <= 0.65:  if 0.15 <= ratio\_protein <= 0.25:  if 0.15 <= ratio\_fat <= 0.25:  list4.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list41[0][2] > int(df.loc[i]['cost'])+ int(df.loc[j]['cost']):  del list41[:]  del list42[:]  list41.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list42.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  elif list41[0][2] == int(df.loc[i]['cost']) + int(df.loc[j]['cost']):  list41.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list42.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  else:  continue  else:  continue  if list4 == []:  list4.append({'Wednesday':'No diet'})  print(list4)  print(list41)  print(list42)  print('=============================')  # Thursday  df = pd.read\_csv("dite\_Thurs.csv", encoding='euc\_kr')  list5 = []  list51 = [[1,1,110000000000]]  list52 =[]  for i in range(len(df)):  for j in range(i, len(df)):  if df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate'] >= 220.5 and df.loc[i]['Protein'] + df.loc[j]['Protein'] >= 38.5 and df.loc[i]['Fat'] + df.loc[j]['Fat']>= 36.75:  carb = df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate']  protein = df.loc[i]['Protein'] +df.loc[j]['Protein']  fat = df.loc[i]['Fat'] + df.loc[j]['Fat']  under = carb + protein + fat  ratio\_carb = carb/under  ratio\_protein = protein/under  ratio\_fat = protein/under  if 0.55 <= ratio\_carb <= 0.65:  if 0.15 <= ratio\_protein <= 0.25:  if 0.15 <= ratio\_fat <= 0.25:  list5.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list51[0][2] > int(df.loc[i]['cost']) + int(df.loc[j]['cost']):  del list51[:]  del list52[:]  list51.append(  [df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list52.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  elif list51[0][2] == int(df.loc[i]['cost']) + int(df.loc[j]['cost']):  list51.append(  [df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list52.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  else:  continue  else:  continue  if list5 == []:  list5.append({'Thursday':'No diet'})  print(list5)  print(list51)  print(list52)  print('=============================')  # Friday  df = pd.read\_csv("dite\_Fri.csv", encoding='euc\_kr')  list6 = []  list61 = [[1,1,110000000000]]  list62 = []  for i in range(len(df)):  for j in range(i, len(df)):  if df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate'] >= 220.5 and df.loc[i]['Protein'] + df.loc[j]['Protein'] >= 38.5 and df.loc[i]['Fat'] + df.loc[j]['Fat']>= 36.75:  carb = df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate']  protein = df.loc[i]['Protein'] +df.loc[j]['Protein']  fat = df.loc[i]['Fat'] + df.loc[j]['Fat']  under = carb + protein + fat  ratio\_carb = carb/under  ratio\_protein = protein/under  ratio\_fat = protein/under  if 0.55 <= ratio\_carb <= 0.65:  if 0.15 <= ratio\_protein <= 0.25:  if 0.15 <= ratio\_fat <= 0.25:  list6.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list61[0][2] > int(df.loc[i]['cost'])+ int(df.loc[j]['cost']):  del list61[:]  del list62[:]  list61.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list62.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  elif list61[0][2] == int(df.loc[i]['cost']) + int(df.loc[j]['cost']):  list61.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list62.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  else:  continue  else:  continue  if list6 == []:  list6.append({'Friday':'No diet'})  print(list6)  print(list61)  print(list62) |



**About the result 4:**

{only controlled by limits of amounts of carbohydrates, proteins, and fats}

Monday :

['참치김밥', '짜장면'], ['참치김밥', '학생회관일품(월)'], ['참치김밥', '학생회관쉐프식(월)'], ['참치김밥', '기숙사모던보울(월)'], ['참치김밥', '교수동(월)'], ['치즈김밥', '기숙사모던보울(월)'], ['치즈김밥', '교수동(월)'], ['소고기김밥', '기숙사모던보울(월)'], ['소고기김밥', '교수동(월)'], ['돈가스김밥', '기숙사모던보울(월)'], ['돈가스김밥', '교수동(월)'], ['제육김밥', '기숙사모던보울(월)'], ['제육김밥', '교수동(월)'], ['떡라면', '학생회관일품(월)'], ['치즈라면', '학생회관일품(월)'], ['짬뽕라면', '짜장면'], ['짬뽕라면', '학생회관일품(월)'], ['짬뽕라면', '학생회관쉐프식(월)'], ['짬뽕라면', '기숙사델리버스(월)'], ['짬뽕라면', '기숙사모던보울(월)'], ['짬뽕라면', '교수동(월)'], ['짬뽕라면', '언양식떡갈비덮밥(식당동)'], ['떡국', '기숙사모던보울(월)'], ['떡국', '교수동(월)'], ['떡만두국', '학생회관한식(월)'], ['떡만두국', '학생회관쉐프식(월)'], ['떡만두국', '기숙사모던보울(월)'], ['떡만두국', '교수동(월)'], ['떡만두국', '치즈순두부찌개(식당동)'], ['오징어덮밥', '학생회관한식(월)'], ['오징어덮밥', '학생회관쉐프식(월)'], ['오징어덮밥', '기숙사모던보울(월)'], ['오징어덮밥', '교수동(월)'], ['오징어덮밥', '언양식떡갈비덮밥(식당동)'], ['제육덮밥', '제육덮밥'], ['제육덮밥', '부대찌개'], ['제육덮밥', '짜장면'], ['제육덮밥', '학생회관한식(월)'], ['제육덮밥', '학생회관일품(월)'], ['제육덮밥', '학생회관쉐프식(월)'], ['제육덮밥', '기숙사모던보울(월)'], ['제육덮밥', '교수동(월)'], ['제육덮밥', '순두부찌개(식당동)'], ['제육덮밥', '치즈순두부찌개(식당동)'], ['제육덮밥', '언양식떡갈비덮밥(식당동)'], ['양푼이비빔밥', '기숙사모던보울(월)'], ['양푼이비빔밥', '교수동(월)'], ['부대찌개', '짜장면'], ['부대찌개', '학생회관일품(월)'], ['부대찌개', '학생회관쉐프식(월)'], ['부대찌개', '기숙사델리버스(월)'], ['부대찌개', '기숙사모던보울(월)'], ['부대찌개', '교수동(월)'], ['부대찌개', 'Vegetarian Food(식당동)'], ['부대찌개', '언양식떡갈비덮밥(식당동)'], ['갈비탕', '학생회관일품(월)'], ['갈비탕', '기숙사델리버스(월)'], ['갈비탕', '언양식떡갈비덮밥(식당동)'], ['설렁탕', '학생회관한식(월)'], ['설렁탕', '학생회관쉐프식(월)'], ['설렁탕', '기숙사모던보울(월)'], ['설렁탕', '교수동(월)'], ['설렁탕', '언양식떡갈비덮밥(식당동)'], ['왕돈까스', '기숙사델리버스(월)'], ['치즈돈까스', '기숙사델리버스(월)'], ['고구마돈까스', '기숙사델리버스(월)'], ['짜장면', '짜장면'], ['짜장면', '학생회관한식(월)'], ['짜장면', '학생회관일품(월)'], ['짜장면', '학생회관쉐프식(월)'], ['짜장면', '기숙사모던보울(월)'], ['짜장면', '기숙사휘당(월)'], ['짜장면', '교수동(월)'], ['짜장면', '순두부찌개(식당동)'], ['짜장면', '치즈순두부찌개(식당동)'], ['짜장면', '소양지쌀국수(식당동)'], ['짜장면', '언양식떡갈비덮밥(식당동)'], ['비빔만두', '언양식떡갈비덮밥(식당동)'], ['소고기국밥', '학생회관쉐프식(월)'], ['소고기국밥', '기숙사모던보울(월)'], ['소고기국밥', '교수동(월)'], ['뼈해장국', '언양식떡갈비덮밥(식당동)'], ['학생회관한식(월)', '학생회관한식(월)'], ['학생회관한식(월)', '학생회관일품(월)'], ['학생회관한식(월)', '학생회관쉐프식(월)'], ['학생회관한식(월)', '기숙사델리버스(월)'], ['학생회관한식(월)', '기숙사모던보울(월)'], ['학생회관한식(월)', '교수동(월)'], ['학생회관한식(월)', '순두부찌개(식당동)'], ['학생회관한식(월)', '치즈순두부찌개(식당동)'], ['학생회관한식(월)', '언양식떡갈비덮밥(식당동)'], ['학생회관일품(월)', '학생회관일품(월)'], ['학생회관일품(월)', '학생회관쉐프식(월)'], ['학생회관일품(월)', '기숙사모던보울(월)'], ['학생회관일품(월)', '기숙사휘당(월)'], ['학생회관일품(월)', '교수동(월)'], ['학생회관일품(월)', '순두부찌개(식당동)'], ['학생회관일품(월)', '치즈순두부찌개(식당동)'], ['학생회관일품(월)', '소양지쌀국수(식당동)'], ['학생회관일품(월)', '언양식떡갈비덮밥(식당동)'], ['학생회관쉐프식(월)', '학생회관쉐프식(월)'], ['학생회관쉐프식(월)', '기숙사델리버스(월)'], ['학생회관쉐프식(월)', '기숙사모던보울(월)'], ['학생회관쉐프식(월)', '기숙사휘당(월)'], ['학생회관쉐프식(월)', '교수동(월)'], ['학생회관쉐프식(월)', '순두부찌개(식당동)'], ['학생회관쉐프식(월)', '치즈순두부찌개(식당동)'], ['학생회관쉐프식(월)', 'Vegetarian Food(식당동)'], ['학생회관쉐프식(월)', '언양식떡갈비덮밥(식당동)'], ['기숙사델리버스(월)', '기숙사모던보울(월)'], ['기숙사델리버스(월)', '기숙사휘당(월)'], ['기숙사델리버스(월)', '기숙사샐러드(월)'], ['기숙사델리버스(월)', '교수동(월)'], ['기숙사델리버스(월)', '치즈순두부찌개(식당동)'], ['기숙사델리버스(월)', '뼈다귀해장국(식당동)'], ['기숙사델리버스(월)', '수제돈까스(식당동)'], ['기숙사모던보울(월)', '기숙사모던보울(월)'], ['기숙사모던보울(월)', '기숙사휘당(월)'], ['기숙사모던보울(월)', '교수동(월)'], ['기숙사모던보울(월)', '순두부찌개(식당동)'], ['기숙사모던보울(월)', '치즈순두부찌개(식당동)'], ['기숙사모던보울(월)', '소양지쌀국수(식당동)'], ['기숙사모던보울(월)', '언양식떡갈비덮밥(식당동)'], ['기숙사휘당(월)', '교수동(월)'], ['기숙사휘당(월)', '언양식떡갈비덮밥(식당동)'], ['교수동(월)', '교수동(월)'], ['교수동(월)', '순두부찌개(식당동)'], ['교수동(월)', '치즈순두부찌개(식당동)'], ['교수동(월)', '소양지쌀국수(식당동)'], ['교수동(월)', '언양식떡갈비덮밥(식당동)'], ['순두부찌개(식당동)', '순두부찌개(식당동)'], ['순두부찌개(식당동)', '치즈순두부찌개(식당동)'], ['순두부찌개(식당동)', '언양식떡갈비덮밥(식당동)'], ['치즈순두부찌개(식당동)', '치즈순두부찌개(식당동)'], ['치즈순두부찌개(식당동)', '언양식떡갈비덮밥(식당동)']]

=> Diet having Minimum cost : ['참치김밥', '짜장면'], 2500 + 3500 = 6000

Tuesday :

['참치김밥', '짜장면'], ['참치김밥', '학생회관한식(화)'], ['치즈라면', '학생회관쉐프식(화)'], ['짬뽕라면', '짜장면'], ['짬뽕라면', '학생회관한식(화)'], ['짬뽕라면', '학생회관쉐프식(화)'], ['짬뽕라면', '기숙사델리버스(화)'], ['짬뽕라면', '교수동(화)'], ['짬뽕라면', '언양식떡갈비덮밥(식당동)'], ['떡만두국', '교수동(화)'], ['떡만두국', '치즈순두부찌개(식당동)'], ['오징어덮밥', '교수동(화)'], ['오징어덮밥', '언양식떡갈비덮밥(식당동)'], ['제육덮밥', '제육덮밥'], ['제육덮밥', '부대찌개'], ['제육덮밥', '짜장면'], ['제육덮밥', '학생회관한식(화)'], ['제육덮밥', '학생회관쉐프식(화)'], ['제육덮밥', '기숙사휘당(화)'], ['제육덮밥', '교수동(화)'], ['제육덮밥', '순두부찌개(식당동)'], ['제육덮밥', '치즈순두부찌개(식당동)'], ['제육덮밥', '언양식떡갈비덮밥(식당동)'], ['부대찌개', '짜장면'], ['부대찌개', '학생회관한식(화)'], ['부대찌개', '학생회관쉐프식(화)'], ['부대찌개', '기숙사델리버스(화)'], ['부대찌개', '교수동(화)'], ['부대찌개', 'Vegetarian Food(식당동)'], ['부대찌개', '언양식떡갈비덮밥(식당동)'], ['갈비탕', '학생회관쉐프식(화)'], ['갈비탕', '기숙사델리버스(화)'], ['갈비탕', '언양식떡갈비덮밥(식당동)'], ['설렁탕', '교수동(화)'], ['설렁탕', '언양식떡갈비덮밥(식당동)'], ['왕돈까스', '기숙사델리버스(화)'], ['치즈돈까스', '기숙사델리버스(화)'], ['고구마돈까스', '기숙사델리버스(화)'], ['짜장면', '짜장면'], ['짜장면', '학생회관한식(화)'], ['짜장면', '기숙사휘당(화)'], ['짜장면', '교수동(화)'], ['짜장면', '순두부찌개(식당동)'], ['짜장면', '치즈순두부찌개(식당동)'], ['짜장면', '소양지쌀국수(식당동)'], ['짜장면', '언양식떡갈비덮밥(식당동)'], ['비빔만두', '학생회관쉐프식(화)'], ['비빔만두', '언양식떡갈비덮밥(식당동)'], ['뼈해장국', '언양식떡갈비덮밥(식당동)'], ['학생회관한식(화)', '학생회관한식(화)'], ['학생회관한식(화)', '학생회관쉐프식(화)'], ['학생회관한식(화)', '기숙사휘당(화)'], ['학생회관한식(화)', '교수동(화)'], ['학생회관한식(화)', '순두부찌개(식당동)'], ['학생회관한식(화)', '치즈순두부찌개(식당동)'], ['학생회관한식(화)', '언양식떡갈비덮밥(식당동)'], ['학생회관쉐프식(화)', '기숙사모던보울(화)'], ['학생회관쉐프식(화)', '기숙사휘당(화)'], ['학생회관쉐프식(화)', '교수동(화)'], ['학생회관쉐프식(화)', '순두부찌개(식당동)'], ['학생회관쉐프식(화)', '치즈순두부찌개(식당동)'], ['학생회관쉐프식(화)', '소양지쌀국수(식당동)'], ['학생회관쉐프식(화)', '언양식떡갈비덮밥(식당동)'], ['기숙사델리버스(화)', '기숙사모던보울(화)'], ['기숙사델리버스(화)', '기숙사휘당(화)'], ['기숙사델리버스(화)', '기숙사샐러드(화)'], ['기숙사델리버스(화)', '교수동(화)'], ['기숙사델리버스(화)', '치즈순두부찌개(식당동)'], ['기숙사델리버스(화)', '뼈다귀해장국(식당동)'], ['기숙사델리버스(화)', '수제돈까스(식당동)'], ['기숙사모던보울(화)', '언양식떡갈비덮밥(식당동)'], ['기숙사휘당(화)', '교수동(화)'], ['기숙사휘당(화)', '언양식떡갈비덮밥(식당동)'], ['교수동(화)', '교수동(화)'], ['교수동(화)', '순두부찌개(식당동)'], ['교수동(화)', '치즈순두부찌개(식당동)'], ['교수동(화)', '언양식떡갈비덮밥(식당동)'], ['순두부찌개(식당동)', '순두부찌개(식당동)'], ['순두부찌개(식당동)', '치즈순두부찌개(식당동)'], ['순두부찌개(식당동)', '언양식떡갈비덮밥(식당동)'], ['치즈순두부찌개(식당동)', '치즈순두부찌개(식당동)'], ['치즈순두부찌개(식당동)', '언양식떡갈비덮밥(식당동)']

['참치김밥', '짜장면'], ['참치김밥', '교수동(목)'], ['치즈김밥', '교수동(목)'], ['짬뽕라면', '짜장면'], ['짬뽕라면', '교수동(목)'], ['짬뽕라면', '언양식떡갈비덮밥(식당동)'], ['떡만두국', '기숙사휘당(목)'], ['떡만두국', '치즈순두부찌개(식당동)'], ['오징어덮밥', '교수동(목)'], ['오징어덮밥', '언양식떡갈비덮밥(식당동)'], ['제육덮밥', '제육덮밥'], ['제육덮밥', '부대찌개'], ['제육덮밥', '짜장면'], ['제육덮밥', '학생회관한식(목)'], ['제육덮밥', '학생회관일품(목)'], ['제육덮밥', '기숙사휘당(목)'], ['제육덮밥', '교수동(목)'], ['제육덮밥', '순두부찌개(식당동)'], ['제육덮밥', '치즈순두부찌개(식당동)'], ['제육덮밥', '언양식떡갈비덮밥(식당동)'], ['부대찌개', '짜장면'], ['부대찌개', '교수동(목)'], ['부대찌개', 'Vegetarian Food(식당동)'], ['부대찌개', '언양식떡갈비덮밥(식당동)'], ['갈비탕', '언양식떡갈비덮밥(식당동)'], ['설렁탕', '교수동(목)'], ['설렁탕', '언양식떡갈비덮밥(식당동)'], ['짜장면', '짜장면'], ['짜장면', '학생회관한식(목)'], ['짜장면', '학생회관일품(목)'], ['짜장면', '기숙사모던보울(목)'], ['짜장면', '기숙사휘당(목)'], ['짜장면', '교수동(목)'], ['짜장면', '순두부찌개(식당동)'], ['짜장면', '치즈순두부찌개(식당동)'], ['짜장면', '소양지쌀국수(식당동)'], ['짜장면', '언양식떡갈비덮밥(식당동)'], ['비빔만두', '언양식떡갈비덮밥(식당동)'], ['뼈해장국', '언양식떡갈비덮밥(식당동)'], ['학생회관한식(목)', '학생회관한식(목)'], ['학생회관한식(목)', '학생회관일품(목)'], ['학생회관한식(목)', '기숙사휘당(목)'], ['학생회관한식(목)', '교수동(목)'], ['학생회관한식(목)', '순두부찌개(식당동)'], ['학생회관한식(목)', '치즈순두부찌개(식당동)'], ['학생회관한식(목)', '언양식떡갈비덮밥(식당동)'], ['학생회관일품(목)', '교수동(목)'], ['기숙사모던보울(목)', '교수동(목)'], ['기숙사모던보울(목)', '언양식떡갈비덮밥(식당동)'], ['기숙사휘당(목)', '기숙사휘당(목)'], ['기숙사휘당(목)', '교수동(목)'], ['기숙사휘당(목)

=> Diet having Minimum cost : ['참치김밥', '짜장면'], 2500 + 3500 = 6000

Wednesday :

['참치김밥', '짜장면'], ['참치김밥', '기숙사모던보울(수)'], ['치즈김밥', '기숙사모던보울(수)'], ['돈가스김밥', '기숙사모던보울(수)'], ['짬뽕라면', '짜장면'], ['짬뽕라면', '기숙사델리버스(수)'], ['짬뽕라면', '기숙사모던보울(수)'], ['짬뽕라면', '언양식떡갈비덮밥(식당동)'], ['떡만두국', '기숙사모던보울(수)'], ['떡만두국', '기숙사휘당(수)'], ['떡만두국', '교수동(수)'], ['떡만두국', '치즈순두부찌개(식당동)'], ['오징어덮밥', '기숙사모던보울(수)'], ['오징어덮밥', '기숙사휘당(수)'], ['오징어덮밥', '교수동(수)'], ['오징어덮밥', '언양식떡갈비덮밥(식당동)'], ['제육덮밥', '제육덮밥'], ['제육덮밥', '부대찌개'], ['제육덮밥', '짜장면'], ['제육덮밥', '기숙사모던보울(수)'], ['제육덮밥', '기숙사휘당(수)'], ['제육덮밥', '교수동(수)'], ['제육덮밥', '순두부찌개(식당동)'], ['제육덮밥', '치즈순두부찌개(식당동)'], ['제육덮밥', '언양식떡갈비덮밥(식당동)'], ['양푼이비빔밥', '기숙사모던보울(수)'], ['부대찌개', '짜장면'], ['부대찌개', '기숙사델리버스(수)'], ['부대찌개', '기숙사모던보울(수)'], ['부대찌개', '기숙사휘당(수)'], ['부대찌개', '교수동(수)'], ['부대찌개', 'Vegetarian Food(식당동)'], ['부대찌개', '언양식떡갈비덮밥(식당동)'], ['갈비탕', '기숙사델리버스(수)'], ['갈비탕', '언양식떡갈비덮밥(식당동)'], ['설렁탕', '기숙사모던보울(수)'], ['설렁탕', '기숙사휘당(수)'], ['설렁탕', '교수동(수)'], ['설렁탕', '언양식떡갈비덮밥(식당동)'], ['왕돈까스', '기숙사델리버스(수)'], ['치즈돈까스', '기숙사델리버스(수)'], ['고구마돈까스', '기숙사델리버스(수)'], ['짜장면', '짜장면'], ['짜장면', '학생회관한식(수)'], ['짜장면', '기숙사모던보울(수)'], ['짜장면', '기숙사휘당(수)'], ['짜장면', '교수동(수)'], ['짜장면', '순두부찌개(식당동)'], ['짜장면', '치즈순두부찌개(식당동)'], ['짜장면', '소양지쌀국수(식당동)'], ['짜장면', '언양식떡갈비덮밥(식당동)'], ['비빔만두', '언양식떡갈비덮밥(식당동)'], ['소고기국밥', '기숙사모던보울(수)'], ['뼈해장국', '언양식떡갈비덮밥(식당동)'], ['학생회관한식(수)', '기숙사델리버스(수)'], ['학생회관한식(수)', '기숙사모던보울(수)'], ['학생회관한식(수)', '언양식떡갈비덮밥(식당동)'], ['학생회관일품(수)', '언양식떡갈비덮밥(식당동)'], ['학생회관쉐프식(수)', '언양식떡갈비덮밥(식당동)'], ['기숙사델리버스(수)', '기숙사모던보울(수)'], ['기숙사델리버스(수)', '기숙사휘당(수)'], ['기숙사델리버스(수)', '기숙사샐러드(수)'], ['기숙사델리버스(수)', '교수동(수)'], ['기숙사델리버스(수)', '치즈순두부찌개(식당동)'], ['기숙사델리버스(수)', '뼈다귀해장국(식당동)'], ['기숙사델리버스(수)', '수제돈까스(식당동)'], ['기숙사모던보울(수)', '기숙사모던보울(수)'], ['기숙사모던보울(수)', '기숙사휘당(수)'], ['기숙사모던보울(수)', '교수동(수)'], ['기숙사모던보울(수)', '순두부찌개(식당동)'], ['기숙사모던보울(수)', '치즈순두부찌개(식당동)'], ['기숙사모던보울(수)', '소양지쌀국수(식당동)'], ['기숙사모던보울(수)', '언양식떡갈비덮밥(식당동)'], ['기숙사휘당(수)', '기숙사휘당(수)'], ['기숙사휘당(수)', '교수동(수)'], ['기숙사휘당(수)', '순두부찌개(식당동)'], ['기숙사휘당(수)', '치즈순두부찌개(식당동)'], ['기숙사휘당(수)', 'Vegetarian Food(식당동)'], ['기숙사휘당(수)', '언양식떡갈비덮밥(식당동)'], ['교수동(수)', '교수동(수)'], ['교수동(수)', '순두부찌개(식당동)'], ['교수동(수)', '치즈순두부찌개(식당동)'], ['교수동(수)', '언양식떡갈비덮밥(식당동)'], ['순두부찌개(식당동)', '순두부찌개(식당동)'], ['순두부찌개(식당동)', '치즈순두부찌개(식당동)'], ['순두부찌개(식당동)', '언양식떡갈비덮밥(식당동)'], ['치즈순두부찌개(식당동)', '치즈순두부찌개(식당동)'], ['치즈순두부찌개(식당동)', '언양식떡갈비덮밥(식당동)']

=> Diet having Minimum cost : ['참치김밥', '짜장면'], 2500 + 3500 = 6000

Thursday :

', '순두부찌개(식당동)'], ['기숙사휘당(목)', '치즈순두부찌개(식당동)'], ['기숙사휘당(목)', '언양식떡갈비덮밥(식당동)'], ['교수동(목)', '교수동(목)'], ['교수동(목)', '순두부찌개(식당동)'], ['교수동(목)', '치즈순두부찌개(식당동)'], ['교수동(목)', '소양지쌀국수(식당동)'], ['교수동(목)', '언양식떡갈비덮밥(식당동)'], ['순두부찌개(식당동)', '순두부찌개(식당동)'], ['순두부찌개(식당동)', '치즈순두부찌개(식당동)'], ['순두부찌개(식당동)', '언양식떡갈비덮밥(식당동)'], ['치즈순두부찌개(식당동)', '치즈순두부찌개(식당동)'], ['치즈순두부찌개(식당동)', '언양식떡갈비덮밥(식당동)']

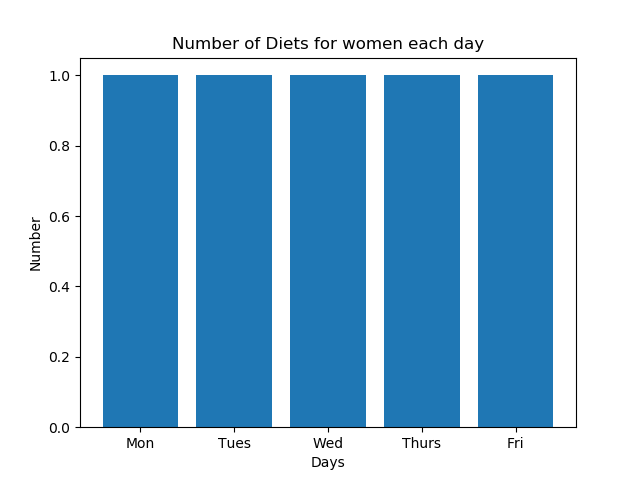
=> Diet having Minimum cost : ['참치김밥', '짜장면'], 2500 + 3500 = 6000

Friday :

['참치김밥', '짜장면'], ['참치김밥', '학생회관한식(금)'], ['치즈라면', '기숙사델리버스(금)'], ['짬뽕라면', '짜장면'], ['짬뽕라면', '학생회관한식(금)'], ['짬뽕라면', '기숙사델리버스(금)'], ['짬뽕라면', '기숙사모던보울(금)'], ['짬뽕라면', '언양식떡갈비덮밥(식당동)'], ['떡만두국', '기숙사모던보울(금)'], ['떡만두국', '기숙사휘당(금)'], ['떡만두국', '치즈순두부찌개(식당동)'], ['오징어덮밥', '기숙사모던보울(금)'], ['오징어덮밥', '언양식떡갈비덮밥(식당동)'], ['제육덮밥', '제육덮밥'], ['제육덮밥', '부대찌개'], ['제육덮밥', '짜장면'], ['제육덮밥', '학생회관한식(금)'], ['제육덮밥', '기숙사모던보울(금)'], ['제육덮밥', '기숙사휘당(금)'], ['제육덮밥', '순두부찌개(식당동)'], ['제육덮밥', '치즈순두부찌개(식당동)'], ['제육덮밥', '언양식떡갈비덮밥(식당동)'], ['양푼이비빔밥', '학생회관한식(금)'], ['부대찌개', '짜장면'], ['부대찌개', '학생회관한식(금)'], ['부대찌개', '기숙사델리버스(금)'], ['부대찌개', '기숙사모던보울(금)'], ['부대찌개', 'Vegetarian Food(식당동)'], ['부대찌개', '언양식떡갈비덮밥(식당동)'], ['갈비탕', '기숙사델리버스(금)'], ['갈비탕', '언양식떡갈비덮밥(식당동)'], ['설렁탕', '학생회관한식(금)'], ['설렁탕', '기숙사모던보울(금)'], ['설렁탕', '언양식떡갈비덮밥(식당동)'], ['짜장면', '짜장면'], ['짜장면', '학생회관한식(금)'], ['짜장면', '기숙사모던보울(금)'], ['짜장면', '기숙사휘당(금)'], ['짜장면', '교수동(금)'], ['짜장면', '순두부찌개(식당동)'], ['짜장면', '치즈순두부찌개(식당동)'], ['짜장면', '소양지쌀국수(식당동)'], ['짜장면', '언양식떡갈비덮밥(식당동)'], ['비빔만두', '기숙사델리버스(금)'], ['비빔만두', '언양식떡갈비덮밥(식당동)'], ['뼈해장국', '언양식떡갈비덮밥(식당동)'], ['학생회관한식(금)', '학생회관한식(금)'], ['학생회관한식(금)', '기숙사모던보울(금)'], ['학생회관한식(금)', '기숙사휘당(금)'], ['학생회관한식(금)', '교수동(금)'], ['학생회관한식(금)', '순두부찌개(식당동)'], ['학생회관한식(금)', '치즈순두부찌개(식당동)'], ['학생회관한식(금)', '소양지쌀국수(식당동)'], ['학생회관한식(금)', '언양식떡갈비덮밥(식당동)'], ['기숙사델리버스(금)', '기숙사모던보울(금)'], ['기숙사델리버스(금)', '기숙사휘당(금)'], ['기숙사델리버스(금)', '교수동(금)'], ['기숙사델리버스(금)', '치즈순두부찌개(식당동)'], ['기숙사델리버스(금)', '언양식떡갈비덮밥(식당동)'], ['기숙사모던보울(금)', '기숙사모던보울(금)'], ['기숙사모던보울(금)', '기숙사휘당(금)'], ['기숙사모던보울(금)', '교수동(금)'], ['기숙사모던보울(금)', '순두부찌개(식당동)'], ['기숙사모던보울(금)', '치즈순두부찌개(식당동)'], ['기숙사모던보울(금)', '언양식떡갈비덮밥(식당동)'], ['기숙사휘당(금)', '기숙사휘당(금)'], ['기숙사휘당(금)', '순두부찌개(식당동)'], ['기숙사휘당(금)', '치즈순두부찌개(식당동)'], ['기숙사휘당(금)', '언양식떡갈비덮밥(식당동)'], ['교수동(금)', '언양식떡갈비덮밥(식당동)'], ['순두부찌개(식당동)', '순두부찌개(식당동)'], ['순두부찌개(식당동)', '치즈순두부찌개(식당동)'], ['순두부찌개(식당동)', '언양식떡갈비덮밥(식당동)'], ['치즈순두부찌개(식당동)', '치즈순두부찌개(식당동)'], ['치즈순두부찌개(식당동)', '언양식떡갈비덮밥(식당동)']

=> Diet having Minimum cost : ['참치김밥', '짜장면'], 2500 + 3500 = 6000

|  |
| --- |
| import pandas as pd  # Monday  df = pd.read\_csv("dite\_Mon.csv", encoding='euc\_kr')  list2 = []  list21 = [[1,1,110000000000]]  list22 = []  for i in range(len(df)):  for j in range(i, len(df)):  if df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate'] >= 220.5 and df.loc[i]['Protein'] + df.loc[j]['Protein'] >= 38.5 and df.loc[i]['Fat'] + df.loc[j]['Fat']>= 36.75:  carb = df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate']  protein = df.loc[i]['Protein'] +df.loc[j]['Protein']  fat = df.loc[i]['Fat'] + df.loc[j]['Fat']  under = carb + protein + fat  ratio\_carb = carb/under  ratio\_protein = protein/under  ratio\_fat = protein/under  list2.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list21[0][2] > int(df.loc[i]['cost'])+ int(df.loc[j]['cost']):  del list21[:]  del list22[:]  list21.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list22.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  elif list21[0][2] == int(df.loc[i]['cost']) + int(df.loc[j]['cost']):  list21.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list22.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list2 == []:  list2.append({'Monday': 'No diet'})  print(list2)  print(list21)  print(list22)  print('=============================')  # Tuesday  df = pd.read\_csv("dite\_Tues.csv", encoding='euc\_kr')  list3 = []  list31 = [[1,1,10000000000]]  list32 = []  for i in range(len(df)):  for j in range(i, len(df)):  if df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate'] >= 220.5 and df.loc[i]['Protein'] + df.loc[j]['Protein'] >= 38.5 and df.loc[i]['Fat'] + df.loc[j]['Fat']>= 36.75:  carb = df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate']  protein = df.loc[i]['Protein'] +df.loc[j]['Protein']  fat = df.loc[i]['Fat'] + df.loc[j]['Fat']  under = carb + protein + fat  ratio\_carb = carb/under  ratio\_protein = protein/under  ratio\_fat = protein/under  list3.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list31[0][2] > int(df.loc[i]['cost'])+ int(df.loc[j]['cost']):  del list31[:]  del list32[:]  list31.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list32.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  elif list31[0][2] == int(df.loc[i]['cost']) + int(df.loc[j]['cost']):  list31.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list32.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list3 == []:  list3.append({'Tuesday': 'No diet'})  print(list3)  print(list31)  print(list32)  print('=============================')  # Wednesday  df = pd.read\_csv("dite\_Wed.csv", encoding='euc\_kr')  list4 = []  list41 = [[1,1,110000000000]]  list42 = []  for i in range(len(df)):  for j in range(i, len(df)):  if df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate'] >= 220.5 and df.loc[i]['Protein'] + df.loc[j]['Protein'] >= 38.5 and df.loc[i]['Fat'] + df.loc[j]['Fat']>= 36.75:  carb = df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate']  protein = df.loc[i]['Protein'] + df.loc[j]['Protein']  fat = df.loc[i]['Fat'] + df.loc[j]['Fat']  under = carb + protein + fat  ratio\_carb = carb/under  ratio\_protein = protein/under  ratio\_fat = protein/under  list4.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list41[0][2] > int(df.loc[i]['cost'])+ int(df.loc[j]['cost']):  del list41[:]  del list42[:]  list41.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list42.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  elif list41[0][2] == int(df.loc[i]['cost']) + int(df.loc[j]['cost']):  list41.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list42.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list4 == []:  list4.append({'Wednesday':'No diet'})  print(list4)  print(list41)  print(list42)  print('=============================')  # Thursday  df = pd.read\_csv("dite\_Thurs.csv", encoding='euc\_kr')  list5 = []  list51 = [[1,1,110000000000]]  list52 =[]  for i in range(len(df)):  for j in range(i, len(df)):  if df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate'] >= 220.5 and df.loc[i]['Protein'] + df.loc[j]['Protein'] >= 38.5 and df.loc[i]['Fat'] + df.loc[j]['Fat']>= 36.75:  carb = df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate']  protein = df.loc[i]['Protein'] +df.loc[j]['Protein']  fat = df.loc[i]['Fat'] + df.loc[j]['Fat']  under = carb + protein + fat  ratio\_carb = carb/under  ratio\_protein = protein/under  ratio\_fat = protein/under  list5.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list51[0][2] > int(df.loc[i]['cost'])+ int(df.loc[j]['cost']):  del list51[:]  del list52[:]  list51.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list52.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  elif list51[0][2] == int(df.loc[i]['cost']) + int(df.loc[j]['cost']):  list51.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list52.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list5 == []:  list5.append({'Thursday':'No diet'})  print(list5)  print(list51)  print(list52)  print('=============================')  # Friday  df = pd.read\_csv("dite\_Fri.csv", encoding='euc\_kr')  list6 = []  list61 = [[1,1,110000000000]]  list62 = []  for i in range(len(df)):  for j in range(i, len(df)):  if df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate'] >= 220.5 and df.loc[i]['Protein'] + df.loc[j]['Protein'] >= 38.5 and df.loc[i]['Fat'] + df.loc[j]['Fat']>= 36.75:  carb = df.loc[i]['Carbonhydrate'] + df.loc[j]['Carbonhydrate']  protein = df.loc[i]['Protein'] +df.loc[j]['Protein']  fat = df.loc[i]['Fat'] + df.loc[j]['Fat']  under = carb + protein + fat  ratio\_carb = carb/under  ratio\_protein = protein/under  ratio\_fat = protein/under  list6.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list61[0][2] > int(df.loc[i]['cost'])+ int(df.loc[j]['cost']):  del list61[:]  del list62[:]  list61.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list62.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  elif list61[0][2] == int(df.loc[i]['cost']) + int(df.loc[j]['cost']):  list61.append([df.loc[i]['cost'], df.loc[j]['cost'], int(df.loc[i]['cost']) + int(df.loc[j]['cost'])])  list62.append([df.loc[i]['Menu'], df.loc[j]['Menu']])  if list6 == []:  list6.append({'Friday':'No diet'})  print(list6)  print(list61)  print(list62) |



# Results & Conclusions

According to our result1 of men, optimal solution is infeasible satisfying the lower limit of total amount of carbohydrate, protein and fat and ratio of them simultaneously. This is because the ratio of fat in a diet is smaller than carbohydrates and proteins. That is, even without considering cost, it is impossible to find a diet that satisfies the ratio of ideal nutrients and the total amount of intake. Therefore, we drive one more optimization value that considers only the total amount, not the ratio and draw the optimal solution in result 2 of men.

In order to satisfy the ideal intake of carbohydrates, proteins and fats as assumed initially, it is recommended to diversify the diets in UNIST or to develop a diet which contains higher fat. Although we could not derive a weekday diet that satisfies all of our constraints, since we defined nutrient lack in overall meal like fat and identified the optimal diet with ideal nutrients minimizing the cost, it is meaningful.

Therefore, it is expected to contribute to academia. Firstly, we can define any nutrient lack in overall meal: what nutrient intake is deficient and where improvements are needed. Second, identifying the optimal diet through an nutritional approach can be a good cornerstone in diet-cost minimization research. In the future, our research will be a good reference when conducting optimization studies on more samples than Unist students.

# Bibliography

1. Ha, A. W., & Kim, W. K. (2019). The food and nutrient intakes from daily processed food in Korean adults: based on the 6th Korea National Health and Nutrition Examination Survey data (2013~ 2015). *Journal of Nutrition and Health*, *52*(5), 422-434.
2. Maillot, M., Ferguson, E. L., Drewnowski, A., & Darmon, N. (2008). Nutrient profiling can help identify foods of good nutritional quality for their price: a validation study with linear programming. The Journal of nutrition, 138(6), 1107-1113.

1. Sobal, J., & Bisogni, C. A. (2009). Constructing food choice decisions. *Annals of Behavioral Medicine*, *38*(suppl\_1), s37-s46.
2. 정효지, 윤진숙, 최슬기, 신상아, 최영선, 권오란, & 장남수. (2009). 코덱스 영양소 기준치 설정시 영양섭취기준의 적용 방안. *한국영양학회지*, *42*(4), 366-373.