Assignment 2

```python
import heapq
from collections import defaultdict

class HuffmanNode:
    def __init__(self, char, freq):
        self.char = char
        self.freq = freq
        self.left = None
        self.right = None

    # Implementing comparison operators for heapq
    def __lt__(self, other):
        return self.freq < other.freq

    def __eq__(self, other):
        return self.freq == other.freq

def build_huffman_tree(freq_table):
    priority_queue = [HuffmanNode(char, freq) for char, freq in freq_table.items()]
    heapq.heapify(priority_queue)

    while len(priority_queue) > 1:
        left_node = heapq.heappop(priority_queue)
        right_node = heapq.heappop(priority_queue)

        merged_freq = left_node.freq + right_node.freq
        merged_node = HuffmanNode(None, merged_freq)
        merged_node.left = left_node
        merged_node.right = right_node

        heapq.heappush(priority_queue, merged_node)

    return priority_queue[0]

def build_huffman_codes(node, code='', code_dict={}):
    if node is None:
        return

    if node.char is not None:
        code_dict[node.char] = code

    build_huffman_codes(node.left, code + '0', code_dict)
    build_huffman_codes(node.right, code + '1', code_dict)

def encode_text(text, code_dict):
    encoded_text = ''.join(code_dict[char] for char in text)
    return encoded_text

def main():
    text = input("Enter the text to be encoded: ")

    freq_table = defaultdict(int)
    for char in text:
        freq_table[char] += 1

    huffman_tree = build_huffman_tree(freq_table)
    huffman_codes = {}
    build_huffman_codes(huffman_tree, '', huffman_codes)
    encoded_text = encode_text(text, huffman_codes)


    print("Original text:", text)
    print("Encoded text:", encoded_text)
    print("Huffman codes:", huffman_codes)


if __name__ == "__main__":
    main()
```

Output:

```
Enter the text to be encoded: hello world
Original text: hello world
Encoded text: 1110000101011011110111001010001
Huffman codes: {'e': '000', 'd': '001', 'r': '010', 'w': '011', 'l': '10', 'o': '110', 'h': '1110', ' ': '1111'}
```