

Aayush Shah
19BCE245
27 August 2021

Practical 3

Frame Generation

• Write a program to read a stream of data from data file to create frames by implementing bit stuffing and character stuffing.

• Code :

```
#!/usr/bin/env python3

def byte_stuff_msg(message, flag_seq, esc_seq):
    return message.replace(esc_seq,
esc_seq+esc_seq).replace(flag_seq, esc_seq+flag_seq)

def byte_unstuff_msg(message, flag_seq, esc_seq):
    return
message.replace(esc_seq+flag_seq, flag_seq).replace(esc_seq+esc
_seq, esc_seq)

def bit_stuff_msg(message, flag_seq):
    flag_seq = '011111'
    return message.replace(flag_seq, '0111110')

def bit_unstuff_msg(message, flag_seq):
    flag_seq = '011111'
    return message.replace(flag_seq+'0', flag_seq)

message =
'01111110asdfsdf011111100111111001111110abcbcasdfasdfsdfabc'
flag_seq = '01111110'
esc_seq = 'abc'
print('Original      : ', message)
```

```

#bin_data = ' '.join(format(ord(x), 'b') for x in message)
#decoded = "".join([chr(int(binary, 2)) for binary in
bin_data.split(" ")])

bit_stuffed_msg = bit_stuff_msg(message, flag_seq)
print('BIT stuffing      : ',bit_stuffed_msg)

bit_unstuffed_msg = bit_unstuff_msg(bit_stuffed_msg, flag_seq)
print('BIT unstuffing    : ',bit_unstuffed_msg)

byte_stuffed_msg = byte_stuff_msg(message, flag_seq,esc_seq)
print('BYTE stuffing     : ',byte_stuffed_msg)

byte_unstuffed_msg = byte_unstuff_msg(byte_stuffed_msg,
flag_seq,esc_seq)
print('BYTE unstuffing   : ',byte_unstuffed_msg)

print('Are bit unstuffed, byte unstuffed and original strings
same : ',bit_unstuffed_msg==byte_unstuffed_msg==message)

```

- **Execution :**

```

stuff_unstuff.py
Python
Language Run Stop Run Settings... Back/Forward View
stuff_unstuff.py
52 message = '01111110asdfsdf011111100111111001111110111110abcbacasdfasdfsdfabc'
53 flag_seq = '01111110'
54 esc_seq = 'abc'
55 print('Original      : ',message)
56
57 #bin_data = ' '.join(format(ord(x), 'b') for x in message)
58 #decoded = "".join([chr(int(binary, 2)) for binary in bin_data.split(" ")])
59
60 bit_stuffed_msg = bit_stuff_msg(message, flag_seq)
61 print('BIT stuffing   : ',bit_stuffed_msg)
62
63 bit_unstuffed_msg = bit_unstuff_msg(bit_stuffed_msg, flag_seq)
64 print('BIT unstuffing : ',bit_unstuffed_msg)
65
66 byte_stuffed_msg = byte_stuff_msg(message, flag_seq,esc_seq)
67 print('BYTE stuffing  : ',byte_stuffed_msg)
68
69 byte_unstuffed_msg = byte_unstuff_msg(byte_stuffed_msg, flag_seq,esc_seq)
70 print('BYTE unstuffing : ',byte_unstuffed_msg)

Original      : 01111110asdfsdf011111100111111001111110111110abcbacasdfasdfsdfabc
BIT stuffing   : 01111110asdfsdf011111101001111110100111111010abcbacasdfasdfsdfabc
BIT unstuffing : 01111110asdfsdf011111100111111001111110111110abcbacasdfasdfsdfabc
BYTE stuffing  : abc01111110asdfsdfabc01111110abc01111110abc01111110abcbacabcbacasdfasdfsdfabcabc
BYTE unstuffing : 01111110asdfsdf011111100111111001111110111110abcbacasdfasdfsdfabc
Are bit unstuffed, byte unstuffed and original strings same : True

Run Succeeded Time 61 ms Peak Memory 7.3M bit_unstuff_msg Tabs: 4 Line 113, Column 52

```