

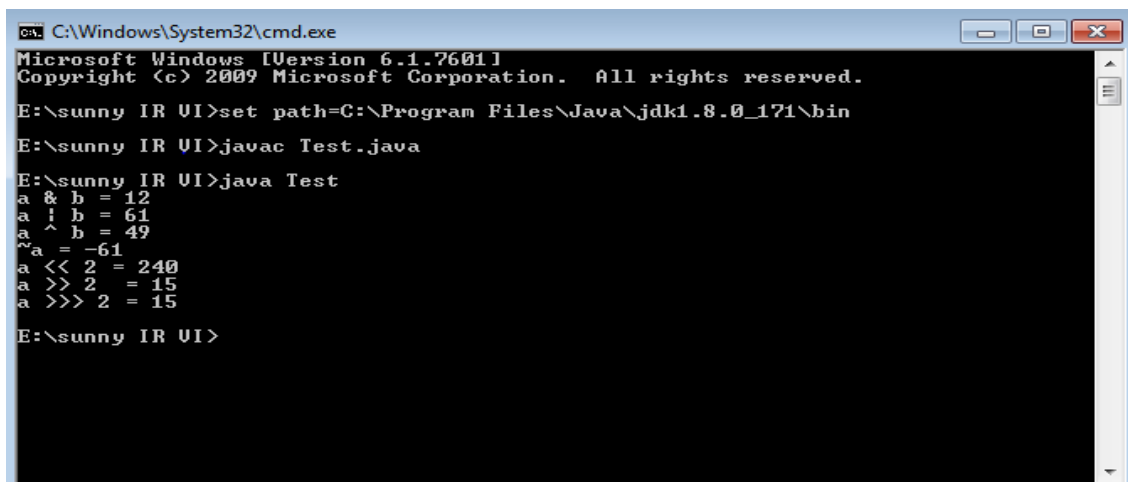
Practical no 1

Aim: Write a program to demonstrate bitwise operation.

Code:

```
public class Test {  
    public static void main(String args[]) {  
        int a = 60;  
        int b = 13;  
        int c = 0;  
        c = a & b;  
        System.out.println("a & b = " + c);  
        c = a | b;  
        System.out.println("a | b = " + c);  
        c = a ^ b;  
        System.out.println("a ^ b = " + c);  
        c = ~a;  
        System.out.println("~a = " + c);  
        c = a << 2;  
        System.out.println("a << 2 = " + c);  
  
        c = a >> 2;  
  
        System.out.println("a >> 2 = " + c);  
  
        c = a >>> 2; /* 15 = 0000 1111 */  
  
        System.out.println("a >>> 2 = " + c);  
    }  
}
```

Output:



```
C:\Windows\System32\cmd.exe  
Microsoft Windows [Version 6.1.7601]  
Copyright (c) 2009 Microsoft Corporation. All rights reserved.  
  
E:\sunny IR UI>set path=C:\Program Files\Java\jdk1.8.0_171\bin  
E:\sunny IR UI>javac Test.java  
E:\sunny IR UI>java Test  
a & b = 12  
a | b = 61  
a ^ b = 49  
~a = -61  
a << 2 = 240  
a >> 2 = 15  
a >>> 2 = 15  
E:\sunny IR UI>
```

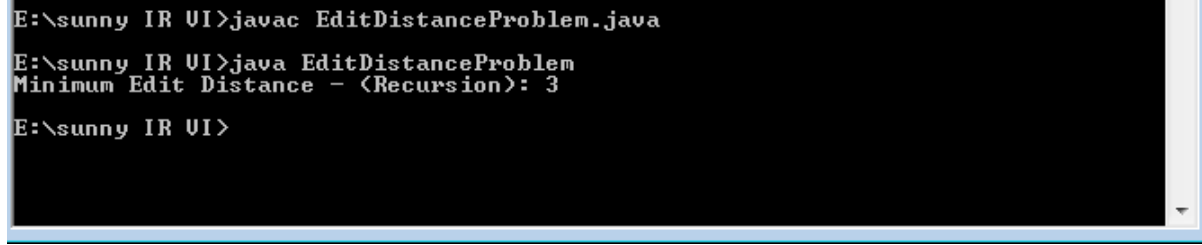
Practical No.3

Aim:- Implement Dynamic programming algorithm for computing the edit distance between

Code:

```
public class EditDistanceProblem
{
    public int editDistanceRecursion(String s1,String s2,int m,int n)
    {
        if(m==0)
            return n;
        if(n==0)
            return m;
        if(s1.charAt(m-1)==s2.charAt(n-1))
            return editDistanceRecursion(s1,s2,m-1,n-1);
        return 1 + Math.min(editDistanceRecursion(s1, s2, m, n-1 ),
            Math.min(editDistanceRecursion(s1, s2 , m-1 , n ),
                editDistanceRecursion(s1 ,s2 , m-1 , n-1 ) ) );
    }
    public static void main(String[] args)
    {
        String s1 = "horizon";
        String s2 = "horizontal";
        EditDistanceProblem ed = new EditDistanceProblem(); System.out.println("Minimum Edit
        Distance - (Recursion): " +
            ed.editDistanceRecursion(s1,s2,s1.length(),s2.length() ) );
    }
}
```

Output:



```
E:\sunny IR UI>javac EditDistanceProblem.java
E:\sunny IR UI>java EditDistanceProblem
Minimum Edit Distance - (Recursion): 3
E:\sunny IR UI>
```

Practical No.7

Aim:- Write a program for Pre-processing of a Text Document: stopwords removal.

Stopwords1.py:-

```
>>> import nltk
>>> from nltk.corpus import stopwords
>>> set(stopwords.words('english'))
```

Output:-

```
{'their', 'has', 'am', 'against', 'our', 'than', 'been', 'an', "shouldn't", 'd',
 'each', 'didn', 'shan', 'won', 'out', "hasn't", 'ourselves', 'down', 'but', 'he
re', 'there', 'them', "shan't", 'in', 'herself', 'these', 'your', 'ma', 'as', 'h
aving', "didn't", "isn't", 'some', 'during', 're', 'and', "she's", 'to', 't', 'a
re', 'which', 'by', 'o', 'those', 'couldn', 'haven', 'until', 'can', 'its', 'onc
e', 'why', 'after', 'the', 'with', 'wasn', "won't", 'doing', 'few', 'should', "m
ustn't", 'through', 'i', 'nor', 'hasn', 'is', 'when', 'me', 'myself', "weren't",
 'this', "you'd", 'yourself', 'both', 'hadn', 'about', 'of', 'will', 'do', 'then
', 'they', 'before', 'y', 'only', 'had', 'all', "it's", 'off', 'above', 'up', 't
heirs', 'between', 'll', 'over', 'wouldn', 'being', 'same', 'aren', 'did', 'a',
 'any', 's', 'how', 'weren', 'just', 've', 'because', 'again', 'don', 'him', 'him
self', 'we', 'more', 'she', "aren't", "wasn't", 'for', "you've", 'what', 'itself
', 'hers', 'have', "you'll", 'who', 'too', 'under', 'mightn', 'doesn', 'it', 'in
to', 'so', "should've", "mightn't", 'does', 'were', 'mustn', 'shouldn', 'other',
 'no', 'he', 'yourselves', "doesn't", 'you', 'themselves', 'below', 'or', 'on',
 'needn', "wouldn't", 'that', "hadn't", "that'll", 'from', "don't", 'ain', 'such
', 'further', 'where', "needn't", 'whom', 'very', 'my', "haven't", 'own', "you're
", 'while', 'yours', 'ours', 'was', 'm', 'her', 'isn', 'if', 'not', 'his', "coul
dn't", 'now', 'at', 'be', 'most'}
```

Stopwords1.py:-

```
from nltk.corpus import stopwords

from nltk.tokenize import word_tokenize

example_sent = "This is a sample sentence, showing off the stop words filtration."

stop_words = set(stopwords.words('english'))

word_tokens = word_tokenize(example_sent)

filtered_sentence = [w for w in word_tokens if not w in stop_words]

filtered_sentence = []

for w in word_tokens:
    if w not in stop_words:
```

```
filtered_sentence.append(w)
```

```
print(word_tokens)
```

```
print(filtered_sentence)
```

Output:-

```
===== RESTART: D:/IR/IR/stopwords2.py =====  
['This', 'is', 'a', 'sample', 'sentence', ',', 'showing', 'off', 'the', 'stop',  
'words', 'filtration', '.']  
['This', 'sample', 'sentence', ',', 'showing', 'stop', 'words', 'filtration', '.']  
>>>
```

Practical No.8

Aim:- Write a program for tkinter.

Code:

```
from tkinter import *
root=Tk()
l1=Label(root,text="Enter Number 1:")
l1.pack()
t1=Entry(root,bd="3")
t1.pack()
l2=Label(root,text="Enter Number 2:")
l2.pack()
t2=Entry(root,bd="3")
t2.pack()
def addNumber():
    a=int(t1.get())
    b=int(t2.get())
    c=a+b
    print("Addition of two NOS:",C)
b1=Button(root,text="Addition",fg="red",bg="green",command=addNumber) b1.pack()
root.mainloop()
```

Output:-

