# 1 Problem 1.

You are working on a simulation project where you must generate random samples from a target distribution defined as:

$$p(x) = 5x^4/1, \quad \text{for } x \in [0, 1]$$

Since direct sampling from this distribution is difficult, you decide to use **Rejection Sampling** with a uniform proposal distribution $q(x) = 1$ over the interval $[0, 1]$.

## Part A:

Explain how rejection sampling works and describe, in your own words:

- The purpose of the proposal distribution $q(x)$
- The role of the constant $M$
- The logic behind accepting or rejecting a sample

Provide a short description of the algorithm.

## Part B:

1. Compute the maximum value of $p(x)/q(x)$ on the interval $[0, 1]$ and determine the smallest valid constant $M$.

## Part C:

Write a Python-style pseudo-code to generate 1000 samples using rejection sampling based on the distributions above.

# 2 Problem 2.

You are working with a Bayesian network that models the following dependency structure:

$$A \to B \to C$$

Each variable is binary: $A, B, C \in \{0, 1\}$. You are given the following conditional probability tables (CPTs):

- $P(A = 1) = 0.6$, $P(A = 0) = 0.4$
- $P(B = 1 \mid A = 1) = 0.7$, $P(B = 1 \mid A = 0) = 0.2$
- $P(C = 1 \mid B = 1) = 0.9$, $P(C = 1 \mid B = 0) = 0.3$

## Part A:

Simulate **one complete sample** $(A, B, C)$ using ancestral sampling. Use the following uniform random numbers:

- For $A$: $u_1 = 0.65$
- For $B$: $u_2 = 0.25$
- For $C$: $u_3 = 0.85$

State the value sampled for each variable and justify based on the provided CPTs.

## Part B:

You are given this ancestral sampling code:

```
import numpy as np

def ancestral_sample():
    u1 = np.random.rand()
    A = 1 if u1 < 0.6 else 0

    u2 = np.random.rand()
    if A == 1:
        B = 1 if u2 < 0.7 else 0
    else:
        B = 1 if u2 < 0.2 else 0

    u3 = np.random.rand()
    if B == 0:
        C = 1 if u3 < 0.9 else 0
    else:
        C = 1 if u3 < 0.3 else 0

    return A, B, C
```

1. Identify and fix the bug(s) in the code. Explain how it affects correctness.
2. How would this approach change for a DAG with 100 nodes and multiple parents per node?

# 3 Problem 3.

You are working with a bivariate joint distribution over variables $X$ and $Y$, defined as follows:

$$P(X, Y) \propto \exp\left(-\frac{1}{2}(x^2 + y^2 + 2\rho xy)\right)$$

where $\rho \in (-1, 1)$ is the correlation parameter. Assume $\rho = 0.6$. This distribution is a non-standard bivariate Gaussian-like form (not normalized).

## Part A:

1. Explain how Gibbs Sampling works using the concept of conditional distributions.
2. In the above joint distribution, why would Gibbs Sampling be easier to use than Rejection Sampling?

## Part B:

You are told that the conditional distributions for $X$ and $Y$ given the other are:

$$X \mid Y = y \sim \mathcal{N}(-0.6y,\ 0.8) \qquad Y \mid X = x \sim \mathcal{N}(-0.6x,\ 0.8)$$

Assume an initial value of $(x_0, y_0) = (0, 0)$.
Using the following standard normal samples:

$$z_1 = 0.5, \quad z_2 = -0.2$$

Simulate one full Gibbs iteration:

1. Sample $x_1 \sim P(X \mid Y = y_0)$
2. Sample $y_1 \sim P(Y \mid X = x_1)$
3. State the final values $(x_1, y_1)$

# 4 Problem 4.

You are modeling how likely a person is to **soft launch** their relationship on Instagram.
The decision depends on their daily **vibe level** $x \in [0, 10]$, which is influenced by:

- Whether they got their matcha latte this morning
- How chaotic their group chat was last night
- And how delulu (delusional) they feel — because if they believe it's real, it might as well be

Data shows that students are most likely to post when their vibe level is around 8. You model their likelihood to post with the following unnormalized probability:

$$p(x) \propto \exp\left(-\frac{(x-8)^2}{2}\right)$$

To simulate daily changes in vibe, you use the **Metropolis algorithm** with a symmetric proposal distribution:

$$x' = x + \epsilon, \quad \epsilon \sim \mathcal{N}(0, 1)$$

## Part A:

1. What does the proposal distribution represent in this simulation?
2. Why doesn't the algorithm always accept higher-vibe values?
3. Why is symmetry in the proposal distribution important?

## Part B:

Assume the current vibe level is $x = 6.5$, and the algorithm proposes $x' = 7.8$.

Use the Metropolis acceptance ratio formula:

$$r = \exp\left(\frac{(x-8)^2 - (x'-8)^2}{2}\right)$$

1. Compute the acceptance ratio $r$
2. If a uniform random number $u = 0.2$, does the student accept the proposal?
3. What is the final decision: stay at $x$, or move to $x'$?

## Part C:

1. What might happen if the proposal distribution has too large a variance?
2. How can you determine whether the simulation has converged?
3. Why is it beneficial for the algorithm to sometimes reject proposals?