1. Пишемо умову завдання: закодувати “Don't worry if it doesn't work right. If everything did, you'd be out of a job” (Не хвилюйся якщо щось не працює, якщо би все працювало ти був би звільнений з роботи). Можете писати транслитом, якшо не получиться кирилицею «Ne khvyliuisia yakshcho shchos ne pratsiuie, yakshcho by vse pratsiuvalo ty buv by zvilnenyi z roboty». Я на наступній парі розкажу про використання кирилиці.
2. Здійснюємо підрахунок кількості символів. Для цього пропоную використати сайт <http://www.siteprojects.ru/?article=seo-kolichestvo>:

**Частота Символ**

1 N

6 a

4 b

3 c

5 e

7 h

8 i

3 k

3 l

3 n

6 o

2 p

3 r

8 s

4 t

4 u

5 v

8 y

2 z

3. Наводимо исходный код з коментарями

…

# Функція побудови дерева. На вході ліва і права гілки дерева

def make\_code\_tree(left, right):

# Повертаємо ліві, праві гілки паралельно добавивши їх в список

return (left, right, append(symbols(left), symbols(right)),

# Додаємо значення для лівої і правої гілки

weight(left)+weight(right))

…

1. Наводимо або бітовий рядок (це буде простіше як на мене, оскільки Вам необхідно буде просто скористатися командою print) або дерево рішень малюємо. Отримуємо щось типу того:

|  |  |  |
| --- | --- | --- |
| **Символ** | **Частота** | **Код** |
| ' ' | 15 | 101 |
| ',' | 1 | 010000 |
| 'N' | 1 | 010001 |
| 'a' | 6 | 0111 |
| 'b' | 4 | 11001 |
| 'c' | 3 | 01011 |
| 'e' | 5 | 0011 |
| 'h' | 7 | 1001 |
| 'i' | 8 | 1110 |
| 'k' | 3 | 01001 |
| 'l' | 3 | 01100 |
| 'n' | 3 | 01101 |
| 'o' | 6 | 1000 |
| 'p' | 2 | 110001 |
| 'r' | 3 | 01010 |
| 's' | 8 | 1111 |
| 't' | 4 | 11010 |
| 'u' | 4 | 11011 |
| 'v' | 5 | 0010 |
| 'y' | 8 | 000 |
| 'z' | 2 | 110000 |

1. Виводимо наше закодоване повідомлення. Для нашого повідомлення на транслите буде:

010001 0011 101 01001 1001 0010 000 01100 1110 11011 1110 1111 1110 0111 101 000 0111 01001 1111 1001 01011 1001 1000 101 1111 1001 01011 1001 1000 1111 101 01101 0011 101 110001 01010 0111 11010 1111 1110 11011 1110 0011 010000 101 000 0111 01001 1111 1001 01011 1001 1000 101 11001 000 101 0010 1111 0011 101 110001 01010 0111 11010 1111 1110 11011 0010 0111 01100 1000 101 11010 000 101 11001 11011 0010 101 11001 000 101 110000 0010 1110 01100 01101 0011 01101 000 1110 101 110000 101 01010 1000 11001 1000 11010 000

1. Висновок