Applied Algorithms Programming Assignment -4

Text File used ::
http://www.gutenberg.org/ebooks/60630

- Initially i have loaded the text file,after processing the file I have calculated the word count and it's frequency .
- After calculating the frequency , I have sorted them according to the increasing order of the word count occurence.Created tree nodes with characters and their frequencies.
- I have entered the values into the priority queue and built the huffman Tree.
- Picked the left node and right node.created a new node with the frequency of left & right node.(until the priority queue has ">1" elements)
- Then performed walk down of the dictionary to assign Huffman Codes.Binary codes to each leaf node according to their root-to-leaf path.
- Finally Calculated and displayed the number of bits for the final saving.
- To overcome the tie - breaking rule I have used the priority queue.
- I din't handle the case :-> characters that never appeared in the text.

Comparison with the 7 bit encoding & savings ::

	15586	110
e	8755	000
t	6761	1010
a	5871	1000
o	5649	0111
n	4564	0011
i	4422	0010
r	4418	11111
s	4166	11110
h	4047	11100
d	2820	01101
l	2730	01100
.	2285	01000
m	2023	111010
u	1951	101111
c	1789	101101
g	1720	101100
w	1469	100101
f	1388	100100
p	1325	010110
y	1324	010101
b	1089	1110111
I	957	1011101
k	829	1001111

S	750	1001101
R	634	0101110
E	620	0101000
A	597	0100111
v	592	0100110
T	566	0100100
,	433	10111000
H	368	10011100
N	360	10011000
M	347	01011110
W	294	01001011
D	289	01001010
U	241	111011001
P	238	111011000
O	222	101110011
-	185	100110011
Y	179	010111111
:	169	010111110
G	155	010100110
C	151	010100100
K	147	1110110111
F	124	1110110101
j	122	1110110100
L	105	1001110111
1	97	1001110101
_	93	1001110100
x	90	1001100101
[	83	0101001110
]	83	0101001111
B	79	0101001011
0	61	11101101100
J	54	10111001001
q	54	10111001010
3	36	01010010100
2	35	111011011011
z	32	111011011010
!	28	101110010110
4	26	100111011011
*	26	101110010000
5	26	101110010001
(	25	100111011000
)	25	100111011001
6	23	100110010011
?	22	100110010000
"	22	100110010001
9	22	100110010010

```
  /      19        010100101011
  7       18        010100101010
  8       14         1011100101110
  V       13         1001110110101
  '      9        10111001011111
  Q       7         10011101101001
  X       5         10011101101000
  &       2         1011100101111010
  $       2        1011100101111011
  %       1         10111001011110000
  @       1         10111001011110001
  #      1        10111001011110010
  ;      1        10111001011110011
The bits required for calculating the ascii code :: 678727
The bits required if huffman encoding is used is : 456831
----------------------------------------------------
The final bits saving  : 221896
```