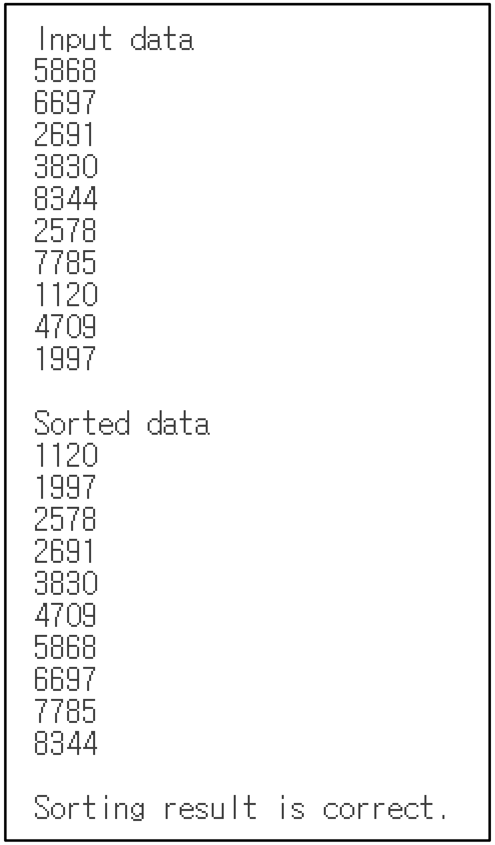
1. (Programming: **30 points**)
   Implement the heap sort by using ‘build_max_heap’ function as below.
   - Refer to the attach file (providing the skeleton code).
   - Implement `‘build_max_heap’`.
   - Generate an input array of size 10, 100, 1000, and 5000, and then apply the heap sort and measure the runtime.



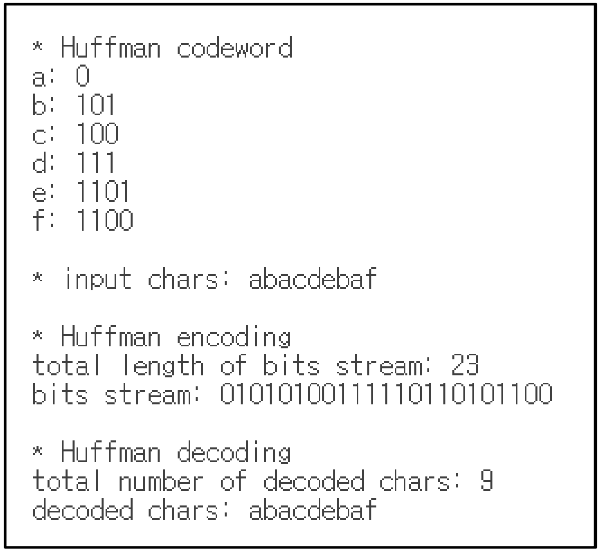
```
Input data
5868
6697
2691
3830
8344
2578
7785
1120
4709
1997

Sorted data
1120
1997
2578
2691
3830
4709
5868
6697
7785
8344

Sorting result is correct.
```

Execution result

2. (Programming: **70 points**)
Implement the code for encoding and decoding an input data using Huffman binary tree.
- Refer to the attach file (providing the skeleton code).
- The following functions should be implemented.
`‘huffman_traversal’, ‘huffman_encoding’, ‘huffman_decoding’`
- It is not mandatory to follow the skeleton code. Feel free to implement your own codes, if needed.



```
* Huffman codeword
a: 0
b: 101
c: 100
d: 111
e: 1101
f: 1100

* input chars: abacdebaf

* Huffman encoding
total length of bits stream: 23
bits stream: 01010100111110110101100

* Huffman decoding
total number of decoded chars: 9
decoded chars: abacdebaf
```

Execution result