Pintero, Adrian Joriz G.
NW-301

Q1) Fee Calculation (Service code = 1) For the Consultation service, the base price is 10,000. I calculated the 12% tax as 1,200.

- **Total Service Fee:** 11,200

Q2) Fee Calculation (Service code = 2) For the Documents service, the math involves a service charge first.

- Base: 2,000
- 10% Service Charge: 200 (making it 2,200)
- 12% Tax on 2,200: 264
- **Total Service Fee:** 2,464

Q3) Message Construction (Service code = 1) Using the name Maria Santos Cruz:

- First letter of First Name: **M**
- Last letter of Middle Name: **s**
- First letter of Last Name: **C**
- Service code: **1**
- First digit of total fee (from 11,200): **1**
- **Resulting Message:** MsC11

Q4) Message Construction (Service code = 2) Using the name Maria Santos Cruz:

- First letter of First Name: **M**
- Last letter of Middle Name: **s**
- First letter of Last Name: **C**
- Service code: **2**
- First digit of total fee (from 2,464): **2**
- **Resulting Message:** MsC22

Q5) Hashing Choice

- a) Service code 1: **abi.encodePacked**
- b) Service code 2: **abi.encode**

Q6) Concept Check

- **Answer: False**.
- *Explanation:* They don't always produce the same bytes because abi.encode adds padding to align data to 32 bytes, while abi.encodePacked compresses everything together without padding.

Q7) Collision Thought Exercise The risk with abi.encodePacked is called a "hash collision." Since it doesn't include any information about where one string ends and the next begins, different inputs can end up looking identical. For example, if you pack "A" and "BC", it looks like "ABC". If you pack "AB" and "C", it also looks like "ABC". This causes them to have the same hash even though the original data was different. abi.encode is safer because it keeps the data distinct.