

Practical 9

Aim: Write a program to implement LZ78 compression algorithm

Program:

```
a=input('Enter The Sequence To Encode ')
l=[]
s=""
m=[]
b=0
for i in a:
    s=s+i
    if s not in l:
        l.append(s)
        m.append([b,s[-1]])
        b=0
        s=""
    else:
        b=(l.index(s)+1)
print('Encoded Sequence Is ',m,sep='\n')
```

Output:

```
PS C:\Users\JOY PATEL\Desktop> python .\P9.py
Enter The Sequence To Encode abracadabra
Encoded Sequence Is
[[0, 'a'], [0, 'b'], [0, 'r'], [1, 'c'], [1, 'd'], [1, 'b'], [3, 'a']]
PS C:\Users\JOY PATEL\Desktop> █
```

Conclusion: LZ78 is an algorithm in which all the letters are passed through the algorithm and the output is a doublet specifying the index value and character encoded.

Practical 10

Aim: Write a program to implement LZ78 decompression algorithm

Program:

```
a=int(input('Enter Total Doublets '))
```

```
l=[]
```

```
for i in range(a):
```

```
    b=int(input('Enter Index '))
```

```
    c=input('Enter Symbol ')
```

```
    l.append([b,c])
```

```
m=[]
```

```
for i in l:
```

```
    if i[0]==0:
```

```
        m.append(i[1])
```

```
    else:
```

```
        m.append(m[i[0]-1]+i[1])
```

```
s=""
```

```
for i in m:
```

```
    s=s+i
```

```
print('Decoded Sequence Is ',s)
```

Output:



```
PS C:\Users\JOY PATEL\Desktop> python .\P10.py
Enter Total Doublets 7
Enter Index 0
Enter Symbol a
Enter Index 0
Enter Symbol b
Enter Index 0
Enter Symbol r
Enter Index 1
Enter Symbol c
Enter Index 1
Enter Symbol d
Enter Index 1
Enter Symbol b
Enter Index 3
Enter Symbol a
Decoded Sequence Is  abracadabra
PS C:\Users\JOY PATEL\Desktop> █
```

Conclusion: LZ78 decompression is an algorithm in which given doublets are given as input and corresponding alphabetic sequence is obtained as output

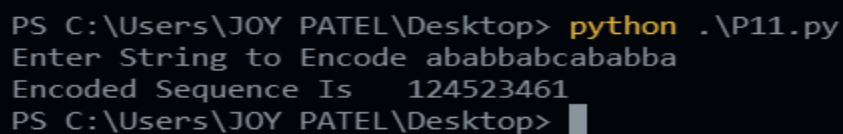
Practical 11

Aim: Write a program to implement LZW compression algorithm

Program:

```
a=input('Enter String to Encode ')
d=[]
for i in a:
    if i not in d:
        d.append(i)
l=[]
s=""
for i in a:
    s=s+i
    if s not in d:
        d.append(s)
        if len(s)!=1:
            l.append(d.index(s[:-1]))
            s=s[:-1]
        else:
            l.append(d.index(s[0]))
            s=s[0]
s=""
for i in l:
    s=s+str(i+1)
print('Encoded Sequence Is ',s+'1')
```

Output:



```
PS C:\Users\JOY PATEL\Desktop> python .\P11.py
Enter String to Encode ababbabababba
Encoded Sequence Is 124523461
PS C:\Users\JOY PATEL\Desktop>
```

Conclusion: LZW is an algorithm in which all the letters are passed through the algorithm and the output is an index value of the character encoded.

Practical 12

Aim: Write a program to implement LZW decompression algorithm

Program:

```
a=input('Enter Sequence to Decode ')
b=int(input('Enter Length of Initial Table '))
l=[]
for i in range(b):
    c=input('Enter Symbol ')
    l.append(c)
m=""
j=0
s=""
for i in a:
    if int(i)<=len(l):
        m=m+(l[int(i)-1])
    else:
        while(j<len(m)):
            s=s+m[j]
            j=j+1
        if s not in l:
            l.append(s)
            s=s[-1]
        if(len(l)>(int(i)-1)):
            break
    if int(i)<=len(l):
        m=m+(l[int(i)-1])
print(m)
```

Output:

```
PS C:\Users\JOY PATEL\Desktop> python .\P12.py
Enter Sequence to Decode 124523461
Enter Length of Initial Table 3
Enter Symbol a
Enter Symbol b
Enter Symbol c
ababbabababba
PS C:\Users\JOY PATEL\Desktop> █
```

Conclusion: LZW decompression is an algorithm in which the index values and the initial dictionary is given as the input and the output is the corresponding alphabetic string