**GDSC PROGECT**

**Answer 1:**

*string=input("Enter string:")*

*def is\_valid\_parentheses(s: str) -> bool:*

*i=[]*

*parentheses={')': '(', '}': '{', ']': '['}*

*for char in s:*

*if char in parentheses.values():*

*i.append(char)*

*elif char in parentheses.keys():*

*if not i:*

*return False*

*return not i*

*output=is\_valid\_parentheses(string)*

*print(f"String:{string}\nOutput:{output}")*

**Answer 2:**

*def matrix\_function(order):*

*matrix=[]*

*print("Enter the Matrix:")*

*for i in range(order):*

*row=list(map(int,input().split()))*

*matrix.append(row)*

*for i in range(1,order-1):*

*matrix[i][1:-1]=sorted(matrix[i][1:-1])*

*print("Sorted Matrix:")*

*sum\_of\_diagonal=sum(matrix[i][i] for i in range(order))*

*for row in matrix:*

*print(" ".join(map(str,row)))*

*print("Sum of Diagonals:",sum\_of\_diagonal)*

*order=int(input("Enter the Order:"))*

*matrix\_function(order)*

**Answer 3:**

*def palindrome(word):*

*return word==word[::-1]*

*def convert\_to\_palindromes(sentence):*

*words=sentence.split()*

*word1=[]*

*for word in words:*

*if not palindrome(word):*

*word+=word[::-1]*

*word1.append(word)*

*return ' '.join(word1)*

*user\_sentence=input("Input a sentence:")*

*output\_sentence=convert\_to\_palindromes(user\_sentence)*

*print("Output:")*

*print(output\_sentence)*

**Answer 4:**

*def maximum\_sum\_of\_contiguous\_subarray(nums):*

*if not nums:*

*return 0*

*maximum\_sum=nums[0]*

*sum1=nums[0]*

*for num in nums[1:]:*

*sum1=max(num,sum1+num)*

*maximum\_sum=max(maximum\_sum,sum1)*

*return maximum\_sum*

*nums=[-2, 1, -3, 4, -1, 2, 1, -5, 4]*

*result=maximum\_sum\_of\_contiguous\_subarray(nums)*

*print("Output:", result)*