KODATI BHANUSRI

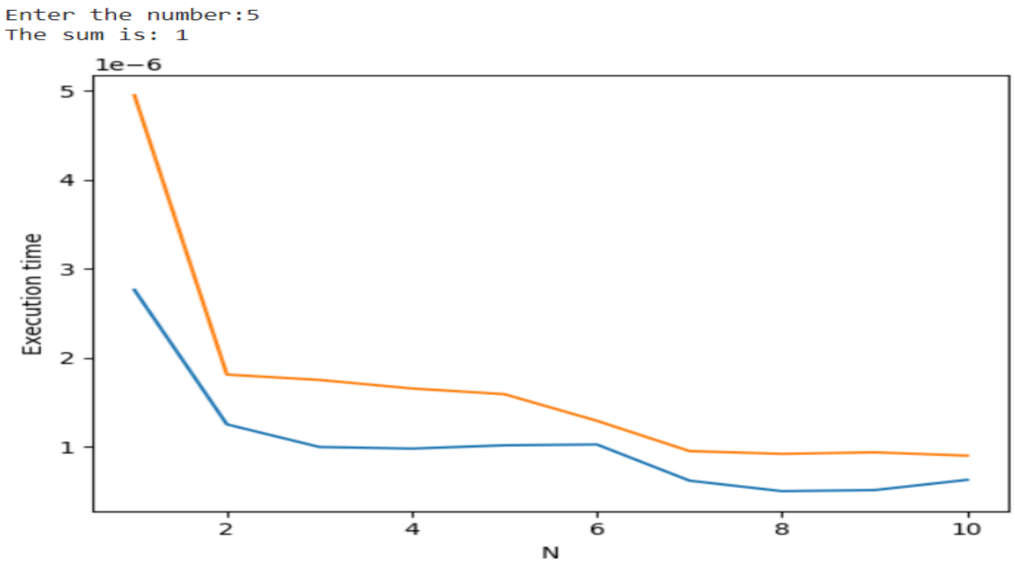
CSE-B

BL.EN.U4CSE21100

1. Find the sum of first N natural numbers using Iterative and Recursive algorithms. Find the time taken to execute the same by varying 'N's value and plot it using python's plot function.

```
import matplotlib.pyplot as plt
import time
#iterative algorithm
def isum_n(n):

  sum = 0
  for i in range(1, n+1):

    sum += i
    return sum
#recursive algorithm
def rsum_n(n):

  if n == 1:
    return 1
  else:
    return n + rsum_n(n-1)

it_sum = []
re_sum = []

n = int(input("Enter the number:"))
x = isum_n(n)
print("The sum is:",x)

for _ in range(10):

  start_time = time.perf_counter()
  iterative_sum = isum_n(n)
  end_time = time.perf_counter()
  it_sum.append(end_time - start_time)
  start_time = time.perf_counter()
  recursive_sum = rsum_n(n)
  end_time = time.perf_counter()
  re_sum.append(end_time - start_time)

plt.plot(range(1,11), it_sum, label = "Iterative sum")
plt.plot(range(1,11), re_sum, label = "Recursive sum")
plt.xlabel('N')
plt.ylabel('Execution time')
plt.show()
```

OUTPUT:



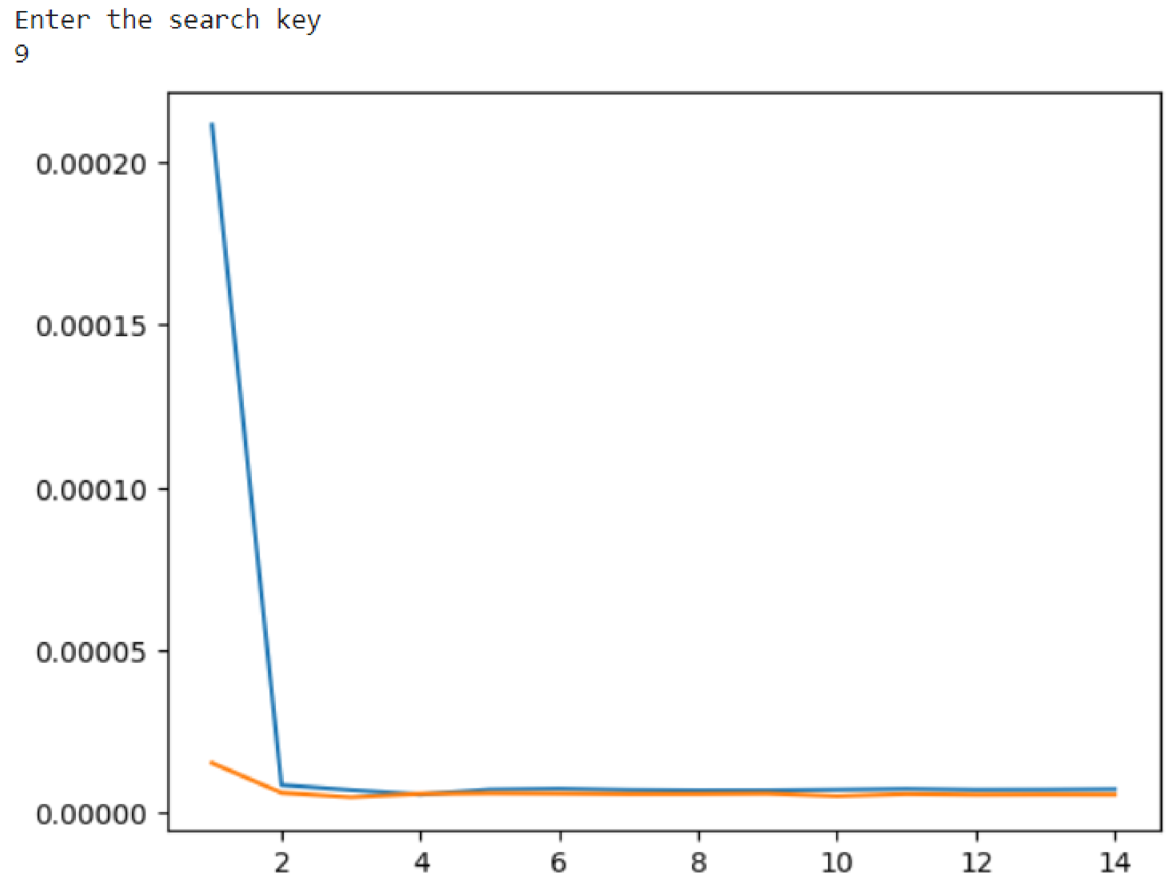
```
Enter the number:5
The sum is: 1
```

2. Perform linear and binary searches for an array of 10000 elements. Use random function in Python to generate the integer array elements in the range 1 to 1000. The search key is an input given by the user. Plot the time taken by the algorithm for 5 different searches when executing the two algorithms.

```
import matplotlib.pyplot as plt
import random
import time
def linearsearch(n,key):
 for i in n:
  if i == key:
    return 1
    return 0
def binarysearch(n,key):
 left = 0
 right = len(n) - 1
 while left < right:
  mid = (left + right) // 2
  if n[mid] == key:
    return mid
  elif key < mid:
    right = mid - 1
  else:
   left = mid + 1
arr = [random.randint(1,1000) for _ in range(10000)]
key = int(input("Enter the search key\n"))
l_search = []
b_search = []
```

```
for _ in range(14):
 start_time = time.perf_counter()
 lin_re = linearsearch(arr, key)
 end_time = time.perf_counter()
 l_search.append(end_time - start_time)
 arr.sort()
 start_time = time.perf_counter()
 bin_re = binarysearch(arr, key)
 end_time = time.perf_counter()
 b_search.append(end_time - start_time)
plt.plot(range(1,15), l_search, label = "Linear Search")
plt.plot(range(1,15), b_search, label = "Binary Search")
plt.show()
```

OUTPUT:

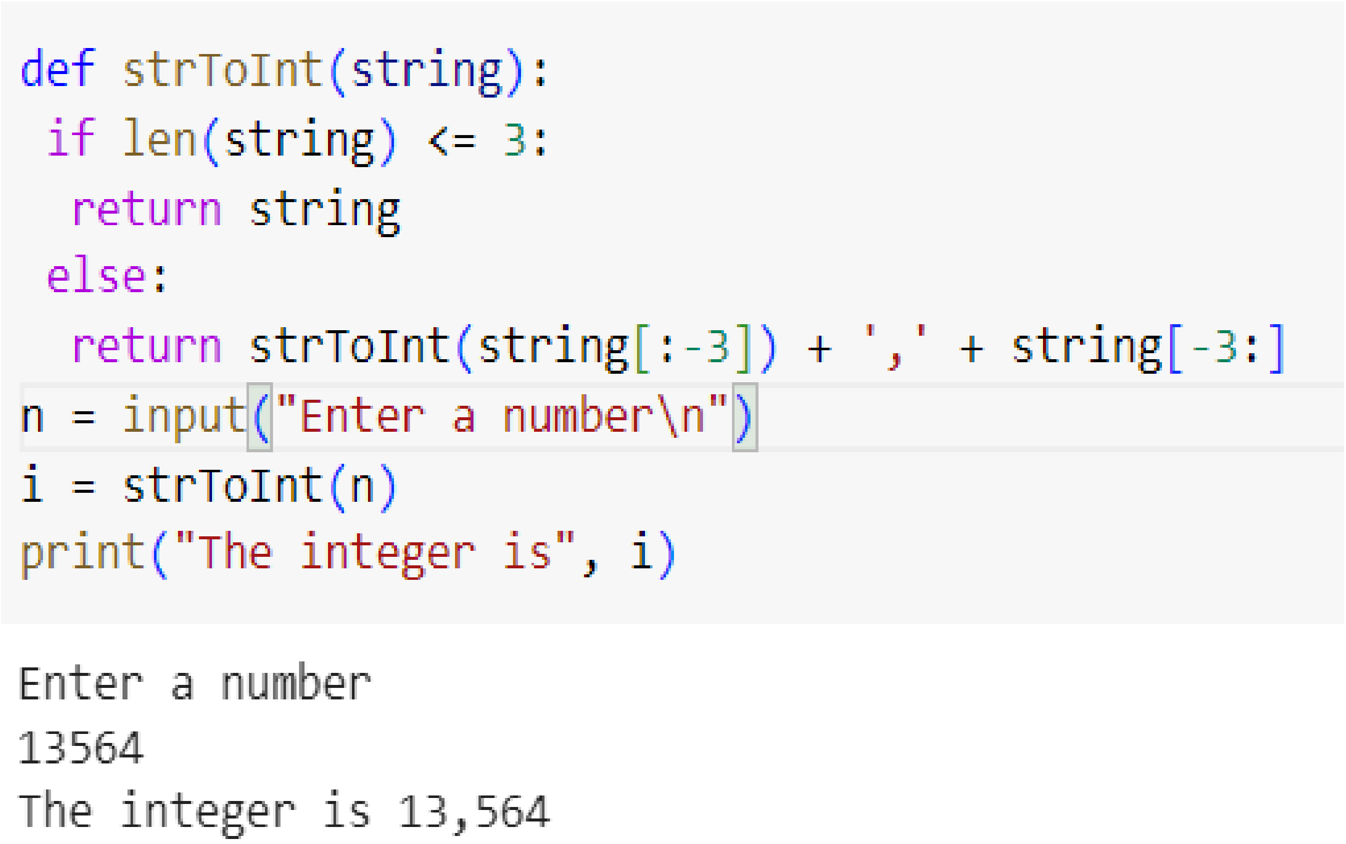


```
Enter the search key
9
```

3. Write a recursive function to convert the entered string of digits into the integer it represents. For example, 13531 represents the integer 13,531.

```
def strToInt(string):
 if len(string) <= 3:
  return string
 else:
  return strToInt(string[:-3]) + ',' + string[-3:]
n = input("Enter a number\n")
i = strToInt(n)
print("The integer is", i)
```

OUTPUT:



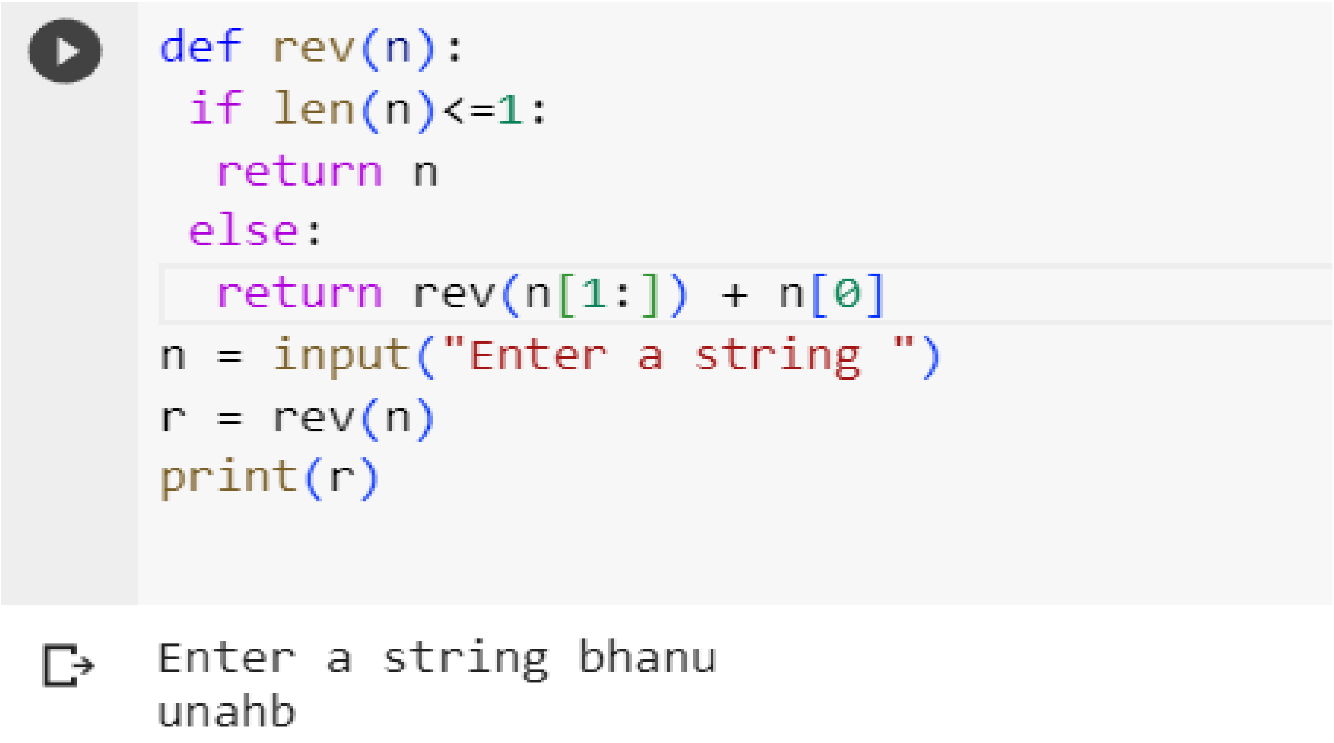
```
def strToInt(string):
 if len(string) <= 3:
  return string
 else:
  return strToInt(string[:-3]) + ',' + string[-3:]
n = input("Enter a number\n")
i = strToInt(n)
print("The integer is", i)
```

```
Enter a number
13564
The integer is 13,564
```

4. Write a short recursive Python function that takes a character string s and outputs its reverse. For example, the reverse of pots&pans would be snap&stop.

```
def rev(n):
 if len(n)<=1:
  return n
 else:
  return rev(n[1:]) + n[0]
n = input("Enter a string ")
r = rev(n)
print(r)
```

OUTPUT:



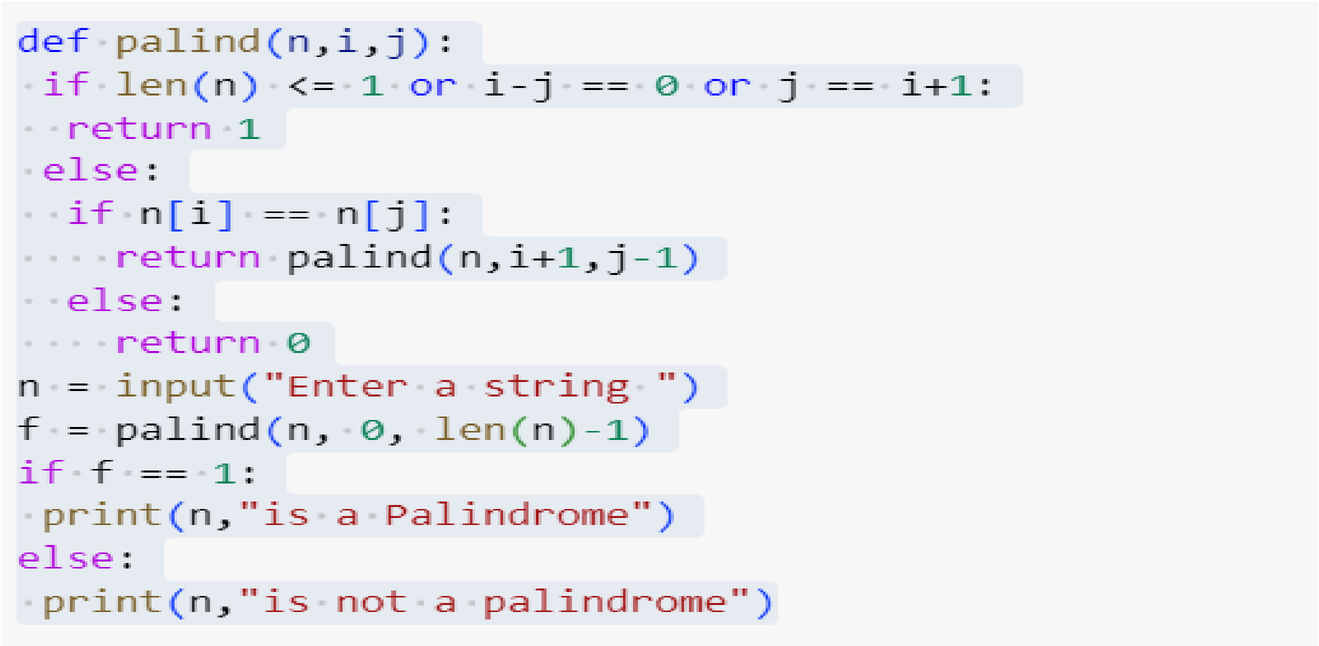
```
def rev(n):
 if len(n)<=1:
  return n
 else:
  return rev(n[1:]) + n[0]
n = input("Enter a string ")
r = rev(n)
print(r)
```

```
Enter a string bhanu
unahb
```

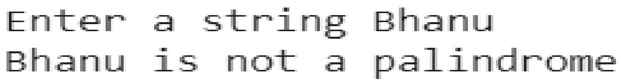
5. Write a short recursive Python function that determines if a string s is a palindrome. For example, racecar and gohangasalamiimalasagnahog are palindromes.

```
def palind(n,i,j):
 if len(n) <= 1 or i-j == 0 or j == i+1:
  return 1
 else:
  if n[i] == n[j]:
    return palind(n,i+1,j-1)
  else:
    return 0
n = input("Enter a string ")
f = palind(n, 0, len(n)-1)
if f == 1:
 print(n,"is a Palindrome")
else:
 print(n,"is not a palindrome")
```

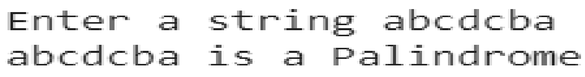
OUTPUT:



```
def palind(n,i,j):
 if len(n) <= 1 or i-j == 0 or j == i+1:
  return 1
 else:
  if n[i] == n[j]:
    return palind(n,i+1,j-1)
  else:
    return 0
n = input("Enter a string ")
f = palind(n, 0, len(n)-1)
if f == 1:
 print(n,"is a Palindrome")
else:
 print(n,"is not a palindrome")
```



```
Enter a string Bhanu
Bhanu is not a palindrome
```



```
Enter a string abcdcba
abcdcba is a Palindrome
```