**PYTHON PROGRAM 1:**

**Python Program to find the length of the list using Recursion.**

def length(a):

if len(a)==0:

return 0

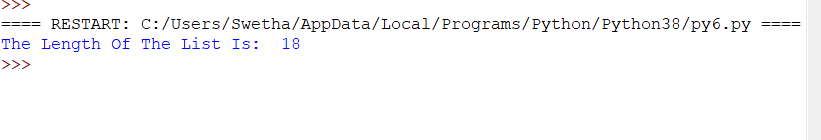
else:

return 1+length(a[1:])

a = [1,7,8,9,2,3,5,6,0,12,5,6,7,6,9,0,1,2]

print("The Length Of The List Is: ", length(a))

**OUTPUT:**



**C PROGRAM 2:**

**write a c program to print the sum of boundary elements of a matrix.**

#include<stdio.h>

#include<limits.h>

int main()

{

int m, n, sum = 0;

printf(“\nEnter the order of the matrix : “);

scanf(“%d %d”,&m,&n);

int i, j;

int mat[m][n];

printf(“\nInput the matrix elements\n”);

for(i = 0; i < m; i++)

{

for(j = 0; j < n; j++)

scanf(“%d”,&mat[i][j]);

}

printf(“\nBoundary Matrix\n”);

for(i = 0; i < m; i++)

{

for(j = 0; j < n; j++)

{

if (i == 0 || j == 0 || i == n – 1 || j == n – 1)

{

printf(“%d “, mat[i][j]);

sum = sum + mat[i][j];

}

else

printf(” “);

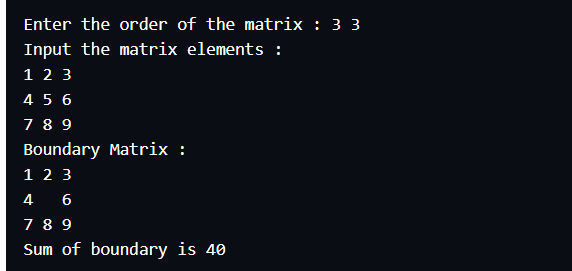
}

printf(“\n”);

}

printf(“\nSum of boundary is %d”, sum);

**OUTPUT:**



**JAVA PROGRAM 3:**

**Write a java program to find the maximum and minimum value node from a circular linked list.**

public class MinMax {

public class Node{

int data;

Node next;

public Node(int data) {

this.data = data;

}

}

public Node head = null;

public Node tail = null;

public void add(int data){

Node newNode = new Node(data);

if(head == null) {

head = newNode;

tail = newNode;

newNode.next = head;

}

else {

tail.next = newNode;

//New node will become new tail.

tail = newNode;

tail.next = head;

}

}

public void minNode() {

Node current = head;

int min = head.data;

if(head == null) {

System.out.println("List is empty");

}

else {

do{

if(min > current.data) {

min = current.data;

}

current= current.next;

}while(current != head);

System.out.println("Minimum value node in the list: "+ min);

}

}

public void maxNode() {

Node current = head;

int max = head.data;

if(head == null) {

System.out.println("List is empty");

}

else {

do{

if(max < current.data) {

max = current.data;

}

current= current.next;

}while(current != head);

System.out.println("Maximum value node in the list: "+ max);

}

}

public static void main(String[] args) {

MinMax cl = new MinMax();

cl.add(20);

cl.add(30);

cl.add(40);

cl.add(50);

cl.minNode();

cl.maxNode();

}

}

**OUTPUT:**

