# Assignment 4

## Student Name: Anuj Kumar UID:22BCS11581

## Branch: BE-CSE (General) Section/Group: FL\_IOT-602 A

## Semester:6th Date of Performance: 21-02-25

## Subject Name: Advanced Programming Lab-2 Subject Code: 22CSP-351

# Aim: 1763. Longest Nice Substring

# Implementation/ Code:

# class Solution {

# public String longestNiceSubstring(String s) {

# int n = s.length();

# String result = "";

# for (int i = 0; i < n; i++) {

# for (int j = i; j < n; j++) {

# String sub = s.substring(i, j + 1);

# if (isNice(sub) && sub.length() > result.length()) {

# result = sub;

# }

# }

# }

# return result;

# }

# private boolean isNice(String s) {

# for (char c : s.toCharArray()) {

# if (Character.isLowerCase(c) && s.indexOf(Character.toUpperCase(c)) == -1) {

# return false;

# }

# if (Character.isUpperCase(c) && s.indexOf(Character.toLowerCase(c)) == -1) {

# return false;

# }

# }

# return true;

# }

# }

# Output:

# 

# Aim: 190. Reverse Bits

# Implementation/ Code:

# class Solution {

# public:

# uint32\_t reverseBits(uint32\_t n) {

# uint32\_t res=0;

# for(int i=0; i<32; i++){

# int last=n&1;

# int mask=last<<(31-i);

# res= res|mask;

# n=n>>1;

# }

# return res;

# }

# };

# Output:

# 

# Aim: 191. Number of 1 Bits

# Implementation/ Code:

# class Solution {

# public:

# int hammingWeight(int n) {

# int count=0;

# while(n){

# count+=(n&1);

# n>>=1;

# }

# return count;

# }

# };

# Output:

# 

# Aim: 53. Maximum Subarray

# Implementation/ Code:

# class Solution {

# public:

# int maxSubArray(vector<int>& nums) {

# int maxSum = INT\_MIN;

# int currentSum = 0;

# for (int num : nums) {

# currentSum += num;

# maxSum = max(maxSum, currentSum);

# if (currentSum < 0) currentSum = 0;

# }

# return maxSum;

# }

# };

# Output:

# 

# Aim: 240. Search a 2D Matrix II

# Implementation/ Code:

# class Solution {

# public:

# bool searchMatrix(vector<vector<int>>& matrix, int target) {

# int row = 0, col = matrix[0].size() - 1;

# while (row < matrix.size() && col >= 0) {

# if (matrix[row][col] == target) return true;

# matrix[row][col] > target ? col-- : row++;

# }

# return false;

# }

# };

# Output:

# 

# Aim: 372. Super Pow

# Implementation/ Code:

# class Solution {

# public:

# int modPow(int a, int b, int mod) {

# int res = 1;

# a %= mod;

# while (b) {

# if (b % 2) res = (1LL \* res \* a) % mod;

# a = (1LL \* a \* a) % mod;

# b /= 2;

# }

# return res;

# }

# 

# int superPow(int a, vector<int>& b) {

# int mod = 1337, res = 1;

# for (int d : b) res = modPow(res, 10, mod) \* modPow(a, d, mod) % mod;

# return res;

# }

# };

# Output:

# 