Jovenry Lloyd L. Villa-or **202107461**
BSIT4-1

In this activity, we are tasked to create a program that will show how to encode and decode messages using Base64. The encode_base64 function changes the message into binary, splits it into 6-bit chunks, and matches each chunk with a Base64 character. It adds = padding to make the result a multiple of 4 characters. The decode_base64 function reverses this by removing padding, converting Base64 characters back to binary, and grouping the binary into 8-bit chunks to recreate the original message. This program helped me understand how Base64 works and how padding ensures the data stays correct.

CODE:

```
: > Users > NEW WINDOW 10 > Downloads > lab2.py > ...
1   base64_chars = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"
2
3   def encode_base64(data):
4       binary_data = ''.join(f"{ord(char):08b}" for char in data)
5       padding_needed = (6 - len(binary_data) % 6) % 6
6       binary_data += '0' * padding_needed
7       encoded_data = ''.join(base64_chars[int(binary_data[i:i+6], 2)] for i in range(0, len(binary_data), 6))
8       padding_chars = '=' * ((4 - len(encoded_data) % 4) % 4)
9       return encoded_data + padding_chars
10
11  def decode_base64(encoded_data):
12      padding_chars = encoded_data.count('=')
13      encoded_data = encoded_data.rstrip('=')
14      binary_data = ''.join(f"{base64_chars.index(char):06b}" for char in encoded_data)
15      binary_data = binary_data[:len(binary_data) - padding_chars * 2]
16      decoded_data = ''.join(chr(int(binary_data[i:i+8], 2)) for i in range(0, len(binary_data), 8))
17      return decoded_data
18
19  if __name__ == "__main__":
20      # Ask the user to input a message
21      user_input = input("Enter the message to encrypt and decrypt: ")
22
23      # Encrypting the data (Base64 Encoding)
24      encrypted_data = encode_base64(user_input)
25      print(f"Encrypted Data: {encrypted_data}")
26
27      # Decrypting the data (Base64 Decoding)
28      decrypted_data = decode_base64(encrypted_data)
29      print(f"Decrypted Data: {decrypted_data}")
30
```

OUTPUT:

```
Enter the message to encrypt and decrypt: AB
Encrypted Data: QUI=
Decrypted Data: AB
```