Problem Solving And Programming

Date 12 June 2019

Day Objectives

- String Slicing
- Function in Python
- Basic Problems related to conditional statements using functions
- Iteration in python
- Python Data Structures-lists,tuples and dictionaries
- Basic operations on data structures
- Applying Data Structures to solve problems

In [ ]:

```

```

Srting Slicing

In [22]:

```
s1="python"

s1[0] ###Accessing the first character in a string

s1[1] ###Accessing the second character in a string

s1[len(s1)-1] ### Accessing the last character in a string

s1[-1] ### Accessing the last character in a string

s1[-2] ### Accessing the last penultimate character in a string

s1[0:2] ### Accessing the first two characters in a string

s1[-2:] ### Accessing the last two characters in a srring

s1[4:] ### Accessing the character 5th character to end of a string

### Accessing all characters except first and last character

s1[1:-1]

#### Accessing the middle character in a string

s1[len(s1)//2]

s1[len(s1)//3]

###Reverse of a string
s1[-1::-1]

s1[-1:-3:-1] ## Accessing the last two characters in  a reverse string

#reverse the middle two characters in an even length string

s1[len(s1)//2:(len(s1)//2)-2:-1]

##Alternative characters of a string same oreder
### "python" -"pto"

s1[::2]

##Acceessing alternate character of a string in reverse order
###"python" -"nhy"

s1[::-2]
```

Out[22]:

```
'nhy'
```

In [15]:
```
s1[0]

```
Out[15]:
```
'p'
```

In [11]:
```
s1[-1]
```
Out[11]:
```
'v'
```

In [13]:
```
s1[len(s1)-2]
```
Out[13]:
```
'c'
```

In [ ]:
```

```

functions

In [28]:
```
###Function to reverse of a string
def reverseString(s1):
    return s1[::-1]

reverseString("Python")
```
Out[28]:
```
'nohtyP'
```

In [38]:
```
###Function to check if a string is Palindrome or not
s=str(input("enter the string: "))
def palindrome(s):
    if s==s[::-1]:
        return True
    else:
        return False
palindrome(s)
```
```
enter the string: 101
```
Out[38]:
```
True
```

In [43]:
```
###Function to check if a given year is leap year
y=int(input("Enter the year:"))
def year(y):
        if(y%400==0 or(y%100!=0 and y%4==0)):
            return True
        else:
            return False

year(y)

```
```
Enter the year:1996
```
Out[43]:
```
True
```

In [11]:

```
### count no of digits
n=int(input("Enter the number:"))
def num(n):
    c=0
    while n!=0:
         c=c+1
         n=n//10


    return c
num(n)

```

```
Enter the number:123
```

Out[11]:

```
3
```

In [1]:

```
def number(n):
    return len(n)
n=(input("Enter the string:"))
number(n)
```

```
Enter the string:123
```

Out[1]:

```
3
```

In [17]:

```
def greatest(n1,n2,n3,n4):
    if n1 > n2 and n1 > n3 and n1 > n4:
        return n1
    elif n2 > n3 and n2 > n4:
        return n2
    elif n3 > n4:
        return n3
    return n4
greatest(1,12,13,14)
```

Out[17]:

```
14
```

In [ ]:

```
### Iterations
- for
- while
#### for Loop in python
[lb,ub]
  for number in range (lb,ub+1)

```

In [29]:

```python
###Function to print n natural nos

def printnNaturalNumbers(n):
    for i in range(1,n+1):
        print(i,end=" ")
    return

printnNaturalNumbers(30)
printnNaturalNumbers(10)



```

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 3
0 1 2 3 4 5 6 7 8 9 10
```

In [32]:

```python
###Function to print N natural numbers using while loop

def natural(n):
    i=1
    while i <= n:
        print(i,end=" ")
        i=i+1
    return
natural(20)
```

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

In [11]:

```python
#Function to print the alternative values in a range
#[500,550] ->500 502 504.........550
#(500,550) ->501 503 505 .........549
#range(500,550) ->500 501 502 ..... 549
#All set based functions in python have start value inclusive end value is e



###write a program to print alternate values
def alternatevalue(start,end):
    for i in range(start,end+1,):
        print(i,end=" ")
    return
alternatevalue(500,525)

```

```
500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519
520 521 522 523 524 525
```

In [14]:

```
###function to print reverse of a given range in the same
def reverserange(start,end):
    for i in range(end,start,-1):
        print(i,end=" ")
    return
reverserange(500,525)

```

```
525 524 523 522 521 520 519 518 517 516 515 514 513 512 511 510 509 508 507 506
505 504 503 502 501
```

In [20]:

```
#function to print the odd nos in a range

def odd(start,end):
    for i in range(end,start,-1):
        if(i%2!=0):
            print(i,end=" ")
    return

odd(1,10)
```

```
9 7 5 3
```

In [38]:

```
#Function to print sumof numbers in a range

def sumnumbers(start,end):
    sum=0
    for i in range(start,end+1):
            sum=sum+i
    return sum

sumnumbers(1,10)




```

Out[38]:

```
55
```

In [41]:

```
#Function to calculate the average of a given range

def average(start,end):
    sum=0
    c=0
    for i in range(start,end+1):
            sum=sum+i
            c=c+1
    return sum//c
average(1,5)




```

Out[41]:

```
3
```

In [10]:

```
#### Function to print all numbers divisible by 6 and not a factor of 100 in
a=int(input("Enter the first range:"))
b=int(input("Enter the second range:"))
def divisible(a,b):
    for i in range(a,b+1):
        if i%6==0 and i%100!=0:
            print(i,end=" ")
    return
divisible(a,b)
```

```
Enter the first range:1
Enter the second range:100
6 12 18 24 30 36 42 48 54 60 66 72 78 84 90 96
```

In [17]:

```
## Function to find the average of cubes of all even numbers ina a given ran
a=int(input("Enter the first range:"))
b=int(input("Enter the second range:"))
def avg(a,b):
    sum=0
    count=0
    for i in range(a,b+1):
        if i%2==0:
            sum +=i ** 3
            count +=1
    return(sum/count)


avg(a,b)
```

```
Enter the first range:1
Enter the second range:3
```

Out[17]:

```
8.0
```

In [ ]:

```
### Function to generate the list of factors of a given number
n=int(input("Enter the number:"))
def fact(n):
    for i in range(1,n+1):
        if(n%i==0):
            print(i,end=" ")

fact(n)

```

In [12]:

```
### Function to calculate of a given number
n=int(input("Enter the number:"))
def factorial(n):
    f=1
    for i in range(1,n+1):
        f=f*i;
    return f

factorial(n)

```

```
Enter the number:12
```

Out[12]:

```
479001600
```

In [2]:

```
###Function to check if a given no is prime
n=int(input("Enter the number:"))
def prime(n):
    c=0
    for i in range(1,n+1):
        if(n%i==0):
            c=c+1
    if(c==2):
        print("prime")
    else:
        print("Not a prime")
prime(n)

```

```
Enter the number:5
prime
```

In [8]:

```
i=1
x = int(input("Enter the number:"))
for k in range (1, (x+1), 1):
    c=0
    for j in range (1, (i+1), 1):
        a = i%j
        if (a==0):
            c = c+1

    if (c==2):
          print (i)
    else:
          k = k-1

    i=i+1
```

```
Enter the number:7
2
3
5
7
```

In [20]:

```
####average of the given prime number
n=int(input("Enter the first  number:"))
n1=int(input("Enter the second number"))
def prime(n,n1):
    sum=0
    for i in range(n,n1+1):
        c=0

        for j in range(1,i+1):
            if(i%j==0):
                 c=c+1
    if(c==2):
        sum=sum+i

    return sum//c

prime(n,n1)

```

```
Enter the first  number:1
Enter the second number7
```

Out[20]:

```
3
```

Advanced Problems

In [ ]:

```
###average of the factorials
n1=int(input("enter the first range:"))
n2=int(input("enter the second range:"))
def factorials(n1,n2):
    t=0
    count=0
    for i in range(n1,n2+1):
        sum=1
        for j in range(1,i+1):
            sum=sum*j
        count=count+1
        t=t+sum
    print(t/count)

factorials(n1,n2)


```

In [ ]:

```
###Function to a table
n=int(input("Enter the number:"))
lb=int(input("Enter the lower bound:"))
ub=int(input("Enter the upper bound:"))
def table(n,lb,ub):
    for i in range(lb,ub+1):


        print(n,"x",i,"=",n*i)

    return


table(n,lb,ub)
```

In [9]:

```
### odd Armstrong numbers in a given range
n=int(input("Enter the number:"))
def amstrong(n):
        count=0
        i=1
        while(count<n):
            j=i
            rev=0
            while(j!=0):
                rem=j%10
                rev=rev+rem**3
                j=j//10
            if rev==i and i%2==1:
                print(i,end=" ")
                count=count+1
            i=i+1
amstrong(n)


```

```
Enter the number:3
1 153 371
```

In [36]:

```python
#Function to generate all leap years in a given time period
# 2000-2020 ->2000 2004 2008 2012 2016 2020

def isLeapYear(y):
    if y%400==0 or (y%100!=0 and y%4==0):
        return True
    else:
        return False

isLeapYear(2016)



def generateLeapYears(startyear,endyear):
    for year in range(startyear,endyear+1):
        if isLeapYear(year):
            print(year,end=" ")
    return
generateLeapYears(1919,2019)

```

```
1920 1924 1928 1932 1936 1940 1944 1948 1952 1956 1960 1964 1968 1972 1976 1980
1984 1988 1992 1996 2000 2004 2008 2012 2016
```

In [41]:

```python
def isLeapYear(y):
    if y%400==0 or (y%100!=0 and y%4==0):
        return True
    else:
        return False
def numberofDays(startyear, endyear):
    sum=0
    for y in range(startyear, endyear+1):
        if isLeapYear(y):
            sum=sum+366
        else:
            sum=sum+365
    return sum
numberofDays(2001,2003)
```

Out[41]:

```
1095
```

In [49]:

```python
#Function to calculate number of hours for a given period
#number of hours(11,1975,3,1999)->2014504 or 2015248
#number of hours(5,2019,6,2019)->1464
#2,2017,6,2019
#No of hours=24*No of days
#3 steps
  #1.start month year to end of year-calculate no
  #2.calculate days for all years b/w start year

#excluding feb
#first six months- 1,3,4,5,6,7
                    #All odd months have





def numberofdaysmonth(month,year):
    if(month==2):
        if isLeapYear(year):
            return 29
        return 28
    elif (month<=7 and month % 2!= 0)or(month >7 and month %2==0):
        return 31
    else:
        return 30

def daysinstartyear(startmonth,startyear):
    days=0
    for month in range(startmonth,13):
        days += numberofdaysmonth(month,startyear)
    return days

def daysinendyear(endmonth,endyear):
    days=0
    for month in range(1,endmonth+1):
        days += numberofdaysmonth(month,endyear)
    return days

def numberofhours(startmonth,startyear,endmonth,endyear):
    days=0
    if startyear!=endyear:
        days += daysinstartyear(startmonth,startyear)
        days += daysinendyear(endmonth,endyear)

        if endyear-startyear==2:
            days += numberofDays(startyear+1, startyear+1)
        elif endyear-startyear > 2:
            days += numberofDays(startyear+1,endyear-1)
    else:
        for month in range(startmonth,endmonth+1):

            days += numberofdaysmonth(month,startyear)
    return 24 * days

numberofhours(11,1975,3,1999)
```

Out[49]:

```
205248
```

In [31]:

```
#Function to print all numbers
```

In [ ]:

```

```