String Slicing:

In [ ]:

In [ ]:

In [ ]:

Iterations in python:

For loop:

[101,210]

for number in range(101,210+1) O/P:- 101 102 103 . . . 210

In [18]:

```

# Function to print n natural numbers

def printNnaturalNumber(n):
    for counter in range (1,n+1):
        print(counter,end=" ")
       # print()
    return
printNnaturalNumber(10)
```

```
1aa 2aa 3aa 4aa 5aa 6aa 7aa 8aa 9aa 10aa
```

In [32]:

```
# Function to print the alternamte values in a range
# [500,550] -> 500 502 503 .... 550
# For all set based functions in python have start values is inclusive and end value is exc
# (500,550)-> 500 501 502 .... 549(Exclusive)
def alternates(n,n1):
    for n in range(n,n1+1,2):  # (init,until,steps)
        print(n,end=" ")
    return
alternates(500,550)
```

```
500 502 504 506 508 510 512 514 516 518 520 522 524 526 528 530 532 534 536
538 540 542 544 546 548 550
```

In [17]:

```
# Function to print reverse of a range in the same

def revAlternate(n,n1):
    for n1 in range(n1,n-1,-1):
        print(n1,end=" ")
    return
revAlternate(500,550)
```

```
550 549 548 547 546 545 544 543 542 541 540 539 538 537 536 535 534 533 532
531 530 529 528 527 526 525 524 523 522 521 520 519 518 517 516 515 514 513
512 511 510 509 508 507 506 505 504 503 502 501 500
```

In [16]:

```
# Print odd numbers in reverse order in a range
def revOdd(start,end):
    for end in range(end,start-1,-1):
        if end%2!=0 :
            print(end,end=" ")
    return
revOdd(500,550)
```

```
549 547 545 543 541 539 537 535 533 531 529 527 525 523 521 519 517 515 513
511 509 507 505 503 501
```

In [26]:

```
# Function to calculate the sum of numbers in a range


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

sumOfnum(int(input("enter start value:")),int(input("enter end value:")))
```

```
enter start value:1
enter end value:3
```

Out[26]:

```
6
```

In [25]:

```
# Function to generate Multiplication table for a number

def printTable(number,start,end):
    for start in range(start,end+1):
        print(number,"*",start,"=",number*start)
    return
printTable(10,100,110)
```

```
10 * 100 = 1000
10 * 101 = 1010
10 * 102 = 1020
10 * 103 = 1030
10 * 104 = 1040
10 * 105 = 1050
10 * 106 = 1060
10 * 107 = 1070
10 * 108 = 1080
10 * 109 = 1090
10 * 110 = 1100
```

In [35]:

```
# Functions to calculate average of all factorials in a given range

def calulateNfactorials(start,end):
    sum = 0
    counter =0
    for start in range(start,end+1):
        sum+=fact(start)
        counter+=1
    return sum//counter
def fact(number):
    res = 1
    for i in range(1,number+1):
        res = res*i
    return res
calulateNfactorials(1,3)
```

Out[35]:

```
3
```

In [29]:

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

def calcAVGofRange(start,end):
    sum = 0
    counter = 0
    for start in range(start,end+1):
        sum += start
        counter +=1
    return sum//counter
calcAVGofRange(int(input("enter start value:")),int(input("enter end value:")))
```

```
enter start value:1
enter end value:3
```

Out[29]:

```
2
```

In [48]:

```
def arms(l,u):
    for k in range(l,u+1):
        count=0
        n=k
        t=n
        s=t
        sum=0
        while(n>0):
            n=n//10
            count+=1
        while(t>0):
            a=(t%10)**count
            sum=sum+a
            t=t//10
        if(s==sum):
                print(s,end=" ")

arms(1,100000)
```

```
1 2 3 4 5 6 7 8 9 153 370 371 407 1634 8208 9474 54748 92727 93084
```

In [50]:

```
#  Function to generate all leap years in a given time period

def printLeapsInRange(start,end):
    for start in range(start,end+1):
        checkLeap(start)
    return
def checkLeap(number):
    if(number%400==0 or (number%100!=0 and number%4==0)):
        print(number,end=" ")
    return
printLeapsInRange(2000,2020)
```

```
2000 2004 2008 2012 2016 2020
```

In [52]:

```
# Function to print all numbers divisible by 6
# and Not a factor of 100 in a given range(lb,ub)inclusive

def printNumbers(start,end):
    for start in range(start,end+1):
        if(start%6==0 and 100%start!=0):
            print(start)
    return
printNumbers(1,24)
```

```
6
12
18
24
```

In [54]:

```
# Function to generate N odd armstrong numbers

def generateNumbers(n):
    for i in range(1,n+1):
        checkAms(i)
    return
def checkAms(number):
    sum=0
    temp = number
    while(number>0):
            a=(number%10)**len(str(number))
            sum=sum+a
            number = number//10
    if(temp == sum and temp%2!=0):
        print(sum)
    return
generateNumbers(100)
```

```
1
3
5
7
9
89
```

In [ ]: