CASE STUDY

8

Compressing String and Files

Before sending a message, storing a string in a file, it is always a good idea to compress it. Compressing means making the string smaller in terms of the number of bytes of data it contains. To be useful, compression should be lossless. That is, the compressed file when uncompressed is exactly the same as the original file. When you create a zip folder, it uses lossless compression. While all zip programs are lossless, converting uncompressed audio files to a compressed audio format such as WMA (Windows Media Audio), on the other hand, uses lossy compression. This is because data representing sound that is beyond the range of human hearing is removed from the file during the conversion process. So, the played audio with removed data will not be noticed by a listener.

Advantages of Compression

- Compressing files allows you to store more files or data in the available storage space. Lossless compression can even reduce a file to 50 percent of its original size.
- Compressed files contain fewer bits of data than uncompressed files. Therefore, they can be downloaded and transferred at a faster speed.
- Compressed files also save money. First, they use less space on hard disk. So, you need not buy additional storage space (hard disk) to save big files. They can be compressed and make efficient utilization of space on the hard disk. Second, we know that ISPs charge money used on amount of data downloaded. Compressed files means downloading fewer bits for the same file. Hence, this results in reduction in costs.

```
#1 Program to compress strings by the number of characters within the string

def compress(msg):
    msg_list = list(msg)
    comp_str = []
    prev = msg_list[0]
    count = 1
    for i in range(1,len(msg_list)):
        if msg_list[i] == prev:
            count += 1
        else:
            comp_str.append(prev)
            comp_str.append(str(count))
            prev = msg_list[i]
            count = 1
```

```
    # add the last character
    comp_str.append(prev)
    comp_str.append(str(count))
    return ''.join(comp_str)
mes = input("Enter the message:")
print("The compressed message is:", compress(mes))
```

OUTPUT

```
Enter the message : abaabbcccddddd
The compressed message is :  a1b1a2b2c3d5
```

#2 Program to compress and decompress text stored in a file

```
import re
from ast import literal_eval
import os
def compress():
    try:
        fileName = input('Enter the name of the file to be compressed:')
        file = open(fileName)
        text = file.read()
        file.close()
        p = re.compile(r'[\w]+|[\W]')
        split = p.findall(text)
        b = []
        wordList = []
        for word in split:
            try:
                r = wordList.index(word) + 1
            except ValueError:
                wordList.append(word)
                r = len(wordList)
            b.append(r)
        file = open('compressed.txt', 'w')
        file.write(str(wordList)+'\n'+str(b))
        file.close()
    except:
        print('File does not exist')

def decompress():
    try:
        fileName = input('Enter the name of the file to be decompressed:')
        file = open(fileName)
    except:
        print('File does not exist')
    print("Contents of the compressed file is:")
    words = literal_eval(file.readline().rstrip('\n'))
    pos = literal_eval(file.readline())
    temp = []
```

```
    for index in pos:
        temp.append(words[index-1])
    sentence = ''.join(temp)
    print(sentence)
compress()
decompress()
```

OUTPUT

```
Enter the name of the file to be compressed : File.txt
Enter the name of the file to be decompressed : compressed.txt
Contents of the compressed file is :
Greetings to All !!!
Welcome to the world of programming
```