NAME:- MARVIN PATEL
ER NO :- 22162101013
SUB :- AAD

PRACTICAL - 10

Institute of Computer Technology
B. Tech Computer Science and Engineering

Sub: Algorithm Analysis and Design

Practical 10

Huffman coding assigns variable length code words to fixed length input characters based on their frequencies. More frequent characters are assigned shorter code words and less frequent characters are assigned longer code words. All edges along the path to a character contain a code
digit. If they are on the left side of the tree, they will be a 0 (zero). If on the right, they&#39;ll be a 1
(one). Only the leaves will contain a letter and its frequency count. All other nodes will contain a null instead of a character, and the count of the frequency of all of it and its descendant characters.
Construct the Huffman tree for the following data and obtain its Huffman code.

Characters A B C D E -

Frequency/Probability 0.5 0.35 0.5 0.1 0.4 0.2

(i) Encode text CAD-BE using the above code.
Input: CAD-BE
Output: 10011100110111100
(ii) Decode the text 1100110110 using the above information.
Input: 0011011100011100
Output: E-DAD

Python code:-

```
from flask import Flask, render_template, request
import heapq
from collections import namedtuple

# Define a named tuple for tree nodes
class Node:
    def __init__(self, frequency, character=None, left=None,
right=None):
        self.frequency = frequency
```

NAME:- MARVIN PATEL
ER NO :- 22162101013
SUB :- AAD

PRACTICAL - 10

```
        self.character = character
        self.left = left
        self.right = right

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


app = Flask(__name__)

def build_huffman_tree(characters, frequencies):
    # Create a list of nodes from characters and frequencies
    nodes = [Node(frequency, character) for character, frequency
in zip(characters, frequencies)]
    heapq.heapify(nodes)  # Convert the list into a min-heap

    while len(nodes) > 1:
        # Remove two nodes with the lowest frequency
        left = heapq.heappop(nodes)
        right = heapq.heappop(nodes)
        # Create a new node with combined frequency
        merged_node = Node(left.frequency + right.frequency,
None, left, right)
        heapq.heappush(nodes, merged_node)

    return nodes[0]  # Root of the Huffman Tree

def generate_huffman_codes(node, code, mapping):
    if node.character is not None:  # Leaf node
        mapping[node.character] = code
    else:
        generate_huffman_codes(node.left, code + '0', mapping)
```

NAME:- MARVIN PATEL
ER NO :- 22162101013
SUB :- AAD

PRACTICAL - 10

```
        generate_huffman_codes(node.right, code + '1', mapping)

@app.route("/", methods=["GET", "POST"])
def index():
    result = None
    if request.method == "POST":
        try:
            characters = request.form["characters"].split()
            frequencies = list(map(float,
request.form["frequencies"].split()))
            # Build Huffman tree
            huffman_tree = build_huffman_tree(characters,
frequencies)
            # Generate Huffman codes
            huffman_mapping = {}
            generate_huffman_codes(huffman_tree, '',
huffman_mapping)
            result = huffman_mapping
        except ValueError:
            result = {"error": "Please enter valid characters
and frequencies."}
    return render_template("index.html", result=result)

if __name__ == "__main__":
    app.run(debug=True)
```

Index.html:-

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
```

NAME:- MARVIN PATEL
ER NO :- 22162101013
SUB :- AAD

PRACTICAL - 10

```
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Huffman Coding</title>
    <link rel="stylesheet" href="{{ url_for('static',
filename='style.css') }}">
</head>
<style>
    /* General Styles */
    body {
        font-family: Arial, sans-serif;
        background: linear-gradient(135deg, #6a11cb, #2575fc);
        display: flex;
        justify-content: center;
        align-items: center;
        height: 100vh;
        margin: 0;
        animation: fadeIn 1s ease-in-out;
    }

    @keyframes fadeIn {
        from {
            opacity: 0;
        }
        to {
            opacity: 1;
        }
    }

    .container {
        width: 350px;
        background-color: #ffffff;
        padding: 20px;
        border-radius: 12px;
```

NAME:- MARVIN PATEL
ER NO :- 22162101013
SUB :- AAD

PRACTICAL - 10

```css
    box-shadow: 0 6px 20px rgba(0, 0, 0, 0.15);
    transform: translateY(-20px);
    animation: slideDown 0.8s ease-out;
}

@keyframes slideDown {
    from {
        transform: translateY(-50px);
        opacity: 0;
    }
    to {
        transform: translateY(0);
        opacity: 1;
    }
}

h1 {
    font-size: 24px;
    margin-bottom: 20px;
    color: #333;
    text-align: center;
    text-shadow: 1px 1px 2px rgba(0, 0, 0, 0.2);
}

label {
    font-weight: bold;
    color: #555;
    display: block;
    margin-bottom: 5px;
}

input[type="text"] {
    width: 100%;
```

NAME:- MARVIN PATEL
ER NO :- 22162101013
SUB :- AAD

PRACTICAL - 10

```css
        padding: 10px;
        margin: 10px 0 20px;
        border-radius: 6px;
        border: 1px solid #ddd;
        transition: all 0.3s ease;
    }

    input[type="text"]:focus {
        border-color: #6a11cb;
        box-shadow: 0 0 8px rgba(106, 17, 203, 0.2);
        outline: none;
    }

    button {
        width: 100%;
        padding: 12px;
        background: linear-gradient(135deg, #6a11cb, #2575fc);
        color: white;
        border: none;
        border-radius: 6px;
        font-size: 16px;
        cursor: pointer;
        transition: background 0.3s ease, transform 0.2s ease;
    }

    button:hover {
        background: linear-gradient(135deg, #2575fc, #6a11cb);
        transform: scale(1.05);
    }

    button:active {
        transform: scale(0.98);
    }
```

NAME:- MARVIN PATEL
ER NO :- 22162101013
SUB :- AAD

PRACTICAL - 10

```css
.error {
    color: red;
    font-weight: bold;
    text-align: center;
    animation: shake 0.4s ease-in-out;
}

@keyframes shake {
    0%, 100% {
        transform: translateX(0);
    }
    25% {
        transform: translateX(-5px);
    }
    50% {
        transform: translateX(5px);
    }
    75% {
        transform: translateX(-5px);
    }
}

h2 {
    color: #333;
    margin-top: 20px;
    text-align: center;
}

ul {
    list-style-type: none;
    padding: 0;
    animation: fadeIn 0.8s ease;
    }

    li {
        font-size: 14px;
        color: #555;
        margin-bottom: 8px;
        padding: 10px;
        background-color: #f9f9f9;
        border-radius: 6px;
        box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
        transition: transform 0.3s ease;
    }

    li:hover {
        transform: scale(1.02);
        background-color: #eef2ff;
    }
</style>

<body>
    <div class="container">
        <h1>Huffman Coding Generator</h1>
        <form method="POST">
            <label>Characters (space-separated):</label>
            <input type="text" name="characters"
placeholder="e.g., a b c d e" required>

            <label>Frequencies (space-separated):</label>
            <input type="text" name="frequencies"
placeholder="e.g., 0.1 0.2 0.3 0.2 0.2" required>

            <button type="submit">Generate Huffman
Codes</button>
```

NAME:- MARVIN PATEL
ER NO :- 22162101013
SUB :- AAD

PRACTICAL - 10

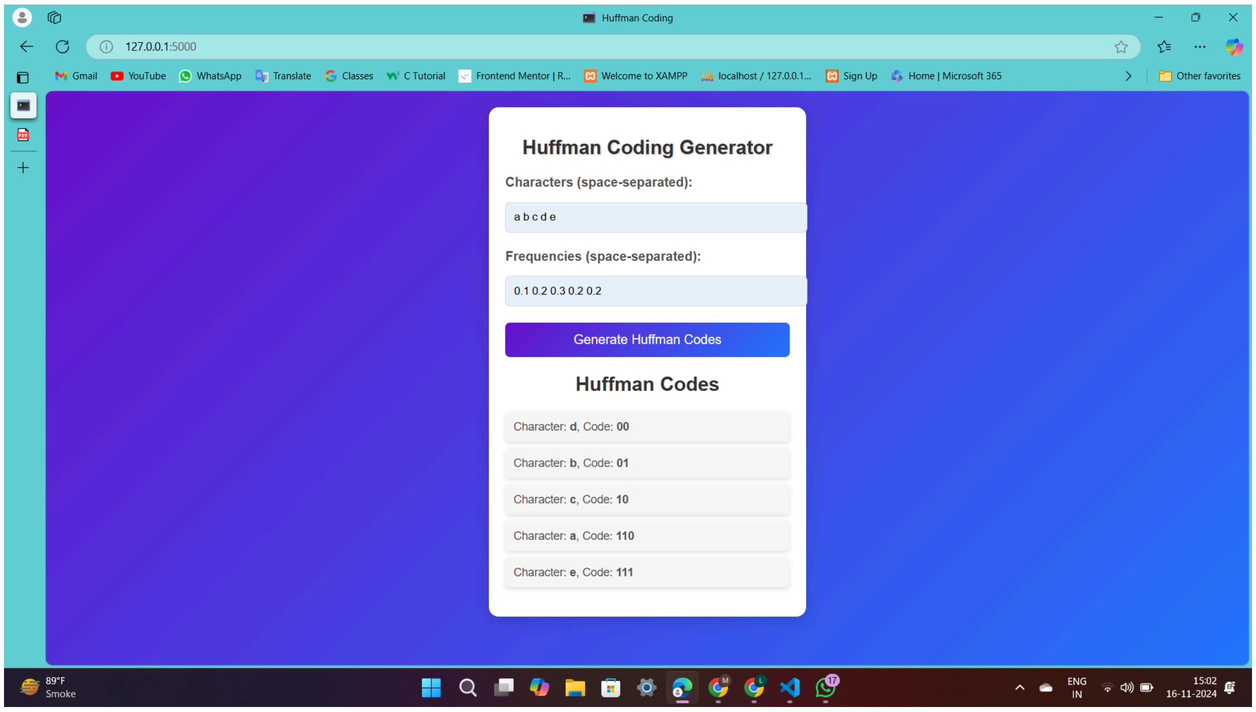
```
        </form>
        {% if result %}
        {% if result.error %}
        <p class="error">{{ result.error }}</p>
        {% else %}
        <h2>Huffman Codes</h2>
        <ul>
            {% for char, code in result.items() %}
            <li>Character: <strong>{{ char }}</strong>, Code:
<strong>{{ code }}</strong></li>
            {% endfor %}
        </ul>
        {% endif %}
        {% endif %}
    </div>
</body>

</html>
```

Output ➖

NAME:- MARVIN PATEL
ER NO :- 22162101013
SUB :- AAD

PRACTICAL - 10



Huffman Coding Generator

Characters (space-separated):

a b c d e

Frequencies (space-separated):

0.1 0.2 0.3 0.2 0.2

Generate Huffman Codes

Huffman Codes

Character: **d**, Code: **00**

Character: **b**, Code: **01**

Character: **c**, Code: **10**

Character: **a**, Code: **110**

Character: **e**, Code: **111**