Huriye Ceylin Gebeş - 2021400306
Ali Tarık Şahin - 2020400207
Ahmet Fırat Gamsız - 2020400180
Serhan Çakmak - 2020400225

1)

Decision variables:

X_1: Amount of BoBurgers
X_2: Amount of XL BoBurgers
X_3: Amount of Chicken Sandwiches
X_4: Amount of Salads with Chicken
X_5: Amount of French Fries'

Objective function:

minimize $z = X_1 + 3X_2 + 2.50X_3 + 3X_4 + X_5$

Constraints:

Sodium requirement: $480X_1 + 1170X_2 + 800X_3 + 580X_4 + 190X_5 \leq 1100$
Calorie requirement: $250X_1 + 770X_2 + 360X_3 + 190X_4 + 230X_5 \geq 600$
Calorie requirement: $250X_1 + 770X_2 + 360X_3 + 190X_4 + 230X_5 \leq 900$
Protein requirement: $30X_1 + 45X_2 + 15X_3 + 25X_4 + 3X_5 \geq 30$
Calories from fat requirement: $20X_1 - 52X_2 - X_3 + 31X_4 - 8X_5 \geq 0$
Non-negativity: $X_1, X_2, X_3, X_4, X_5 \geq 0$

Explanations:

Since the daily sodium intake of the human body should be less than 2200 mg per day and the management requires that the meal has less than 50% of the daily sodium requirement, the meal should have less than 1100 mg of sodium.

Similarly, the management requires that the meal has more than 600 calories and less than 900 calories, which are written as separate constraints.

The management also requires that the meal has at least 30 grams of protein, which is written as a protein requirement.

If the calories from fat should be at most %40 of the total calories, the following calculations will determine or constraint:

1. Calories from fat for the meal: $80X_1 + 360X_2 + 145X_3 + 45X_4 + 100X_5$
2. 40% of the total calories for the meal: $40.(250X_1 + 770X_2 + 360X_3 + 190X_4 + 230X_5)/100 \longrightarrow 100X_1 + 308X_2 + 144X_3 + 76X_4 + 92X_5$

Combining 1. and 2.: $80X_1 + 360X_2 + 145X_3 + 45X_4 + 100X_5 \leq 100X_1 + 308X_2 + 144X_3 + 76X_4 + 92X_5$
$\longrightarrow 20X_1 - 52X_2 - X_3 + 31X_4 - 8X_5 \geq 0$

The amount of food that will be in the meal cannot be negative, therefore non-negativity constraint is included.

2) This constraint is not a valid constraint for LP because it contains strictly less than (<) operator. Strictly less than (<) or strictly greater than (>) operators can lead to an unbounded solution space, making the LP problem infeasible or unbounded.

In LP, constraints should be written as linear equations or inequalities. The less than or equal to (≤) constraint and the greater than or equal to (≥) constraint are used because they define bounded regions. These constraints create a feasible solution space, which is necessary for LP.

If we use strictly less than operator, this could result in a solution space where there is no feasible solution that satisfies all the constraints. It may lead to an empty or unbounded solution space since we can get close to the constraint infinitely close but we cannot reach an exact value, making the LP problem unsolvable.

3)
min z =| x_1 - 2|+| x_2 |+| x_1 |+| x_2 + 3|

We can’t have absolute values in LP. They are not linear functions. We define new variables and their corresponding constraints to eliminate absolute values

Let: a, b,c, d

$a >= x_1 - 2$
$a >= -(x_1 - 2) \Rightarrow a >= -x_1 + 2$

$b >= x_2$
$b >= -x_2$

$c >= x_1$
$c >= -x_1$

$d >= x_2 + 3$
$d >= -(x_2 + 3) \Rightarrow d >= -x_2 - 3$

Model the LP with new variables and rearrange the inequalities to the appropriate form.
Decision variables: x_1, x_2, a, b, c, d
Objective function: minimize z= a + b + c + d
Constraints: subject to
$a - x_1 >= -2$
$a + x_1 >= 2$
$b - x_2 >= 0$
$b + x_2 >=0$
$c - x_1 >=0$
$c + x_1 >=0$
$d - x_2 >= 3$
$d + x_2 >= -3$

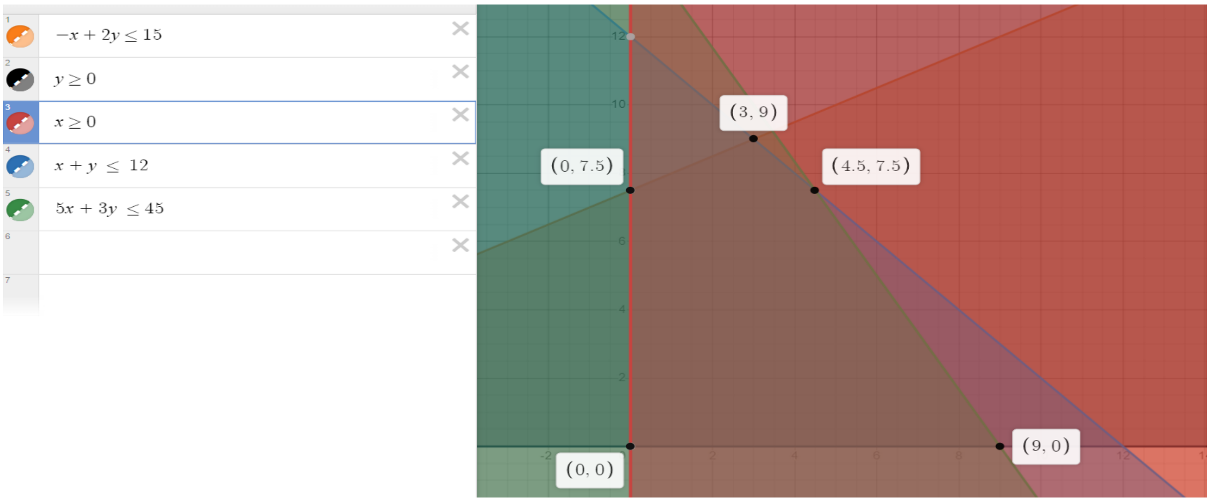
Non-negativity constraint: x_1 , x_2 >=0

Please note that including non-negativity constraints for a, b, c, and d is redundant because the way they are defined (their corresponding constraints enforces them to be non-negative.
For example if $x_2 < 0$ then $b >= -x_2 \Rightarrow$ b is positive
if $x_2 = 0$ then $b >= 0 \Rightarrow$ b is non-negative
if $x_2 > 0$ then $b >= x_2 \Rightarrow$ b is positive

All cases prove that x_2 is non-negative and this check can be applied for other variables too.

4)

To initiate the analysis, it is crucial to visually depict the constraints on a graph and pinpoint their intersections. This area where constraints overlap is denoted as the feasible region.



Intersections defining corners of the feasible region:

$-x_1+2x_2$ <=15 intersected with x_1 >= 0 at (0, 7.5)

$-x_1+2x_2$ <=15 intersected with x_1+ x_2 <= 12 at (3, 9)

x_1+ x_2 <=12 intersected with $5x_1$ + $3x_2$ <= 45 at (4.5, 7.5)

$5x_1$ + $3x_2$ <=45 intersected with x_1 >= 0 at (9, 0)

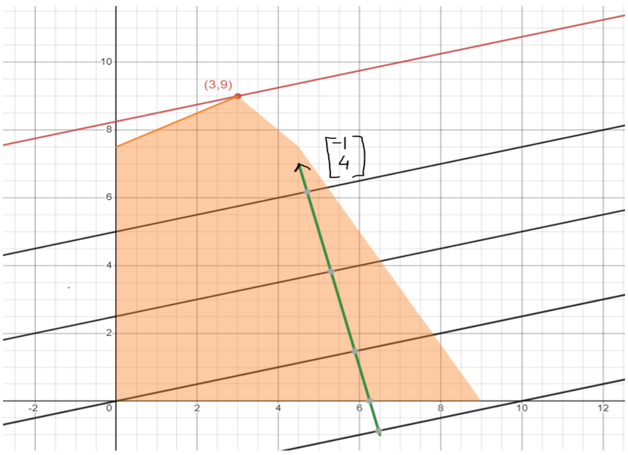
x_1>=0 intersected with x_2 >=0 at (0, 0)

The objective function, max z = $-5x_1$ + $20x_2$, implies a corresponding cost vector in the vector space, denoted as [4, 1].

Determining the minimization vector requires understanding the direction orthogonal to the cost function. By observing how the cost function increases when moving along the direction (the maximization direction) [-1, 4], it becomes evident that the minimization vector is [1, -4].

After visually depicting the feasible region and identifying the direction vector for maximization, the subsequent step entails traversing along isocost lines in the maximization direction. This journey commences from negative infinity and continues until the point where the maximum z value within the feasible region is discovered.

In this case, the intersection occurs at the point (3, 9), yielding a maximum z value of 165.



The green arrow indicates the maximization direction.

5)

a)

Decision variables:

A: The amount invested in bond A in million dollars
B: The amount invested in bond B in million dollars
C: The amount invested in bond C in million dollars
D: The amount invested in bond D in million dollars
E: The amount invested in bond E in million dollars

Objective function:

max 4.3/100*A + 5.4/100*B*½ + 5/100*C*½ + 4.4/100*D*½ + 4.5/100*E

Explanation: Yield to maturity is the field we must focus on since we want to maximize profits. Also, we need to take taxes into account. According to the information given below the question, only municipality bonds are tax-free, so the other profits are taxed. That's why we need to reduce the tax amount from profit by multiplying by 50% (1/2).

Constraints:

Target investment amount: A + B + C + D + E = 10 (in million dollars)
Government and agency bond restrictions: B + C + D ≥ 4 (in million dollars)
Rating of portfolio restriction: 2*A/10 + 2*B/10 + C/10 + D/10 + 5*E/10 ≤ 1.4
Years to maturity of the portfolio restriction: 9*A/10 + 15*B/10 + 4*C/10 + 3*D/10 + 2*E/10 ≤ 5
Non-negativity: A, B, C, D, E ≥ 0

b)

We introduce a new variable called x. x represents the extra amount invested in million dollars and it is up to 1 million dollars. So, it will take its place both in restrictions and in objective function because there will be a repayment of the borrowed amount.

Objective function:

max 4.3/100*A + 5.4/100*B*½ + 5/100*C*½ + 4.4/100*D*½ + 4.5/100*E - 5.5/100*x

Constraints:

Target investment amount: A + B + C + D + E = 10 + x (in million dollars)
Government and agency bond restrictions: B + C + D ≥ 4 (in million dollars)
Rating of portfolio restriction: 2*A/(10+x) + 2*B/(10+x) + C/(10+x) + D/(10+x) + 5*E/(10+x) ≤ 1.4
Years to maturity of the portfolio restriction: 9*A/(10+x) + 15*B/(10+x) + 4*C/(10+x) + 3*D/(10+x) + 2*E/(10+x) ≤ 5
Borrowed amount restriction: x ≤ 1 (in million dollars)
Non-negativity: A, B, C, D, E, x ≥ 0

6) Our algorithm simply checks all the feasible solutions in the solution space to find the optimal point. We utilized the fact that constraints are integers and implemented nested loops to check all the points that follow the number of hr/week constraints (x, y, z, a are the variables that correspond to the number of hr/week of processors 1,2,3 and 4 respectively and b is number of hours of overtime). We checked if the current point is in the solution space by using other constraints as boolean expressions. We initialized the minimum point to infinite (sys.maxsize in Python provides a very large number that can be regarded as infinite) and then updated with a more optimal solution when we found one in the solution space. Also, we recorded the point that yields a new optimal point. When the loops are completed we had the optimal solution and the corresponding point in the space.

The minimum value in the first part is 33225 and the variables that yield the result are (x=23, y=15, z=9, a=0).

For the second part when overtime is allowed, we got a different solution. The minimum value has become 32400 and the variables are (x=19, y=18, z=9, a=0, b=10).

Source code can be found in hw1.py file.