DATA STRUCTURES REPORT

Contributed by:

M. Hamza Shahzad 20i-0796

Momina Hayat 20i-1840

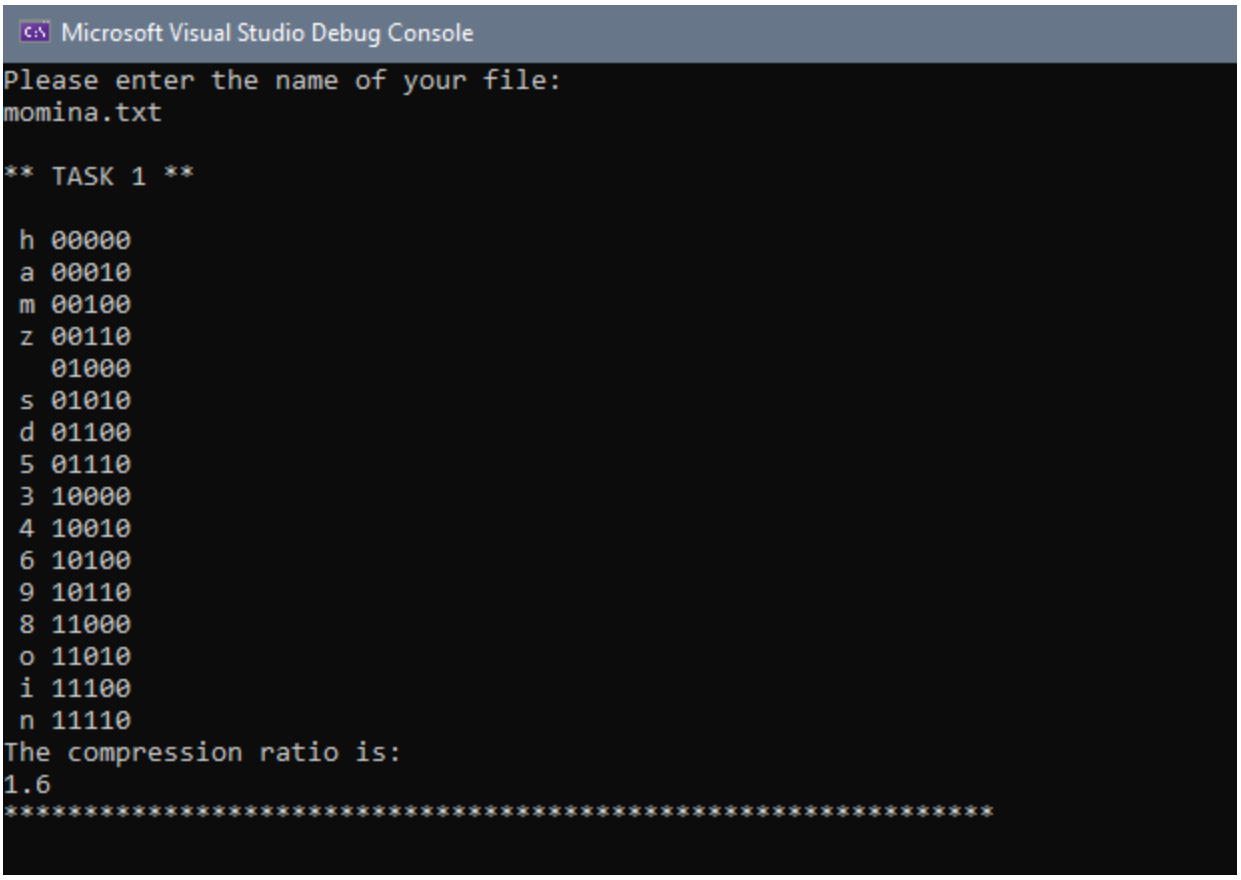
File handling:

Firstly, we did file handling to read a text file. For this we made two arrays one for storing the characters and the other one to store the frequencies of the the characters. Both the arrays work simultaneously. And the arrays are also dynamically allocated. So that if a character repeats it is again added to the previously existing character.

TASK 1:

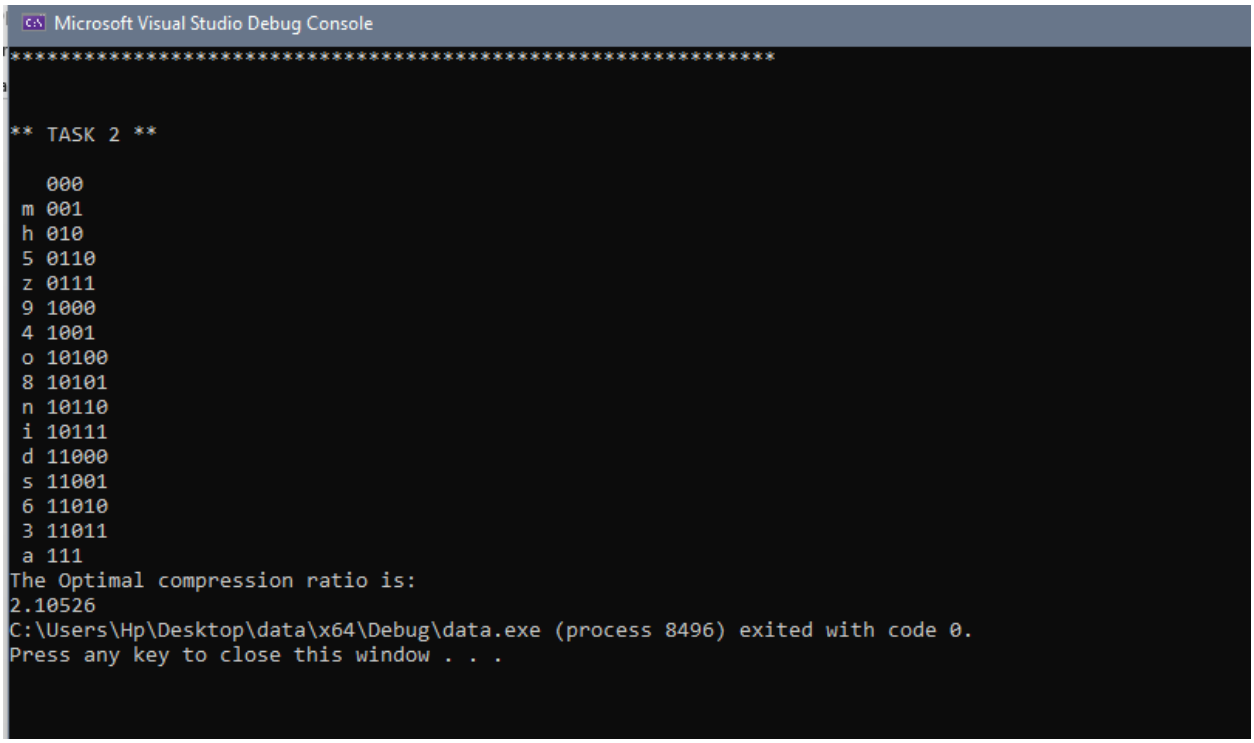
In task 1 we implemented a queue. Made all the member functions of enqueue and dequeue. Another major thing we implemented is the Huffman algorithm. In this algorithm we took first two nodes added them and their sum is stored onto a new node and the smaller node goes onto the left side and the larger one goes onto the right side. And using this algorithm we implemented the tree. Using this we found the compressed bits. If the tree traverses from the left “0” occurs and if the tree traverses from the right “1” occurs. And from the compressed bits we found the compressed ratio. Which is

total number of bits / compressed bits + frequency of each bit



TASK 2:

In task 2 we implemented the same thing as task 1. Here we implemented a priority queue who's enqueue function is different from a normal queue. In that function we sorted it in ascending order and then enqueued all the values. And after that we implemented the Huffman algorithm. Because of using the priority queue we got the optimized compression ratio. The formula for the compression is same as explained earlier.


Microsoft Visual Studio Debug Console
****
** TASK 2 **
000
m 001
h 010
5 0110
z 0111
9 1000
4 1001
o 10100
8 10101
n 10110
i 10111
d 11000
s 11001
6 11010
3 11011
a 111
The Optimal compression ratio is:
2.10526
C:\Users\Hp\Desktop\data\x64\Debug\data.exe (process 8496) exited with code 0.
Press any key to close this window . . .