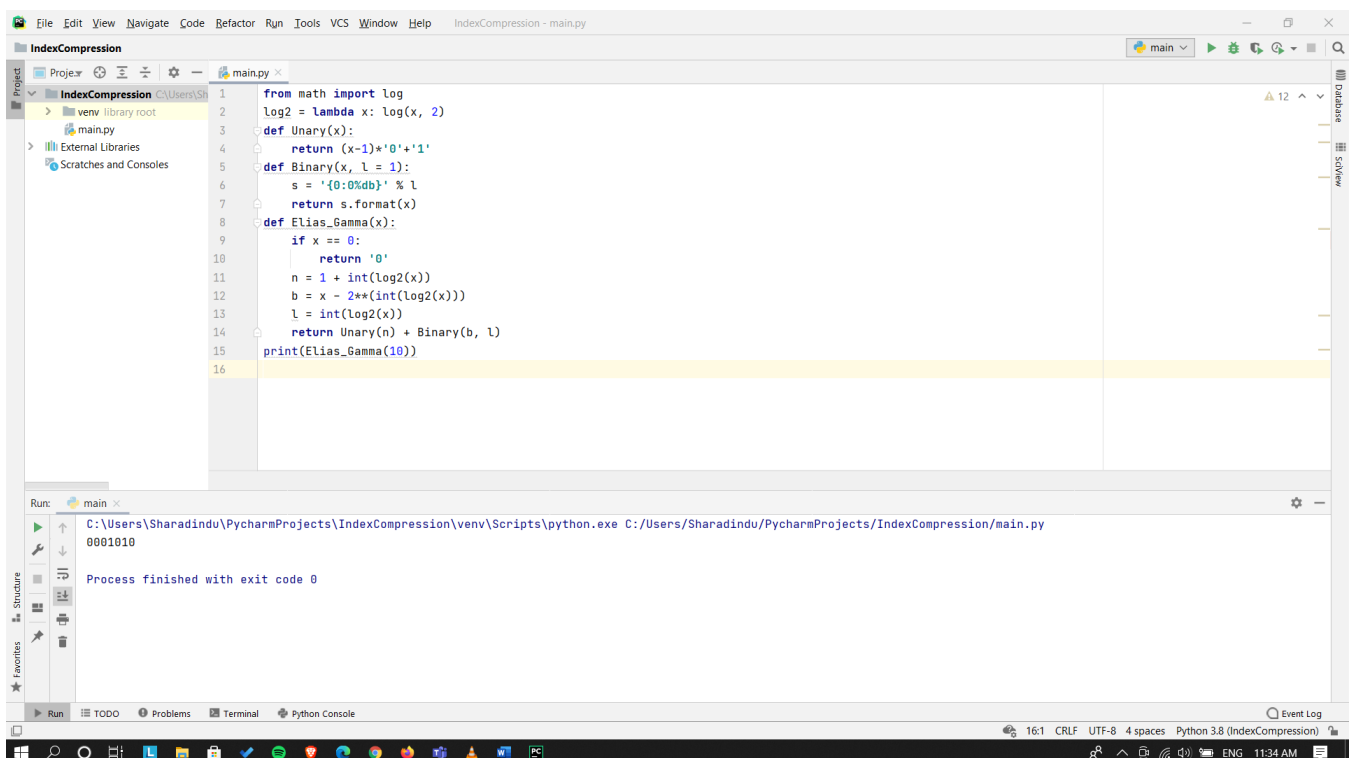


IMPLEMENTATION OF INDEX COMPRESSION TECHNIQUES: Elias gamma and Golomb encoding and decoding.

### Elias Gamma Encoding

```
from math import log
log2 = lambda x: log(x, 2)
def Unary(x):
    return (x-1)*'0'+'1'
def Binary(x, l = 1):
    s = '{0:0%db}' % l
    return s.format(x)
def Elias_Gamma(x):
    if x == 0:
        return '0'
    n = 1 + int(log2(x))
    b = x - 2**(int(log2(x)))
    l = int(log2(x))
    return Unary(n) + Binary(b, l)
print(Elias_Gamma(10))
```



### Elias Gamma Decoding

```
import math
def decode(x):
    num=0;
    for i in range(len(x)):
        num+=(int(x[len(x)-1-i])*(math.pow(2,i))));
```

```

        return num;
x=str(input('Enter code: '));
if(x=='1'):
    print('1');
    exit;
else:
    x=list(x);
    t=0;
    v=[];
    b=0;
    w=[];
    c=0;
    for i in x:
        if(b!=1):
            if(i=='0'):
                t+=1;
            else:
                v.append(i);
                b=1;
        elif(c!=1):
            if(t==0):
                c=1;
                w.append('1');
                w.append(i);
            else:
                v.append(i);
                t-=1;
        else:
            num=decode(v);
            if(num==0):
                break;
            else:
                w.append(i);
                num-=1;
    ans=decode(w);
    print('Original Number: ',int(ans));

```

The screenshot shows the PyCharm IDE interface for a project named 'IndexCompression'. The main.py file is open, showing the following code:

```

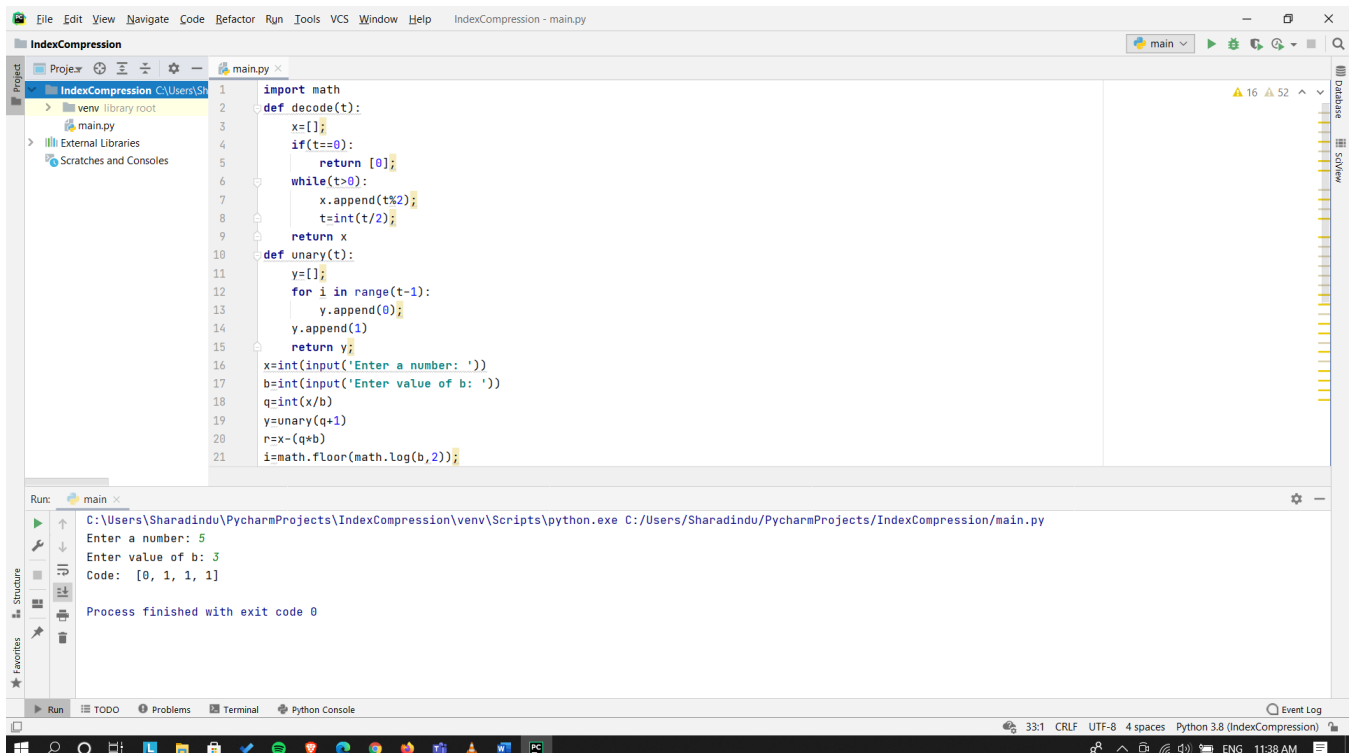
1 import math
2 def decode(x):
3     num=0;
4     for i in range(len(x)):
5         num+=(int(x[len(x)-1-i])*(math.pow(2,i)));
6     return num;
7 x=str(input('Enter code: '));
8 if(x=='1'):
9     print('1');
10    exit;
11 else:
12    x=list(x);
13    t=0;
14    v=[];
15    b=0;
16    w=[];
17    c=0;
18    for i in x:
19        if(b!=1):
20            if(i=='0'):
21                t+=1;

```

The Run console at the bottom shows the execution of the program. The command prompt displays 'Enter code: 0001010' and 'Original Number: 0'. The process finished with exit code 0.

## Golomb Encoding

```
import math
def decode(t):
    x=[];
    if(t==0):
        return [0];
    while(t>0):
        x.append(t%2);
        t=int(t/2);
    return x
def unary(t):
    y=[];
    for i in range(t-1):
        y.append(0);
    y.append(1)
    return y;
x=int(input('Enter a number: '))
b=int(input('Enter value of b: '))
q=int(x/b)
y=unary(q+1)
r=x-(q*b)
i=math.floor(math.log(b,2));
d=math.pow(2,i+1)-b;
if(r>=d):
    r+=int(d);
r2=decode(r);
if(len(r2)<=i and r>=d):
    r2.append(0);
if(len(r2)<i and r<d):
    r2.append(0);
r2=r2[::-1];
y=y+r2;
print('Code: ',y);
```



The screenshot displays the PyCharm IDE interface. The main editor window shows the Python code for Golomb encoding. The code defines two functions: `decode(t)` and `unary(t)`. The `decode` function takes an integer `t` and returns a list of bits. The `unary` function takes an integer `t` and returns a list of bits. The main code block prompts the user to enter a number and a value of `b`, then calculates the Golomb code for the given number and `b`. The Run window at the bottom shows the execution of the code. The output is: Enter a number: 5, Enter value of b: 3, Code: [0, 1, 1, 1]. The process finished with exit code 0.

## Golomb Decoding

```
import math
def decode(x):
    num=0;
    for i in range(len(x)):
        num+=(int(x[len(x)-1-i])*(math.pow(2,i)));
    return num;
x=str(input('Enter code: '))
x=list(x)
b=int(input('Enter value of b: '))
i=math.floor(math.log(b,2))
d=math.pow(2,i+1)-b
p2=0;
l=1;
while(p2<len(x)):
    t=0;
    flag=0;
    r=[];
    k=i;
    q=0;
    for p in range(p2,len(x)):
        if(x[p]=='0' and flag==0):
            t+=1;
            continue;
        if(x[p]=='1' and flag==0):
            q=t;
            flag=1;
            continue;
        r.append(x[p]);
        k-=1;
        if(k==0):
            rnum=decode(r);
            if(rnum<d):
                p2=p+1;
                break;
        if(k==--1):
            rnum=decode(r);
            rnum=rnum-d;
            p2=p+1;
            break;
    ans=q*b+rnum;
    print(ans);
    l=0;
```

The screenshot displays the PyCharm IDE interface. The main editor window shows the Python code for Golomb Decoding, which is identical to the code block above. The code is written in a file named `main.py` within a project named `IndexCompression`. The code defines a `decode` function and a main loop that takes user input for a code string and a value `b`, then prints the decoded result.

Below the editor, the `Run` tab is active, showing the execution output. The command executed is `C:\Users\Sharadindu\PycharmProjects\IndexCompression\venv\Scripts\python.exe C:\Users\Sharadindu\PycharmProjects\IndexCompression/main.py`. The input provided was `Enter code: 10` and `Enter value of b: 3`, resulting in the output `0.0`. The message `Process finished with exit code 0` indicates successful execution.

The bottom status bar shows the current file encoding as `UTF-8`, 4 spaces for indentation, and the Python version as `Python 3.8 (IndexCompression)`. The system clock at the bottom right indicates the time is `11:42 AM` on `ENG`.