Question 1

Consider the following table indicating the nutrtional value of different food types:

Foods	Price ($) / Serving	Cal (g) / Serving	Fat (g) / Serving	Protein (g) / Serving	Carbs (g) / Serving
Raw carrots	0.14	23	0.1	0.6	6
Baked potatoes	0.12	171	0.2	3.7	30
Wheat bread	0.2	65	0	2.2	13
Cheddar cheese	0.75	112	9.3	7	0
Peanut butter	0.15	188	16	7.7	2

You need to decide how many servings of each food to buy each day so that you minimize the total cost of buying your food while satisfying the following daily nutritional requirements:

- Calories must be at least 2000,
- Fat must be at least 50g,
- Protein must be at least 100g,
- Carbohydrates must be at least 250g.

Formulate an LP to determine how many servings of each fof the aforementioned foods meet all of the nutritional requirements, while minimizing the total cost of food.

Optimization Problem:

$$
\begin{aligned}
\min_{x} \quad & z = 0.14carrots + 0.12potatoes + 0.2bread + 0.75cheese + 0.15peanutbutter \\
\text{s.t.} \quad & 23carrots + 171potatoes + 65bread + 112cheese + 188peanutbutter \geq 2000, \\
& 0.1carrots + 0.2potatoes + 9.3cheese + 16peanutbutter \geq 50, \\
& 0.6carrots + 3.7potatoes + 2.2bread + 7cheese + 7.7peanutbutter \geq 100 \\
& 6carrots + 30potatoes + 13bread + 2peanutbutter \geq 250 \\
& carrots, potatoes, bread, cheese, peanutbutter \geq 0.
\end{aligned}
$$

In [ ]:

```
using JuMP
using HiGHS

# definining the model and relevant optimizer
diet = Model(HiGHS.Optimizer);
```

In [ ]:
```
# defining the variables of the optimization problem
@variable(diet, carrots >= 0);
@variable(diet, potatoes >= 0);
@variable(diet, bread >= 0);
@variable(diet, cheese >= 0);
@variable(diet, peanut_butter >= 0);
```

In [ ]:
```
# defining constraints for the model
@constraint(diet, constraint1, 23carrots + 171potatoes + 65bread + 112cheese +
@constraint(diet, constraint2, 0.1carrots + 0.2potatoes + 9.3cheese + 16peanu
@constraint(diet, constraint3, 0.6carrots + 3.7potatoes + 2.2bread + 7cheese +
@constraint(diet, constraint4, 6carrots + 30potatoes + 13bread + 2peanut_butt
```

In [ ]:
```
# defining the objective function for the model
@objective(diet, Min, 0.14carrots + 0.12potatoes + 0.2bread + 0.75cheese + 0.
```

In [ ]:
```
# solving the model
optimize!(diet);

# outputs detailed information about the solution process
@show solution_summary(diet);
```

```
Presolving model
4 rows, 4 cols, 15 nonzeros
4 rows, 2 cols, 8 nonzeros
Presolve : Reductions: rows 4(-0); columns 2(-3); elements 8(-10)
Solving the presolved LP
Using EKK dual simplex solver - serial
  Iteration        Objective     Infeasibilities num(sum)
          0     0.0000000000e+00 Pr: 4(975) 0s
          3     2.3177549195e+00 Pr: 0(0) 0s
Solving the original LP from the solution after postsolve
Model   status      : Optimal
Simplex   iterations: 3
Objective value     :  2.3177549195e+00
HiGHS run time      :          0.03
solution_summary(diet) = * Solver : HiGHS

* Status
  Termination status : OPTIMAL
  Primal status      : FEASIBLE_POINT
  Dual status        : FEASIBLE_POINT
  Message from the solver:
  "kHighsModelStatusOptimal"

* Candidate solution
  Objective value      : 2.31775e+00
  Objective bound      : 0.00000e+00
  Relative gap         : Inf
  Dual objective value : 2.31775e+00

* Work counters
  Solve time (sec)   : 3.41598e-02
  Simplex iterations : 3
  Barrier iterations : 0
  Node count         : -1
```