Aim-

calculate the message digest of a text using the SHA-1 algorithm in python

Program-

```python
import hashlib

def calculate_sha1_digest(text):
    """
    Calculates the SHA-1 digest of the given text.
    :param text: The input text (string).
    :return: The SHA-1 hash digest in hexadecimal format.
    """
    # Encode the text to bytes
    text_bytes = text.encode('utf-8')

    # Create a SHA-1 hash object
    sha1 = hashlib.sha1()

    # Update the hash object with the text bytes
    sha1.update(text_bytes)

    # Get the hexadecimal digest
    digest = sha1.hexdigest()

    return digest

# Example usage
input_text = "Hello, World!"
sha1_digest = calculate_sha1_digest(input_text)

print(f"Input Text: {input_text}")
print(f"SHA-1 Digest: {sha1_digest}")
```

Explaination-

Let's break down the SHA-1 digest calculation code line by line.

Full Code

```
python
Copy code
import hashlib
```

- **Line Purpose**: This imports the `hashlib` module, which provides functions for implementing various hash algorithms, including **SHA-1**.

```
python
Copy code
def calculate_sha1_digest(text):
    """
    Calculates the SHA-1 digest of the given text.
    :param text: The input text (string).
    :return: The SHA-1 hash digest in hexadecimal format.
    """
```

- **Line Purpose**: This defines a function named `calculate_sha1_digest` that takes a string parameter `text`.
- **Docstring**: It explains the function's purpose (calculating a SHA-1 hash) and its input (`text`) and output (SHA-1 hash in hexadecimal format).

```
python
Copy code
    # Encode the text to bytes
    text_bytes = text.encode('utf-8')
```

- **Explanation**:
  - SHA-1 works on binary data (bytes). Since the input `text` is a string, it must first be converted to bytes.
  - `.encode('utf-8')` encodes the string into a sequence of bytes using the UTF-8 character encoding.
  - Example:

- Input: `"Hello, World!"`
- Output: `b'Hello, World!'` (bytes representation of the text).

```
python
Copy code
    # Create a SHA-1 hash object
    sha1 = hashlib.sha1()
```

- **Explanation**:
  - `hashlib.sha1()` creates a new hash object configured to use the **SHA-1 algorithm**.
  - This object is used to compute the hash step by step by feeding it data (like the input text in bytes).

```
python
Copy code
    # Update the hash object with the text bytes
    sha1.update(text_bytes)
```

- **Explanation**:
  - `.update(data)` is a method used to add data (in bytes) to the hash object.
  - Internally, SHA-1 processes the data in fixed-size chunks (512 bits). If the input is longer, it will automatically break it into chunks and process each one.

```
python
Copy code
    # Get the hexadecimal digest
    digest = sha1.hexdigest()
```

- **Explanation**:
  - `.hexdigest()` calculates the final hash value after all data has been fed into the hash object.

    - The result is a **hexadecimal string** representation of the hash, which is commonly used for readability.
    - Example:
        - For input `"Hello, World!"`, the digest will be: `d3486ae9136e7856bc42212385ea797094475802`.

```
python
Copy code
    return digest
```

- **Explanation**:
    - This returns the hexadecimal string representation of the SHA-1 hash digest.

```
python
Copy code
# Example usage
input_text = "Hello, World!"
sha1_digest = calculate_sha1_digest(input_text)
```

- **Explanation**:
    - `input_text` is set to the example text `"Hello, World!"`.
    - `calculate_sha1_digest` is called with `input_text` as an argument, and its result (the SHA-1 digest) is stored in `sha1_digest`.

```
python
Copy code
print(f"Input Text: {input_text}")
print(f"SHA-1 Digest: {sha1_digest}")
```

- **Explanation**:
    - These lines print the input text and the calculated SHA-1 digest to the console.

- The `f` before the string enables **f-string formatting**, allowing variables to be inserted directly into the string using curly braces (`{}`).

Example Output

If the input text is `"Hello, World!"`, the output will be:

```
mathematica
Copy code
Input Text: Hello, World!
SHA-1 Digest: d3486ae9136e7856bc42212385ea797094475802
```

Key Concepts

1. **SHA-1 Algorithm**:
   - A cryptographic hash function that produces a 160-bit (20-byte) hash value.
   - Commonly used for data integrity and verification.
2. **Why Bytes?**
   - Hashing algorithms operate on binary data (bytes), not strings.
3. **Hex Digest**:
   - The `hexdigest` method converts the binary hash output into a readable hexadecimal format, making it easier to compare or store.