

ScreenIO.txt

```
1 *****
2 * PROGRAMMED BY : Wesley Chok
3 * CLASS          : CS1D
4 * SECTION        : MW 2:30 - 4:50p
5 * ASN #15        : Huffman Coding
6 *****
7
8 /*****
9 * OUTPUT (This is a comment, not part of the output)
10 * -----
11 * This outputs the frequency table, huffman code for each letter of the
12 * Gettysberg address, encoded Gettysburg address and the decoded Gettysburg
13 * address. The first output lists all of the characters on the left and
14 * frequencies on the right. The second output outputs the entire encoded
15 * huffman code. Finally the third output outputs the decoded huffman code.
16 * -----
17 /*****
18
19 Character With there Frequencies:
20  00
21 , 100011
22 - 1100111
23 . 11110110
24 B 11110101000
25 F 11110101001
26 G 11110101010
27 I 111101000
28 L 11110101100
29 N 11110101101
30 T 1111010011
31 W 1111010010
32 a 1011
33 b 1010110
34 c 111100
35 d 11010
36 e 011
37 f 110000
38 g 110010
39 h 0101
40 i 11011
41 k 1111010111
42 l 10000
43 m 1010111
44 n 11111
45 o 1001
46 p 1100110
47 q 11110101011
48 r 0100
49 s 10100
50 t 1110
51 u 100010
52 v 101010
53 w 110001
54 y 11110111
55
56 Encoded Huffman Code:1111010100110011000100100001010011110010010100011001011111
57 1110100010100011101010011111110011110111011101010010100001
```

ScreenIO.txt

58 01111001010010010011000100100001100001011111001010110100101
59 00001010110010010011000101100100101111000110000100101001110
60 0101001001111110011100101110111010000111100100111111110110
61 111111011111111101000110010110011111011110001001111110111
62 1101101110011111110001100111100100111111111000111101110101
63 00111101000110111111100111101011001101110101100110100111011
64 1101111000110010111111110100011010011110101101111110010111
65 11001111010001110100100111001010110011001100100100111001101
66 00110100110111110110111001111110011100101101111100010111000
67 0100000010101110111111001011010001100111100010001110111110
68 01111010000111111010101110001010111000011110110001111010110
69 1100111000100110001011001011010001100011111111001010111100
70 10011110100011011111110010110011001001000111011111000111100
71 11011101010110111000000110001101101001000110011100111010011
72 10110111111111001000110001010101111100101011010000111001011
73 0111110001111101111101101110011111110001100100101000010111
74 1111111011100111111011111011011100111111001010010010011110
75 0100111111111000111101110101001111010001011111111101000101
76 0010010011010011110101101111110010111100111101010001100111
77 1001011111110010000100111111110010000111111110101000100100
78 01111110110001111010010011001011010001100101011101111100010
79 01111110010110011001001000111011111000101011010111110111010
80 00001111001111100001101101110000110100010011100000011100101
81 10111110001100011011010011110110001111010010011000101101110
82 10100110011110010011010111011001110100100110100111101011011
83 11110010111110011001011001100110100101001110110111001111110
84 01001110000001110010110111110001100001101101110000110101000
85 1100101110100001011001100001101111111011100000001000111010
86 01110110111111111001000110011010000101111110001100110000100
87 10100001110010110011010001100110001010110010001010110100011
88 00110010101110101001100111001010111101101000010000110111010
89 10011101000011100101101111100011100101101111100011111101111
90 10110111001111110010101111101111001001011110001000011011101
91 01001111110110001111010001110001101110100001011100001110100
92 1110010011111001010110100001100001101111101110110111111110
93 01000101111111110100011001100100100111001100110100001110010
94 11011111000110001011001010001011001100010100001101000110101
95 00100111001011101110100111101100011110101000100010111010001
96 10011011111110010110010000101101001100100110100001010001111
97 11110100011100011001100010110011110010111111111111100111100
98 011010011110101101111110010111110011001111100111100111000
99 101100111100101111111111111100111100011110010011111110100011
100 11110001001011111001100110011111001110011000101100111100101
101 111111111111001111000010110111000010000100111100010011001111
102 10011100111001011101110100001100100100100110001011111110101
103 11101100011110100110101011001010110010010111010100110010101
104 1101111111100011001000011011101010110111111110010001011111
105 11110100011010011101111010100011001100010101100100101001110
106 01001000101100101100101000001111010000101011010001110001100
107 01011011101010011001111001001111111010001111110001001011111
108 00111101000110111110100011001100001011010000101110101101001
109 10101001100100110001001000011001101001100101000011001101001
110 110001011010000111010010010111101011010001001010000110010011
111 11100100101111110011101111011000111101001101010110011000110
112 010100100001110100011000111011100001000000100001101111101110
113 10000011001111110011110011100011001111110010100001000010011
114 11111100100001000111010111011101011110101100110100001100010

ScreenIO.txt

115 10110111110001100010110010100101111110111000101011010001110
116 0011001010110100010111000110111110001111001011111100111110
117 11101010011010000110000100101001100100111110001100010101101
118 1111000111001010111110111001101011011110100001010110100011
119 11110110001111010001110001101110100001100001001010000100010
120 1010000111001010110010000110111010101101111111100101000110
121 00100101111100101011010010001100111010010010101100110011010
122 01111010110111111001011111001111010000101011010001100111010
123 0100111001010110010001011111110000110111111110111010001010
124 11110100011000110010100111101011100110001010111011111100010
125 10011100101011111101110011000101011001001100001001100010110
126 01001011110000101011010001100010110111010100110011100101100
127 01010100001100001011010000101001001001111110011010110100001
128 1110111001011110101010101011111111110001111010111101100011
129 11010001110001101110100000100101111100101011010000110000100
130 10100001000101010000111010010010101100110001010110100011001
131 10100111101011011111100101111100111101000111010010011100101
132 01100110010010001110111110001110101110100111101011100010001
133 1101011110111101111111101111111100100010101100111110000100
134 10100011001000101010000110011111001110011100101101111100011
135 00000100100110101110011100111011101000110001011001111111001
136 0100011110100011010011101111010001100010110011101111111010
137 11101100110111111111110001000111011101000111101000110100111
138 01010100111101101110011111100111010010011100101101111100011
139 11001011100010101000110011000010010100001100010101110111111
140 00010100111001010111111011100110010101110101001100111001010
141 11001000010111010011100011000010001010000100000010101110111
142 011101001000100100011001001110000001101001110101001111011
143 01110011111100110011111001110011100101101111100011000101100
144 01010110100011000101110111100100101100001111011100010001110
145 10010011000010101001100111001011011111000111001010111010001
146 10011010011101111010001010001011011100001000000111111001111
147 000010110111010100110011010110111110100011011111110010101
148 0101111011111100110011111001110011100101101111100011100101
149 11011101000011111101111101101110011111110001100100010111111
150 10100110100001111010101010011101010001100101000101101110000
151 10000000101101110101001100101100111110111100010010101101101
152 10100111001010010011100000011000001000110111101010011010111
153 0011001111100111001011111111010001110010110111110001100101
154 0011010100110100111111010111011111111111100010011100000011100
155 10101100110011001110011100110100000111000110010101101111011
156 100111001010110011001110011100111001101000001110001100110000
157 100101000011100101011001100111001110011100110100000111000110
158 01010001011011100001000000111111001111000110011001101001101
159 11010001010011000001001001101011100111001010110001110110100
160 1110010111110110
161
162

163 Decoded Huffman Code: Four score and seven years ago our fathers brought forth on
164 this continent, a new nation, conceived in Liberty, and
165 dedicated to the proposition that all men are created
166 equal.

167
168 Now we are engaged in a great civil war, testing whether
169 that nation, or any nation so conceived and so dedicated,
170 can long endure. We are met on a great battle-field of that
171 war. We have come to dedicate a portion of that field, as

ScreenIO.txt

172 a final resting place for those who here gave their lives
173 that that nation might live. It is altogether fitting and
174 proper that we should do this.
175
176 But, in a larger sense, we cannot dedicate -- we cannot
177 consecrate -- we cannot hallow -- this ground. The brave
178 men, living and dead, who struggled here, have consecrated
179 it, far above our poor power to add or detract. The world
180 will little note, nor long remember what we say here, but
181 it can never forget what they did here. It is for us the
182 living, rather, to be dedicated here to the unfinished work
183 which they who fought here have thus far so nobly advanced.
184 It is rather for us to be here dedicated to the great task
185 remaining before us -- that from these honored dead we take
186 increased devotion to that cause for which they gave the
187 last full measure of devotion -- that we here highly
188 resolve that these dead shall not have died in vain --
189 that this nation, under God, shall have a new birth of
190 freedom -- and that government of the people, by the
191 people, for the people, shall not perish from the earth.

AS 15 - Huffman Coding.cpp

```

1 #include <bits/stdc++.h>
2 #define MAX_TREE_HT 256
3 using namespace std;
4
5 /*****
6  * ASN #15 - Huffman Coding
7  * -----
8  * This program will output the class heading
9  * -----
10 *
11 *****/
12
13 map<char, string> codes;    // holds the huffman value
14
15 map<char, int> freq;        // holds the frequency of the character via input
16                             // data
17
18
19 struct MinHeapNode
20 {
21     char data;              // holds the input character
22     int freq;               // holds the frequency of the character
23     MinHeapNode *left;     // left child
24     MinHeapNode *right;    // right child
25
26     /*****
27     * Constructor MinHeapNode
28     * -----
29     * Instantiates the variables given in the struct.
30     * -----
31     * *****/
32     MinHeapNode(char data, int freq)
33     {
34         left = right = NULL;
35         this->data = data;
36         this->freq = freq;
37     }
38 };
39
40
41 struct compare
42 {
43     /*****
44     * FUNCTION operator
45     * -----
46     * Function for the priority queue, used for utility purposes only.
47     * -----
48     * *****/
49     bool operator()(MinHeapNode* l, MinHeapNode* r)
50     {
51         return (l->freq > r->freq);
52     }
53 };
54
55 /*****
56 * FUNCTION printCodes
57 * -----

```

```

58 * This function prints out the characters and their respective huffman
59 * values, used for utility purposes only.
60 *
61 * *****/
62 void printCodes(struct MinHeapNode* root, string str)
63 {
64     if (!root)
65     {
66         return;
67     }
68     if (root->data != '$')
69     {
70         cout << root->data << ": " << str << "\n";
71     }
72     printCodes(root->left, str + "0");
73     printCodes(root->right, str + "1");
74 }
75
76
77
78 /*****
79 * FUNCTION storeCodes
80 *
81 * This function stores the characters and their respective huffman values
82 * into a hash table.
83 *
84 * *****/
85 void storeCodes(struct MinHeapNode* root, string str)
86 {
87     if (root==NULL)
88     {
89         return;
90     }
91     if (root->data != '$')
92     {
93         codes[root->data]=str;
94     }
95     storeCodes(root->left, str + "0");
96     storeCodes(root->right, str + "1");
97 }
98
99
100 priority_queue<MinHeapNode*, vector<MinHeapNode*>, compare> minHeap;
101 // stores the heap tree
102
103 /*****
104 * FUNCTION HuffmanCodes
105 *
106 * This function will construct the huffman tree and store it in minHeap.
107 * The for loop will push the heap values until it reaches the end of the
108 * frequency map.
109 *
110 * *****/
111 void HuffmanCodes(int size)
112 {
113     struct MinHeapNode *left, *right, *top;
114     for (map<char, int>::iterator v = freq.begin(); v != freq.end(); v++)

```

```

115     {
116         minHeap.push(new MinHeapNode(v->first, v->second));
117     }
118     while (minHeap.size() != 1)
119     {
120         left = minHeap.top();
121         minHeap.pop();
122         right = minHeap.top();
123         minHeap.pop();
124         top = new MinHeapNode('$', left->freq + right->freq);
125         top->left = left;
126         top->right = right;
127         minHeap.push(top);
128     }
129     storeCodes(minHeap.top(), "");
130 }
131
132 /*****
133 * FUNCTION calcFreq
134 *
135 * This function will store the map of each character with its respective
136 * frequency in the input string located at int main()
137 *
138 * *****/
139 void calcFreq(string str, int n)
140 {
141     for (int i=0; i<str.size(); i++)
142     {
143         freq[str[i]]++;
144     }
145 }
146
147 /*****
148 * FUNCTION decode_file
149 *
150 * This function will iterate the string value. For instance, if string_value
151 * equals 1 then it will move the node to the right, if string_value equals
152 * 0 then it will move the node to the left.
153 *
154 * *****/
155 string decode_file(struct MinHeapNode* root, string string_value)
156 {
157     string ans = "";
158     struct MinHeapNode* curr = root;
159     for (int i = 0; i < string_value.size(); i++)
160     {
161         if (string_value[i] == '0')
162         {
163             curr = curr->left;
164         }
165         else
166         {
167             curr = curr->right;
168         }
169     }
170
171     if (curr->left==NULL and curr->right==NULL)

```

AS 15 - Huffman Coding.cpp

```

172     {
173         ans += curr->data;
174         curr = root;
175     }
176 }
177
178 return ans+'\0';
179 }
180
181
182 int main()
183 {
184
185     /*****
186     * CONSTANT
187     * -----
188     * OUTPUT - USED FOR CLASS HEADING
189     * -----
190     * PROGRAMMER : Wesley Chok
191     * CLASS      : CS 1D
192     * SECTION    : MW 2:30p - 4:50p
193     * ASN_NUM    : 15
194     * ASN_NAME   : Huffman Coding
195     *****/
196     string str = "Four score and seven years ago our fathers brought forth on "
197                 "this continent, a new nation, conceived in Liberty, and dedicated "
198                 "to the proposition that all men are created equal. "
199                 ""
200                 "Now we are engaged in a great civil war, testing whether that "
201                 "nation, or any nation so conceived and so dedicated, can long "
202                 "endure. We are met on a great battle-field of that war. We have "
203                 "come to dedicate a portion of that field, as a final resting place "
204                 "for those who here gave their lives that that nation might live. "
205                 "It is altogether fitting and proper that we should do this. "
206                 ""
207                 "But, in a larger sense, we cannot dedicate -- we cannot "
208                 "consecrate -- we cannot hallow -- this ground. The brave men, "
209                 "living and dead, who struggled here, have consecrated it, far "
210                 "above our poor power to add or detract. The world will little note, "
211                 "nor long remember what we say here, but it can never forget what "
212                 "they did here. It is for us the living, rather, to be dedicated "
213                 "here to the unfinished work which they who fought here have thus "
214                 "far so nobly advanced. It is rather for us to be here dedicated to "
215                 "the great task remaining before us -- that from these honored dead "
216                 "we take increased devotion to that cause for which they gave the "
217                 "last full measure of devotion -- that we here highly resolve that "
218                 "these dead shall not have died in vain -- that this nation, under "
219                 "God, shall have a new birth of freedom -- and that government of "
220                 "the people, by the people, for the people, shall not perish from "
221                 "the earth.";
222
223     string encodedString; // holds data for the encoded string
224     string decodedString; // holds data for the decoded string
225
226     calcFreq(str, str.length());
227     HuffmanCodes(str.length());
228

```



```
229     cout << "Character With there Frequencies:";
230
231     cout << endl;
232
233     for (auto v=codes.begin(); v!=codes.end(); v++)
234     {
235         cout << v->first << ' ' << v->second << endl;
236     }
237
238     for (auto i: str)
239     {
240         encodedString+=codes[i];
241     }
242
243     cout << endl;
244
245     cout << "Encoded Huffman Code:" << encodedString << endl << endl;
246
247     decodedString = decode_file(minHeap.top(), encodedString);
248
249     cout << endl;
250
251     cout << "Decoded Huffman Code:" << decodedString << endl << endl;
252
253
254     return 0;
255 }
256
```