TOC PRACTICAL

Practical-1

```
pip install nltk

import nltk

from nltk.tokenize import sent_tokenize

text="Hello everyone. How are you"

sent_tokenize(text)


from nltk.tokenize import word_tokenize

text="Hello everyone how are you"

word_tokenize(text)

tokenizer= nltk.data.load('tokenizers/punkt//english.pickle')

tokenizer.tokenize(text)

spanish_tokenizer=nltk.data.load('tokenizers/punkt//spanish.pickle')

text='Hola amigo. Estoy bien'

tokenizer.tokenize(text)
```

Practical-2

normal

```
import re

line= "horses are taller than dogs";

searchObj= re.search(r'(.*) are (.*?) .*', line, re.M|re.I)

if searchObj:

  print("searchObj.group():", searchObj.group())
```

```
  print("searchObj.group(1):", searchObj.group(1))
  print("searchObj.group(2):", searchObj.group(2))
else:
  print("Nothing Found")
```

case sensitive

```
line= "horses ARE taller than dogs";
searchObj= re.search(r'(.*) are (.*?) .*', line, re.M)
if searchObj:
  print("searchObj.group():", searchObj.group())
  print("searchObj.group(1):", searchObj.group(1))
  print("searchObj.group(2):", searchObj.group(2))
else:
  print("Nothing Found")
```

to check for multiline

```
line = '''horses ARE taller than dogs\hsjdfjd''';
searchObj= re.search(r'(.*) are (.*?) .*', line, re.M|re.I)
if searchObj:
  print("searchObj.group():", searchObj.group())
  print("searchObj.group(1):", searchObj.group(1))
  print("searchObj.group(2):", searchObj.group(2))
else:
  print("Nothing Found")
```

```
line = '''horses ARE taller than dogs\hsjdfjd''';

searchObj= re.search(r'(.*) are (.*?) ', line, re.M|re.I)

if searchObj:

  print("searchObj.group():", searchObj.group())

  print("searchObj.group(1):", searchObj.group(1))

  print("searchObj.group(2):", searchObj.group(2))

else:

  print("Nothing Found")
```

Practical-3

Derivation sequence

```
def printArray(arr,size):
    for i in range(size):
        print(arr[i],end="")
    print()
    return


def getsuccessor(arr,k,n):
    p=k-1
    while(arr[p]==n and 0<=p<k):
```

```
        p-=1
        if (p<0):
            return 0
    arr[p]= arr[p]+1
    i=p+1
    while(i<k):
        arr[i]=1
        i+=1
    return 1
def printseq(n,k):
    arr=[0]*k
    for i in range(k):
        arr[i]=1
    while(1):
        printArray(arr,k)
        if(getsuccessor(arr,k,n)==0):
            break
    return

n=4
```

```
k=2
printseq(n,k)
```

Practical 4- 3 consecutives a’s s

```
def stateA (n):
    if(n[0]=='a'):
        stateB (n[1:])
    elif(n[0]=='b'):
        stateH (n[1:])

def stateB (n):
    if (len(n)==0):
        print("string not accepted")
    else:
        if(n[0]=='a'):
            stateC(n[1:])
        elif (n[0]=='b'):
            stateI(n[1:])

def stateC (n):
    if (len(n)==0):
        print("string not accepted")
    else:
```

```
        if(n[0]=='a'):
            stateD (n[1:])
        elif(n[0]=='b'):
            stateJ (n[1:])

def stateD (n):
    if (len(n)==0):
        print("string not accepted")
    else:
        if(n[0]=='a'):
            stateQ2 (n[1:])
        elif(n[0]=='b'):
            stateE (n[1:])

def stateE (n):
    if (len(n)==0):
        print("string not accepted")
    else:
        if(n[0]=='a'):
            stateQ2 (n[1:])
        elif(n[0]=='b'):
            stateF (n[1:])
```

```
def stateF (n):
    if (len(n)==0):
        print("string not accepted")
    else:
        if(n[0]=='a'):
            stateQ2 (n[1:])
        elif(n[0]=='b'):
            stateG (n[1:])

def stateG (n):
    if (len(n)==0):
        print("string accepted")
    else:
        if(n[0]=='a'):
            stateQ2 (n[1:])
        elif(n[0]=='b'):
            stateQ1 (n[1:])

def stateH (n):
    if (len(n)==0):
        print("string not accepted")
    else:
        if(n[0]=='a'):
```

```
            stateI (n[1:])
        elif(n[0]=='b'):
            stateK (n[1:])

def stateI (n):
    if (len(n)==0):
        print("string not accepted")
    else:
        if(n[0]=='a'):
            stateJ (n[1:])
        elif(n[0]=='b'):
            stateL (n[1:])

def stateJ (n):
    if (len(n)==0):
        print("string not accepted")
    else:
        if(n[0]=='a'):
            stateE (n[1:])
        elif(n[0]=='b'):
            stateM (n[1:])

def stateK (n):
```

```
    if (len(n)==0):
        print("string not accepted")
    else:
        if(n[0]=='a'):
            stateL (n[1:])
        elif(n[0]=='b'):
            stateN (n[1:])

def stateL (n):
    if (len(n)==0):
        print("string not accepted")
    else:
        if(n[0]=='a'):
            stateM (n[1:])
        elif(n[0]=='b'):
            stateO (n[1:])

def stateM (n):
    if (len(n)==0):
        print("string not accepted")
    else:
        if(n[0]=='a'):
            stateF (n[1:])
```

```
        elif(n[0]=='b'):
            stateP (n[1:])

def stateN (n):
    if (len(n)==0):
        print("string not accepted")
    else:
        if(n[0]=='a'):
            stateO (n[1:])
        elif(n[0]=='b'):
            stateQ1 (n[1:])

def stateO (n):
    if (len(n)==0):
        print("string not accepted")
    else:
        if(n[0]=='a'):
            stateP (n[1:])
        elif(n[0]=='b'):
            stateQ1 (n[1:])

def stateP (n):
    if (len(n)==0):
```

```
        print("string not accepted")
    else:
        if(n[0]=='a'):
            stateG (n[1:])
        elif(n[0]=='b'):
            stateQ1 (n[1:])

def stateQ1(n):
    print("string not accepted")

def stateQ2(n):
    print("string not accepted")

n="aaabb"
stateA(n)
```

practical-5

string ending with 10or 01

```
def stateq1(s,i):
    print("q1-->", end="");
    if(i==len(s)):
        print("no");
        return;
```

```
    if(s[i]=='0'):
        stateq1(s,i+1);
    else:
        stateq3(s, i+1);


def stateq2(s,i):
    print("q2-->", end="");
    if(i==len(s)):
        print("no");
        return;
    if(s[i]=='0'):
        stateq4(s,i+1);
    else:
        stateq2(s, i+1);

def stateq3(s,i):
    print("q3-->", end="");
    if(i==len(s)):
        print("yes");
        return;
    if(s[i]=='0'):
        stateq4(s,i+1);
    else:
        stateq2(s, i+1);
```

```
def stateq4(s,i):
    print("q4-->", end="");
    if(i==len(s)):
        print("yes");
        return;
    if(s[i]=='0'):
        stateq1(s,i+1);
    else:
        stateq3(s, i+1);

def stateq0(s,i):
    print("q0-->", end="");
    if(i==len(s)):
        print("no");
        return;
    if(s[i]=='0'):
        stateq1(s,i+1);
    else:
        stateq2(s, i+1);

s="100100";
print("transition states are:", end="");
stateq0(s,0);
```

Practical-6

End with 101

```
def q1(s,i):
    print("q1-->",end="");
    if(i==len(s)):
        print("no");
        return;
    if(s[i]=='0'):
        q2(s,i+1);
    else:
        q0(s,i+1);


def q2(s,i):
    print("q2-->",end="");
    if(i==len(s)):
        print("no");
        return;
```

```
    if(s[i]=='0'):
        q0(s,i+1);
    else:
        q3(s,i+1);

def q3(s,i):
    print("q3-->",end="");
    if(i==len(s)):
        print("yes");
        return;
    if(s[i]=='0'):
        q0(s,i+1);
    else:
        q1(s,i+1);

def q0(s,i):
    print("q0-->",end="");
    if(i==len(s)):
        print("no");
        return;
```

```
    if(s[i]=='0'):
        q0(s,i+1);
    else:
        q1(s,i+1);

s="101";
print("state transitions are:", end="")
q0(s,0);
```

practical-7

accept a number that is divisible by 2

```
def stateq0 (n):
    if (len(n)==0):
        print("string accepted");
    else:
        if(n[0]=='0'):
            stateq0(n[1:])
        elif(n[0]=='1'):
            stateq1(n[1:])
```

```
def stateq1 (n):
    if (len(n)==0):
        print("string not accepted");
    else:
        if(n[0]=='0'):
            stateq1(n[1:])
        elif(n[0]=='1'):
            stateq0(n[1:])

n=int(input())
n=bin(n).replace("0b","")
stateq0(n)
```

practical-8

equal no of 0,1,2

```
def get(s):
    arr=[];
    n=len(s);

    for i in range(n):
        for j in range(i,n):
```

```
        s1=" "
        for k in range(i,1+j):
            s1+=s[k];
        arr.append(s1);
count=0;

for i in range(len(arr)):
    countzero=0;
    countone=0;
    counttwo=0;
    curs=arr[i];

    for j in range(len(curs)):
        if(curs[j]=='0'):
            countzero+=1;
        if(curs[j]=='1'):
            countone+=1;
        if(curs[j]=='2'):
            counttwo+=1;
```

```
        if(countzero== countone and countone== counttwo):
            count+=1;
    return count;
str="0110201110";
print(get(str));
```

practical-9

count no of 0 and 1

```
def count(s,n):
    ans=0;
    i=0;
    while(i<n):
        cnt0=0; cnt1=0;
        if(s[i]=='0'):
            while(i<n and s[i]=='0'):
                cnt0+=1;
                i+=1;
```

```
    j=1;

    while(j<n and s[j]=='1'):
        cnt1+=1;
        j+=1;

else:
    while(i<n and s[i]=='1'):
        cnt1+=1;
        i+=1;

    j = i;

    while(j<n and s[j]=='0'):
        cnt0+=1;
        j+=1;

    ans+=max(cnt0, cnt1);
```

```
    return ans;


if __name__=="__main__":
    s="110000111000";
    n=len(s);
    print(count(s,n))
```