Name: _____

Descent into code

9. (10 points) Sto Chastic is a student taking 6.036 this semester, and he prepared dilligently for the midterm exam. Unfortunately, his carefully prepared one-page of notes got eaten by a shredder, and now he needs your help derandomizing lines to answer the two questions below.

   The available lines (each prefaced with a letter, as an identifier) are:

   A: `n = y.shape[1]`
   B: `d = y.shape[0]`
   C: `j = np.random.randint(n)`
   D: `j = np.random.randint(d)`
   E: `Xj = X[j:j+1, :]`
   F: `Xj = X[:, j:j+1]`
   G: `yj = y[j:j+1, :]`
   H: `yj = y[:, j:j+1]`
   I: `th = th0`
   J: `th = th - step_size_fn(k) * dJ(Xj, yj, th)`
   K: `th = th + step_size_fn(k) * dJ(Xj, yj, th)`
   L: `th = th - step_size_fn(k) * dJ(th)`
   M: `th = th + step_size_fn(k) * dJ(th)`

   (a) Fill in the blanks below, to give correct python code implementing gradient descent as a function `gd(dJ, th0, step_size_fn, num_steps)` which takes as arguments

   - `dJ`: a function which takes as input the vector of model parameters `th`, and outputs the gradient $dJ/d\theta$ of the objective function J at $\theta =$ `th`.
   - `th0`: an initial value of model parameter vector θ, a column vector.
   - `step_size_fn`: a function that is given the iteration index (an integer) and returns a step size parameter.
   - `num_steps`: the number of iterations to perform

   The `gd` function should return the value of the model parameter vector at the final step.

   Fill in each blank with one letter (**A**, **B**, ...), corresponding to one of the available lines listed above, from Sto Chastic's notes.

```
1. def gd(dJ, th0, step_size_fn, num_steps):

2.       I: th = th0
3.     for k in range(num_steps):

4.         L: th = th - step_size_fn(k) * dJ(th)
5.     return th
```

Name: _____

(b) Fill in the blanks below, to give correct python code implementing *stochastic* gradient descent as a function `sgd(X, y, dJ, th0, step_size_fn, num_steps)` which takes as arguments

- `X`: a standard $d \times n$ data array
- `y`: a standard $1 \times n$ row vector of labels
- `dJ`: a function which takes as input a data point (column vector), a label (1×1), and a vector of model parameters `th`, and outputs the gradient $dJ/d\theta$ of the objective function J for the given data point and label evaluated at the given model parameters.
- `th0`: an initial value of model parameter vector θ, a column vector.
- `step_size_fn`: a function that is given the iteration index (an integer) and returns a step size parameter.
- `num_steps`: the number of iterations to perform

The `sgd` function should return the value of the model parameter vector at the final step.

Fill in each blank with one letter (**A**, **B**, ...), corresponding to one of the available lines listed above, from Sto Chastic's notes.

```
1. def sgd(X, y, dJ, th0, step_size_fn, num_steps):
2.     th = th0

3.     A: n = y.shape[1]
4.     for k in range(num_steps):

5.         C: j = np.random.randint(n)

6.         F: Xj = X[:, j:j+1]

7.         H: yj = y[:, j:j+1]

8.         J: th = th - step_size_fn(k) * dJ(Xj, yj, w)
9.     return th
```