PYTHON PROGRAMS

1. 1)  [Python Program to find whether the given number is palindrome or not?](https://sathyabama.cognibot.in/mod/quiz/view.php?id=40408)

n=int(input(""))

temp=n

rev=0

while(n>0):

dig=n%10

rev=rev\*10+dig

n=n//10

print(rev)

1. 2)  [Python program to reverse a number using recursive function.](https://sathyabama.cognibot.in/mod/quiz/view.php?id=40814)

n=int(input())

temp=n

rev=0

while(n>0):

dig=n%10

rev=rev\*10+dig

n=n//10

if(temp==rev):

print(rev)

else:

print(rev)

1. 3)  [python Program to convert lowercase string to uppercase string](https://sathyabama.cognibot.in/mod/quiz/view.php?id=40966)

st = 'sathyabama'

out = ''

for n in st:

if n not in 'abcdefghijklmnopqrstuvwqxyz':

out = out + n

else:

k = ord(n)

l = k - 32

out = out + chr(l)

print(out)

1. 4)  [Python Program to print the Fibonacci Series.](https://sathyabama.cognibot.in/mod/quiz/view.php?id=41070)

n = int(input(""))

a = 0

b = 1

sum = 0

count = 1

print(end = "")

while(count <= n):

print(sum, end = " ")

count += 1

a = b

b = sum

sum = a + b

1. 5)  [Python program to sort the elements of list in ascending order](https://sathyabama.cognibot.in/mod/quiz/view.php?id=41341)

arr = [int(x) for x in input().split()]

temp = 0;

#Sort the array in ascending order

for i in range(0, len(arr)):

for j in range(i+1, len(arr)):

if(arr[i] > arr[j]):

temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

for i in range(0, len(arr)):

print(arr[i], end=" ");

6) [Python Program to find the product of two matrices.](https://sathyabama.cognibot.in/mod/quiz/view.php?id=41491)

X = [[12,7,3],

[4 ,5,6],

[7 ,8,9]]

# 3x4 matrix

Y = [[5,8,1,2],

[6,7,3,0],

[4,5,9,1]]

# result is 3x4

result = [[0,0,0,0],

[0,0,0,0],

[0,0,0,0]]

# iterate through rows of X

for i in range(len(X)):

# iterate through columns of Y

for j in range(len(Y[0])):

# iterate through rows of Y

for k in range(len(Y)):

result[i][j] += X[i][k] \* Y[k][j]

for r in result:

print(r)

1. 7)  [Python program to check whether an alphabet is vowel or consonant.](https://sathyabama.cognibot.in/mod/quiz/view.php?id=41810)

str = input("")

if str in ('a', 'e', 'i', 'o', 'u', 'A', 'E', 'I', 'O', 'U', 'A', 'E'):

print("Consonant")

print("Vowel")

else:

print("Consonant")

print("Vowel")

1. 8)  [Python Program to Check Prime Number.](https://sathyabama.cognibot.in/mod/quiz/view.php?id=41820)

A = 11

if A > 1:

for i in range(2, A//2):

if (A % i) == 0:

print(A, "is not a prime number")

else:

print(A, "is a prime number")

else:

print(A, "is not a prime number")

1. 9)  [Python Program to Check If a number is Prime or Not.](https://sathyabama.cognibot.in/mod/quiz/view.php?id=41830)

num = 2

# If given number is greater than 1

if num > 1:

# Iterate from 2 to n / 2

for i in range(2, int(num/2)+1):

# If num is divisible by any number between

# 2 and n / 2, it is not prime

if (num % i) == 0:

print(num, "is not a Prime number")

break

else:

print(num, "is a Prime number")

else:

print(num, "is not a Prime number")

1. 10)  [Python program to enter basic salary and calculate gross salary of an employee algorithm.](https://sathyabama.cognibot.in/mod/quiz/view.php?id=42856)

Basic\_Salary = 20000

DA = (Basic\_Salary \* 40) / 100

HRA = (Basic\_Salary \* 20) / 100

Gross\_Salary = Basic\_Salary + DA + HRA

print("Dearness Allowance 40 % of Basic Salary :", DA)

print("House Rent 20 % of Basic Salary :" , HRA)

print("Gross Salary :", Gross\_Salary)

1. 11)  [Python Program to Create Pyramid Patterns.](https://sathyabama.cognibot.in/mod/quiz/view.php?id=42893)

rows = 5

k = 0

for i in range(1, rows+1):

for space in range(1, (rows-i)+1):

print(end=" ")

while k!=(2\*i-1):

print("\* ", end="")

k += 1

k = 0

print()

1. 12)  [Python Program for Selection Sort.](https://sathyabama.cognibot.in/mod/quiz/view.php?id=42917)

def selectionSort(array, size):

for step in range(size):

min\_idx = step

for i in range(step + 1, size):

if array[i] < array[min\_idx]:

min\_idx = i

(array[step], array[min\_idx]) = (array[min\_idx], array[step])

data = [21, 6, 9, 33, 3]

size = len(data)

selectionSort(data, size)

print(data)

1. 13)  [Implement Built In Function and Libraries using python.](https://sathyabama.cognibot.in/mod/quiz/view.php?id=43029)

A)

def tri\_recursion(k):

if(k>0):

result = k+tri\_recursion(k-1)

print(result)

else:

result = 0

return result

print("Recursion Example Results")

tri\_recursion(6)

B)

list1 = ["apple", "banana", "cherry", "apple", "cherry"]

print(list1)

list2 = ["apple", "banana", "cherry"]

print(len(list2))

list3 = ["abc", 34, True, 40, "male"]

print(list3)

list4 = list(("apple", "banana", "cherry"))

print(list4)

1. 14)  [Python program to Range of Indexes,list use the in keyword,Range of Item Values,insert() ,append(),extend(),remove(),pop() ,del keyword in List built in function.](https://sathyabama.cognibot.in/mod/quiz/view.php?id=43050)

list1 = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]

print(list1[2:][:3])

print(list1[3:][:3])

list2= ["apple", "banana", "cherry"]

if "apple" in list2:

print("Yes, 'apple' is in the fruits list")

list2.insert(2, 'watermelon')

print(list2)

list3= ["apple", "banana", "cherry"]

list3.insert(3, 'orange')

print(list3)

list4= ["apple", "banana", "cherry"]

list4.insert(1, 'orange')

print(list4)

list5= ["apple", "banana", "cherry"]

more\_list= ['mango', 'pineapple', 'papaya']

list5.extend(more\_list)

print(list5)

list6= ["apple", "banana", "cherry"]

list6.remove('banana')

print(list6)

list7= ["apple", "banana", "cherry"]

list7.remove('apple')

print(list7)

list7.clear()

print(list7)

1. 15)  [Python program to sort() , reverse() ,copy(),list(),count(),index()method, keyword reverse = True, key = function,+ operator in list Library Function.](https://sathyabama.cognibot.in/mod/quiz/view.php?id=43054)

l1=["orange", "mango", "kiwi", "pineapple", "banana"]

l1.sort()

print(l1)

l2=["banana", "Orange", "Kiwi", "cherry"]

l2.reverse()

print(l2)

l3=["apple", "banana", "cherry"]

l3.copy()

print(l3)

print(l3)

l4=['1', '3', '7', '8', '7', '5', '4', '6', '8', '5']

c = l4.count('8')

print(c)

c=l4.index('8')

print(c)

l5=["orange", "mango", "kiwi", "pineapple", "banana"]

print(sorted(l5, reverse=True))

def myfunc(n):

return abs(n - 50)

l6 = [100, 50, 65, 82, 23]

l6.sort(key = myfunc)

print(l6)

tuple1 = ("a", "b" , "c")

tuple2 = (1, 2, 3)

t=tuple1+tuple2

print(t)

list1=("apple", "banana", "cherry")

l=list1\*2

print(l)

1. 16)  [Implement Optimizing loop using python.](https://sathyabama.cognibot.in/mod/quiz/view.php?id=44184)

# slower

x = 2

y = 5

temp = x

x = y

y = temp

print (x,y)

x,y = 3,5

# faster

x, y = y, x

print (x,y)

1. 17)  [Python program to find the gcd of two numbers](https://sathyabama.cognibot.in/mod/quiz/view.php?id=46838)

def hcf(a, b):

if(b == 0):

return a

else:

return hcf(b, a % b)

a = 60

b = 48

# prints 12

print("The gcd of 60 and 48 is : ", end="")

print(hcf(60, 48))

1. 18)  [Python Program to Find Sum of Natural Numbers Using Recursion](https://sathyabama.cognibot.in/mod/quiz/view.php?id=46839)

def sum(n):

if n == 0:

return 0

else:

return n + sum(n-1)

arr=[x for x in input().split()]

num = int(arr[len(arr)-1])

if num >= 0:

print("The sum is ",sum(num))

else:

print("Natural number begins from 1")

1. 19)  [Python program to convert a binary number to decimal number.](https://sathyabama.cognibot.in/mod/quiz/view.php?id=46843)

b\_num = list(input(""))

value = 0

for i in range(len(b\_num)):

digit = b\_num.pop()

if digit == '1':

value = value + pow(2, i)

print("Equivalent Decimal Value =",value)

1. 20)  [Factorial program with memoization using decorators.](https://sathyabama.cognibot.in/mod/quiz/view.php?id=47003)

def memoize\_factorial(f):

memory = {}

def inner(num):

if num not in memory:

memory[num] = f(num)

return memory[num]

return inner

@memoize\_factorial

def facto(num):

if num == 1:

return 1

else:

return num \* facto(num-1)

print(facto(5))

1. 21)  [Python Program for factorial of a number](https://sathyabama.cognibot.in/mod/quiz/view.php?id=47008)

num = 5

factorial = 1

if num < 0:

print("Sorry, factorial does not exist for negative numbers")

elif num == 0:

print("The factorial of 0 is 1")

else:

for i in range(1,num + 1):

factorial = factorial\*i

print("Factorial of",num,"is",factorial)

1. 22)  [Python program for implementation of MergeSort(Implement Divide and conquer)](https://sathyabama.cognibot.in/mod/quiz/view.php?id=47011)

def mergeSort(arr):

if len(arr) > 1:

mid = len(arr)//2

L = arr[:mid]

R = arr[mid:]

mergeSort(L)

mergeSort(R)

i = j = k = 0

while i < len(L) and j < len(R):

if L[i] < R[j]:

arr[k] = L[i]

i += 1

else:

arr[k] = R[j]

j += 1

k += 1

while i < len(L):

arr[k] = L[i]

i += 1

k += 1

while j < len(R):

arr[k] = R[j]

j += 1

k += 1

def printList(arr):

for i in range(len(arr)):

print(arr[i], end=" ")

print()

if \_\_name\_\_ == '\_\_main\_\_':

arr = [12, 11, 13, 5, 6, 7]

print("Given array is", end="\n")

printList(arr)

mergeSort(arr)

print("Sorted array is: ", end="\n")

printList(arr)

1. 23) [Python3 program to perform basic timSort](https://sathyabama.cognibot.in/mod/quiz/view.php?id=47014)

MINIMUM= 32

def find\_minrun(n):

r = 0

while n >= MINIMUM:

r |= n & 1

n >>= 1

return n + r

def insertion\_sort(array, left, right):

for i in range(left+1,right+1):

element = array[i]

j = i-1

while element<array[j] and j>=left :

array[j+1] = array[j]

j -= 1

array[j+1] = element

return array

def merge(array, l, m, r):

array\_length1= m - l + 1

array\_length2 = r - m

left = []

right = []

for i in range(0, array\_length1):

left.append(array[l + i])

for i in range(0, array\_length2):

right.append(array[m + 1 + i])

i=0

j=0

k=l

while j < array\_length2 and i < array\_length1:

if left[i] <= right[j]:

array[k] = left[i]

i += 1

else:

array[k] = right[j]

j += 1

k += 1

while i < array\_length1:

array[k] = left[i]

k += 1

i += 1

while j < array\_length2:

array[k] = right[j]

k += 1

j += 1

def tim\_sort(array):

n = len(array)

minrun = find\_minrun(n)

for start in range(0, n, minrun):

end = min(start + minrun - 1, n - 1)

insertion\_sort(array, start, end)

size = minrun

while size < n:

for left in range(0, n, 2 \* size):

mid = min(n - 1, left + size - 1)

right = min((left + 2 \* size - 1), (n - 1))

merge(array, left, mid, right)

size = 2 \* size

array = [-2, 7, 15, -14, 0, 15, 0, 7, -7, -4, -13, 5, 8, -14, 12]

print("Given Array is")

print(array)

tim\_sort(array)

print("After Sorting Array is")

print(array)

1. 24)  [Python program to insert N elements in a Linked List one after other at middle position](https://sathyabama.cognibot.in/mod/quiz/view.php?id=47018)

class Node:

def \_\_init\_\_(self):

self.value = 0

self.next = None

class LinkedList:

def \_\_init\_\_(self) :

self.head = None

self.mid = None

self.count = 0

def insertAtMiddle(self , n):

temp = Node()

temp1 = None

temp.next = None

temp.value = n

if (self.count < 2):

if (self.head == None) :

self.head = temp

else:

temp1 = self.head

temp1.next = temp

self.count = self.count + 1

self.mid = self.head

else:

temp.next = self.mid.next

self.mid.next = temp

self.count = self.count + 1

if (self.count % 2 != 0):

self.mid = self.mid.next

def show(self):

temp = None

temp = self.head

while (temp != None) :

print( temp.value, end = " -> ")

temp = temp.next

print( "NULL")

arr = [ 1, 2, 3, 4, 5]

n = len(arr)

L1 = LinkedList()

for i in range(n):

L1.insertAtMiddle(arr[i])

L1.show()

1. 25)  [Monitoring memory usage of a running Python program](https://sathyabama.cognibot.in/mod/quiz/view.php?id=47019)

# importing the module

import tracemalloc

# code or function for which memory

# has to be monitored

def app():

lt = []

for i in range(0, 100000):

lt.append(i)

# starting the monitoring

tracemalloc.start()

# function call

app()

# displaying the memory

print(tracemalloc.get\_traced\_memory())

# stopping the library

tracemalloc.stop()

1. 26)  [Python Program for Linear Search](https://sathyabama.cognibot.in/mod/quiz/view.php?id=47862)
2. 27)  [Python Program to Display Powers of 2 Using Anonymous Function](https://sathyabama.cognibot.in/mod/quiz/view.php?id=48117)

terms = 10

result = list(map(lambda x: 2 \*\* x, range(terms)))

for i in range(terms):

print("2 raised to power",i,"is",result[i])

1. 28)  [Python program for implementation of heap Sort.](https://sathyabama.cognibot.in/mod/quiz/view.php?id=48685)

def heapify(arr, n, i):

largest = i # Initialize largest as root

l = 2 \* i + 1 # left = 2\*i + 1

r = 2 \* i + 2 # right = 2\*i + 2

if l < n and arr[i] < arr[l]:

largest = l

if r < n and arr[largest] < arr[r]:

largest = r

if largest != i:

arr[i],arr[largest] = arr[largest],arr[i] # swap

heapify(arr, n, largest)

def heapSort(arr):

n = len(arr)

for i in range(n // 2 - 1, -1, -1):

heapify(arr, n, i)

for i in range(n-1, 0, -1):

arr[i], arr[0] = arr[0], arr[i] # swap

heapify(arr, i, 0)

arr = [ 12, 11, 13, 5, 6, 7]

heapSort(arr)

n = len(arr)

print ("Sorted array is")

for i in range(n):

print ("%d" %arr[i]),

1. 29) [Python Program for ShellSort.](https://sathyabama.cognibot.in/mod/quiz/view.php?id=48687)

def shellSort(array, a):

gap = a // 2

while gap > 0:

for i in range(gap, a):

temp = array[i]

j = i

while j >= gap and array[j - gap] > temp:

array[j] = array[j - gap]

j -= gap

array[j] = temp

gap //= 2

array = [9, 1, 8, 7, 3, 6, 4, 5]

size = len(array)

shellSort(array, size)

print('Sorted Array in Ascending Order:')

print(array)

1. 30)  [Python Program for Bubble Sort](https://sathyabama.cognibot.in/mod/quiz/view.php?id=48688)

def bubbleSort(array):

for a in range(len(array)):

for b in range(0, len(array) - a - 1):

if array[b] > array[b + 1]:

temp = array[b]

array[b] = array[b+1]

array[b+1] = temp

array = [-2, 45, 0, 11, -9]

bubbleSort(array)

print('Sorted Array in Ascending Order: ')

print(array)

1. 31)  [Python Program for Merge Sort.](https://sathyabama.cognibot.in/mod/quiz/view.php?id=48690)

def Merge\_Sort(array):

if len(array) > 1:

mid = len(array)//2

Left = array[:mid]

Right = array[mid:]

Merge\_Sort(Left)

Merge\_Sort(Right)

i = j = k = 0

while i < len(Left) and j < len(Right):

if Left[i] < Right[j]:

array[k] = Left[i]

i += 1

else:

array[k] = Right[j]

j += 1

k += 1

while i < len(Left):

array[k] = Left[i]

i += 1

k += 1

while j < len(Right):

array[k] = Right[j]

j += 1

k += 1

def printarray(array):

for i in range(len(array)):

print(array[i], end=" ")

print()

if \_\_name\_\_ == '\_\_main\_\_':

array = [12, 11, 13, 5, 6, 7]

Merge\_Sort(array)

print("Sorted array is: ")

printarray(array)

1. 32)  [Python Program for Radix Sort](https://sathyabama.cognibot.in/mod/quiz/view.php?id=48696)

def countingSort(arr, exp1):

n = len(arr)

# The output array elements that will have sorted arr

output = [0] \* (n)

# initialize count array as 0

count = [0] \* (10)

# Store count of occurrences in count[]

for i in range(0, n):

index = (arr[i]/exp1)

count[int((index)%10)] += 1

# Change count[i] so that count[i] now contains actual

# position of this digit in output array

for i in range(1,10):

count[i] += count[i-1]

# Build the output array

i = n-1

while i>=0:

index = (arr[i]/exp1)

output[ count[ int((index)%10) ] - 1] = arr[i]

count[int((index)%10)] -= 1

i -= 1

# Copying the output array to arr[],

# so that arr now contains sorted numbers

i = 0

for i in range(0,len(arr)):

arr[i] = output[i]

# Method to do Radix Sort

def radixSort(arr):

# Find the maximum number to know number of digits

max1 = max(arr)

exp = 1

while max1/exp > 0:

countingSort(arr,exp)

exp \*= 10

arr = [ 170, 45, 75, 90, 802, 24, 2, 66]

radixSort(arr)

for i in range(len(arr)):

print(arr[i]),