Return of the (Coding) Trees

Step 1

As we hinted at in a past discussion about Alan Turing and his work cracking Enigma, there is great use in being able to encode messages. Whether the goal is to encrypt secret messages to prevent unwanted eyes from viewing the content, or to compress files on a computer to reduce its filesize, it is intuitive that there are two key steps to encryption:

- **Original Message → Coded Message** (**Encryption**)
- **Coded Message → Original Message** (**Decryption**)

To encode a message, it is clear that we need some way of mapping words/symbols from the original message to their coded counterparts (and vice-versa). This brings about two vital questions:

- How do we **generate** this mapping?
- How do we **represent** this mapping?

Step 2

CODE CHALLENGE: Encoding a Message as ASCII Values

We will be dealing with a trivial example of message-encoding, where the mapping of original symbols to the "code" is simply taking the ASCII value of the symbol. For example, the "code" version of the letter 'A' would be 65.

You will be given as input a string of any length consisting of any possible ASCII characters. You must print out the numerical ASCII value of each character of the string, separated by spaces.

Sample Input:

```
Hello, World!
```

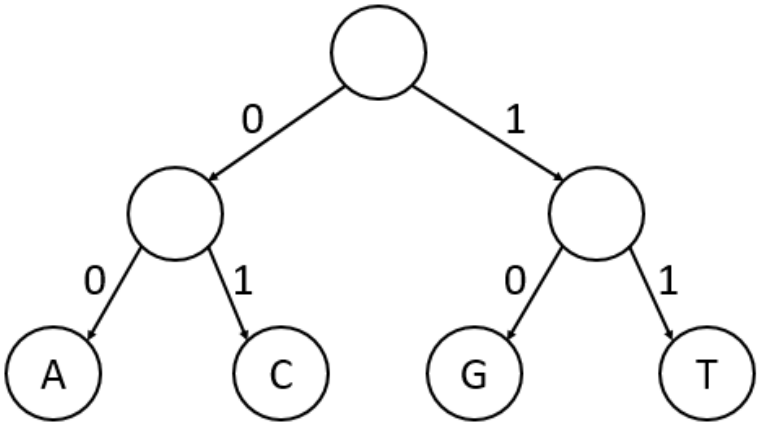
Sample Output:

```
72 101 108 108 111 44 32 87 111 114 108 100 33
```

To solve this problem please visit https://stepik.org/lesson/26125/step/2

Step 3

We can represent the trivial example code mapping as well as any other character-based code mapping using a tree, called a **coding tree**. In this tree, edges are labeled by "code" symbols, and the "non-coded" symbols label the leaves. A non-coded symbol (i.e., a leaf in the coding tree) is encoded by the "code" symbols labeling the edges in the path from the root to the non-coded symbol. For example, take the following simple coding tree:



In this tree, **A** is encoded by **00**, **C** is encoded by **01**, **G** is encoded by **10**, and **T** is encoded by **11**.

Step 4

CODE CHALLENGE: Encoding a Message Using a Coding Scheme

We will be using the same trivial example as the previous step, where we have the following symbol mappings:

- **A ↔ 00**
- **C ↔ 01**
- **G ↔ 11**
- **T ↔ 11**

Given a string of A's, C's, G's, and T's, encode the message via the above mapping and print out the resulting encoded message to standard output.

Sample Input:

```
ACGT
```

Sample Output:

```
00011011
```

To solve this problem please visit https://stepik.org/lesson/26125/step/4

Step 5

CODE CHALLENGE: Decoding a Message Using a Coding Scheme

We will be using the same trivial example as the previous step, where we have the following symbol mappings:

- **A ↔ 00**
- **C ↔ 01**
- **G ↔ 11**
- **T ↔ 11**

Given a string of 0's and 1's, decode the message via the above mapping and print out the resulting decoded message to standard output.

Sample Input:

```
00011011
```

Sample Output:

```
ACGT
```

To solve this problem please visit https://stepik.org/lesson/26125/step/5

Step 6

In this section, we simply wanted you to focus on how to encode or decode a message when given a mapping, and we wanted to demonstrate the fact that such a mapping could be represented using a coding tree. In the next section, we will cover the requirements and restrictions of such a mapping as well as how to go about creating one.