Program Assgnment 2

spec

PART 1

- Create the Huffman tree by following alphabets and their weight
- execute the code by ./tree (without any parameters)

A = 11	B = 5	C = 2	D = 3	E = 12	F = 3	G = 5
H = 6	I = 8	J = 1	K = 2	L = 4	M = 8	N = 7
O = 2	P = 2	Q = 1	R = 9	S = 6	T = 2	U = 4
V = 1	W = 10	X = 10	Y = 1	Z = 1		

Output format (Part1)

```
DSOO-Program2-Demo: ./tree

A = 11 | B =  5 | C =  2 | D =  3 | E = 12 | F =  3 | G =  5 | H =  6 | I =  8 | J =  1 |
K =  2 | L =  4 | M =  8 | N =  7 | O =  2 | P =  2 | Q =  1 | R =  9 | S =  6 | T =  2 |
U =  4 | V =  1 | W = 10 | X = 10 | Y =  1 | Z =  1 |

A : 1111
B : 11100
C : 101001
D : 00101
E : 000
F : 00100
G : 11101
H : 0011
I : 1000
J : 0110110
K : 101000
L : 10010
M : 0111
N : 0101
O : 101010
P : 011010
Q : 0110111
R : 1011
S : 0100
T : 101011
U : 10011
V : 011000
W : 1100
X : 1101
Y : 0110011
Z : 0110010

WEPL : 550
```

without the parameter

Ten elements in a Row.
Follow the letter sequence.

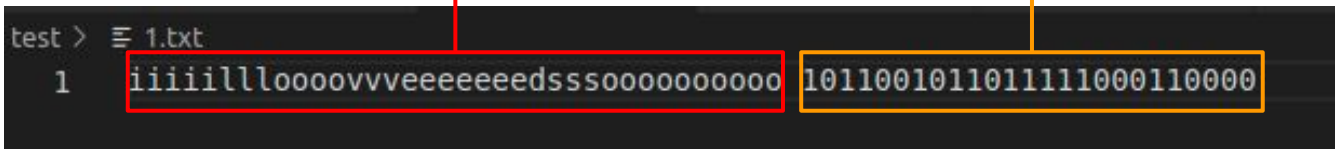
Code list
Follow the letter sequence.

Count the WEPL

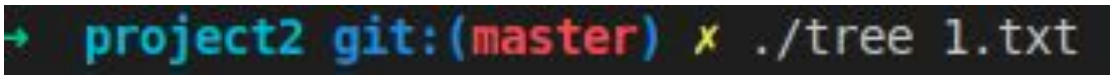
0.txt

PART 2,3

- Create the Huffman tree by the .txt file contexts
- .txt file include two parts
  - characters
  - Huffman code



- You need to read .txt file when you run the code.
  - by the following PATH (e.g., “./test/1.txt”)
- Execute the code with the file path (e.g., ./tree 1.txt)



Output format (Part2,3)

```
test > ≡ 1.txt
  1   iiiiilllooooovvveeeeeeeedsssooooooooooo 10110010110111111000110000
```

read the .txt file

```
DSOO-Program2-Demo: ./tree 1.txt
characters : iiiiilllooooovvveeeeeeeedsssooooooooooo

d =  1 | e =  7 | i =  5 | l =  3 | o = 14 | s =  3 | v =  3 |

d : 1000
e : 111
i : 101
l : 1001
o : 0
s : 1100
v : 1101

Huffman code : 10110010110111111000110000
decode : ilovedsoo
WEPL : 90
```

print code list

Ten elements in a Row.
Follow the letter sequence. (capital letter first)

Print the characters and Huffman code which read in the .txt file

Decode the Huffman code by your Tree
and calculate the WEPL