Data Compression Using Run-Length Encoding

Run-length encoding (**RLE**) is a very simple form of data compression in which consecutive sequences of the same data value *(runs)* are stored or transmitted as a single data value and count, rather than as the original individual data elements. This is particularly useful for data that contains many such runs such as simple graphic images and faxed documents. If the original doesn't have many runs, it could increase rather than decrease the size of the data file or transmission.

A fax machine scans a page and represents that image as black or white pixels, which are sent over telephone lines to another fax machine that will print the pixels onto a blank page. The total number of pixels to be transmitted may be very large which would result in lengthy transmission times. Because fax images often have large blocks of white (*e.g.* margins and inter-line spacing) or black (*e.g.* horizontal lines) they are readily amenable to run-length encoding.

In this exercise, you will decode a signal and reconstruct the transmitted image.

The signal will be represented as a sequence of bits (0 or 1) and markers (P). A sequence of bits represents a number. A single marker ("P") indicates the end of one number and the beginning of the next. A sequence of two consecutive markers ("P P") indicates the end of a row.

Once you have decoded the signal, you will decompress the data to reconstruct the image. The first number of each row indicates the number of consecutive black pixels that begin in the first column of that row. The next number indicates the number of consecutive white pixels that immediately follow. The remaining numbers continue to specify the length of black and white pixel runs. The remaining pixels on a row after the last number are filled with the next alternating value (black or white).

You will also have the opportunity to encode an image of your own creation. The process is simply the reverse of the previous one. Create an image consisting of black and white pixels. For each row of the image, determine the numeric values that provide the run-length encoding. Then convert that encoding into a signal by representing each number as its binary equivalent and inserting markers as appropriate. Exchange your signal with that of a partner. Once you have both decoded and reconstructed the images, compare your results with the originals.

Convert each transmission into corresponding rows of number sequences

```
P P P P P P P P P P 1 0 0 P 1 0 1 1 P P 1 0 0 P 1 0
0 1 P 1 0 P 1 P P 1 0 0 P 1 0 0 1 P 1 0 P 1 P P 1 0
0 P 1 0 1 1 P P 1 0 0 P 1 0 0 1 P P 1 0 0 P 1 0 0 1
P P 1 0 1 P 1 1 1 P P P 1 0 0 0 1 P P 1 P 1 1 1 1 P
```

```
1 1 0 P 1 0 1 P 1 0 P 1 1 P P 1 0 0 P 1 0 P 1 0 1 P
1 0 P 1 1 P 1 P P 1 1 P 1 P 1 0 0 1 P 1 P 1 0 P 1 P
P 1 1 P 1 P 1 0 0 1 P 1 P 1 P 1 P P 1 0 P 1 P 1 0 1
1 P 1 P P 1 0 P 1 P 1 0 1 0 P 1 0 P P 1 0 P 1 P 1 0
0 1 P 1 P 1 P 1 P P 1 0 P 1 P 1 0 0 0 P 1 P 1 0 P 1
P P 1 0 P 1 P 1 1 1 P 1 P 1 1 P 1 P P 1 P 1 P 1 P 1
P 1 0 0 P 1 0 P 1 1 P 1 P P P 1 P 1 0 P 1 P 1 0 P 1
0 P 1 0 1 P 1 P P P 1 P 1 1 P 1 0 P 1 0 1 P 1 0 P P
1 P 1 1 P 1 0 P 1 0 1 P
```