Name: ____________________________________

Quiz 3 *MAT 334 Intermediate Deterministic Models* 3 February 2026

You may use your own **hard-copy** portion of IDM homework #3 only.
You may not use any other sources.
There is no computer portion of this quiz. All answers are to be written on this sheet.

---

1. (10 points) A traveler named Terry is bringing some things along on a trip, packing them in identical pouches. Each pouch has the same weight capacity. Terry formulated this packing challenge as a **bin packing problem** and started on a PuLP/Python implementation shown on the next page. **Fill in all blanks** in the implementation up to the point of solving. Do not worry about the code to display results. Note:

   - all data structure you need are already in the code and you should use those
   - `LpVariable.dicts` can take various Python containers to use for indexing (e.g. list, set, range).

SEE IMPLMENTATION WITH BLANKS ON THE OTHER SIDE

```
from pulp import *
things = ['Shirt', 'Pants', 'Socks', 'Laptop', 'Charger', 'Snacks', 'Jacket']
lbs = [0.4, 0.8, 0.2, 2.8, 0.3, 0.7, 1.0]
lbs_dict = dict(zip(things,lbs))  # = {'Shirt': 0.4, 'Pants': 0.8, etc. }
max_pouch_wt = 3

pack_prob = LpProblem('Travel-Packing-Packing-Problem', LpMinimize)

num_things = len(things)
num_pouches = round(2 * sum(lbs) / max_pouch_wt)

pouches = range(1, num_pouches + 1)

# Decision variables
# assignment of things to pouches variable:
#
x = LpVariable.dicts('x', (_______________, ___________), cat=_______________)

# pouch-used variable:
y = LpVariable.dicts('y', _______________, cat = ___________)

# objective function
obj = _____________________________________________________, 'min # of pouches'
pack_prob += obj

# constraints
#
for __________ in __________:
    thing_in_pouch = __________________________________________________
    pack_prob += thing_in_pouch , f'thing {thing} goes in a pouch'

for __________ in __________:
    lhs = lpSum(lbs_dict[thing] * _______________________________

      for __________________ in ____________________)

    pack_prob += lhs <= ________________________ * ________________________, \
        f'pouch {pouch} capacity constraint'

# Solve
result = pack_prob.solve(PULP_CBC_CMD(msg=False))
```