OOPS ST2 Exam Codes.

Split Array: Nikita and The Game

```
//
// Created by HARSHPREET SINGH on 31.10.2023.
//
#include <bits/stdc++.h>
using namespace std;

int split(vector<int>& arr, int start, int end){
    // base case
    if(start >= end) return 0;

    // recursive case
    for(int i=start;i<=end;i++){
        int left = 0;
        for(int j = start; j<=i ; j++){
            left += arr[j];
        }
        int right = 0;
        for(int j = i+1; j<=end; j++){
            right += arr[j];
        }

        if(left == right){
            return 1 + max(split(arr,start,i), split(arr,i+1,end));
        }
    }
    return 0;
}


int main() {
    int test_cases;
    cin >> test_cases;
    while (test_cases--) {
        int n;
        cin >> n;
        vector<int> arr(n);
        for (int i = 0; i < n; i++) {
            cin >> arr[i];
        }
        int result = split(arr,0,n-1);
        cout << result << endl;
    }
    return 0;
}
```

Stairs Jump

```
//
// Created by HARSHPREET SINGH on 26.09.2023.
//
#include <iostream>
using namespace std;


int ways(int n,int m){
    if(n==0) return 1;
    if(n<0) return 0;

    int way = 0;
    for(int i=1;i<=m;i++){
        way += ways(n-i,m);
    }
    return way;
}

int main(){
    int n = 4; int m = 3;
    cout << ways(n,m);
}
```

Stairs Jump Pt.2 Only 1,2,3 stairs.

```
//
// Created by HARSHPREET SINGH on 31.10.2023.
```

```
//
#include <bits/stdc++.h>
using namespace std;

int countWays(int n) {
    if (n == 0 || n == 1) {
        return 1;
    }
    if (n == 2) {
        return 2;
    }

    return countWays(n-1) + countWays(n-2) + countWays(n-3);
}

int main() {
    int n;
    cout << "Enter the number of stairs: ";
    cin >> n;

    int ways = countWays(n);
    cout << "Number of ways to reach the top: " << ways << endl;

    return 0;
}
```

Integer into Words

```
#include <bits/stdc++.h>
using namespace std;

static string name[] = {"zero","one","two","three","four","five","six","seven","eight","nine"};

void toPrint(int n){
    if(n==0) return;
    else{
        toPrint(n/10);
        cout << name[n%10] << " ";
    }
}

int main(){
    int n; cin >> n;
    toPrint(n);
}
```

Is it possible that the Sum of Array of integers is divisible by 3?

```
#include <bits/stdc++.h>
using namespace std;

int isPossibleToConstructDivisibleBy3(vector<int>& arr) {
    int sum = 0;
    for (int num : arr) {
        sum += num;
    }

    if (sum % 3 == 0) {
        return 1;
    } else {
        return 0;
    }
}

int main() {
    vector<int> arr = {19, 4};

    int result = isPossibleToConstructDivisibleBy3(arr);

    cout << result << endl;

    return 0;
}
```

Matrix Multiplication

```
//
// Created by HARSHPREET SINGH on 31.10.2023.
//
#include <bits/stdc++.h>
const int MAX = 100;
using namespace std;

void mm(int fM[MAX][MAX], int sM[MAX][MAX], int result[MAX][MAX], int r1, int c1, int r2, int c2){
    for(int i=0;i<r1;i++){
        for(int j=0;j<c2;j++){
            result[i][j] = 0;
            for(int k=0;k<c1;k++){
                result[i][j] += fM[i][k] * sM[k][j];
            }
        }
    }
}

int main(){
    int firstM[MAX][MAX], secondM[MAX][MAX], result[MAX][MAX],r1,c1,r2,c2;
    cin >> r1 >> c1;
    for(int i=0;i<r1;i++){
        for(int j=0;j<c1;j++){
            cin >> firstM[i][j];
        }
    }

    cin >> r2 >> c2;
    for(int i=0;i<r2;i++){
        for(int j=0;j<c2;j++){
            cin >> secondM[i][j];
        }
    }

    if(r2 != c1){
        cout << "ERROR";
    }
    else{
        mm(firstM,secondM,result,r1,c1,r2,c2);

        cout << "OUTPUT" << endl;
        for(int i=0;i<r1;i++) {
            for (int j = 0; j < c2; j++) {
                cout << result[i][j] << " ";
            }
            cout << endl;
        }
    }

    return 0;

}
```

Reverse Array K times.

```
//
// Created by HARSHPREET SINGH on 31.10.2023.
//
#include <bits/stdc++.h>
using namespace std;

void reverseArray(int arr[], int size){
    int start = 0; int end = size - 1;
    while(start < end){
        swap(arr[start],arr[end]);
        start++;
        end--;
    }
}

void reverseKTimes(int arr[], int size, int k){
    for(int i=0;i<k;i++){
        reverseArray(arr,size);
    }
}

int main(){
    int arr[] = {1,2,3,4,5};
    int k = 2;
    int size = sizeof(arr) / sizeof(arr[0]);

    reverseKTimes(arr,size,k);

    cout << "Reversed array is : ";
    for(int i=0;i<size;i++){
        cout << arr[i] << " ";
    }
    cout << endl;
    return 0;
}
```

No sense to have this as a question lol this is too easy I think sir meant Left or Right Shift the array k times…

So the code for Left Shift and Right Shift K times…

```
//
// Created by HARSHPREET SINGH on 31.10.2023.
//
#include <bits/stdc++.h>
using namespace std;

void leftShift(int arr[], int size, int k){
    k = k % size;
    for(int i=0;i<k;i++){
        int temp = arr[0];
        for(int j=0;j<size-1;j++){
            arr[j] = arr[j+1];
        }
        arr[size - 1] = temp;
    }
}

void rightShift(int arr[], int size, int k){
    k = k % size;
    for(int i=0;i<k;i++){
        int temp = arr[size - 1];
        for(int j=size-1;j>0;j--){
            arr[j] = arr[j-1];
        }
        arr[0] = temp;
    }
}

void printA(int arr[], int size){
    for(int i=0;i<size;i++){
        cout << arr[i] << " ";
    }
    cout << endl;
}

int main(){
    int arr[] = {1,2,3,4,5};
    int k = 2;
    int size = sizeof(arr)/sizeof(arr[0]);

    cout << "Original array : ";
    printA(arr,size);

    cout << "Left Shift Array : ";
    leftShift(arr,size,k);
    printA(arr,size);

    cout << "Right Shift Array : ";
    rightShift(arr,size,k);
    printA(arr,size);
    return 0;
}
```

Lexicographically Order 0 to N

```
//
// Created by HARSHPREET SINGH on 31.10.2023.
//
#include <bits/stdc++.h>
using namespace std;

void lex(int n){
    vector<string> s;

    for(int i=1;i<=n;i++){
        s.push_back(to_string(i));
    }
    sort(s.begin(),s.end());
    vector<int> ans;

    for(int i=0;i<n;i++) ans.push_back(stoi(s[i]));
    for(int i=0;i<n;i++) cout << ans[i] << " ";
}

//recursive way to solve
void helper(int temp, int n, vector<int>& sol){
    if(temp > n) return;
    sol.push_back(temp);
    helper(temp*10, n, sol);
    if(temp%10!=9){
        helper(temp+1,n,sol);
    }
}

void lexRECURSIVE(int n){
    vector<int> sol;
    helper(1,n,sol);
    for(int i=0;i<sol.size();i++){
        cout << sol[i] << " ";
    }
}

int main(){
    int n; cin >> n;
    lex(n);
    cout << endl;
    lexRECURSIVE(n);
    return 0;
}
```

Column with Maximum Sum

```
//
// Created by HARSHPREET SINGH on 31.10.2023.
//
#include <bits/stdc++.h>
using namespace std;

#define N 5 // Number of rows and columns

void colMaxSum(int mat[N][N]) {
    int idx = -1;
    int maxSum = INT_MIN;

    for (int i = 0; i < N; i++) {
        int sum = 0;
        for (int j = 0; j < N; j++) {
            sum += mat[j][i];
        }

        if (sum > maxSum) {
            maxSum = sum;
            idx = i;
        }
    }

    cout << "MAXSUM = " << maxSum << " " << "COlUMN = " << idx+1;
}

int main() {
    int mat[N][N] = {
            { 1, 2, 3, 4, 5 },
            { 5, 3, 1, 4, 2 },
            { 5, 6, 7, 8, 9 },
            { 0, 6, 3, 4, 12 },
            { 9, 7, 12, 4, 3 },
    };

    colMaxSum(mat);

    return 0;
}
```

90 Degrees Matrix Rotation Anti-Clockwise

```
//
// Created by HARSHPREET SINGH on 31.10.2023.
//
#include<bits/stdc++.h>
using namespace std;


void rotate(int matrix[][3]) {
    int n = 3;

    //Creating new matrix to store rotated values
    int temp[n][n];

    int ind = n - 1;
    for (int i = 0; i < n; i++) {

        for (int j = 0; j < n; j++) {
            temp[i][j] = matrix[j][ind];
        }
        ind--;
    }
    //Printing array after rotation
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            cout << temp[i][j] << " ";
        }
        cout << endl;
    }
}
int main() {
    int matrix[][3] = {{1,2,3},{4,5,6},{7,8,9}};

    rotate(matrix);

}
```

Subset Problem

```
#include <bits/stdc++.h>
using namespace std;

static int counter = 0;

void toPrint(vector<int>& arr, int idx, int target, vector<int>& subset) {
    if (target == 0) {
        for (int num : subset) {
            cout << num << " ";
        }
        cout << "  ";
        counter++;
        return;
    }

    if (idx >= arr.size() || target < 0) return;

    subset.push_back(arr[idx]);
    toPrint(arr, idx + 1, target - arr[idx], subset);
    subset.pop_back();
    toPrint(arr, idx + 1, target, subset);
}

int main() {
    int n;
    cin >> n;
    vector<int> arr(n);
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }

    int target;
    cin >> target;

    vector<int> subset;
    toPrint(arr, 0, target, subset);

    cout << endl;
    cout << counter;

    return 0;
}
```

Split Array Problem

```
#include <bits/stdc++.h>
using namespace std;

static int counter = 0;

void split(vector<int>& a, int i, int s1, int s2, vector<int>& g1, vector<int>& g2) {
    if (i == a.size()) {
        if (s1 == s2) {
            for (int n : g1) cout << n << " ";
            cout << "and ";
            for (int n : g2) cout << n << " ";
            cout << endl;
            counter++;
        }
        return;
    }

    g1.push_back(a[i]);
    split(a, i + 1, s1 + a[i], s2, g1, g2);
    g1.pop_back();

    g2.push_back(a[i]);
    split(a, i + 1, s1, s2 + a[i], g1, g2);
    g2.pop_back();
}

int main() {
    int n;
    cin >> n;
    vector<int> a(n);
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }

    vector<int> g1, g2;
    split(a, 0, 0, 0, g1, g2);
    cout << counter << " ";
    return 0;
}
```

First Index

```
#include<iostream>
using namespace std;

int first(int arr[], int n, int target, int i) {
    if (i == n)
        return -1;

    if (arr[i] == target)
        return i;

    return first(arr, n, target, i + 1);
}

int main() {
    int n;
    cin >> n;

    int arr[n];
    for (int i = 0; i < n; i++)
        cin >> arr[i];

    int target;
    cin >> target;

    cout << first(arr, n - 1, target, 0);

    return 0;
}
```

Last Index

```
#include<iostream>
using namespace std;

int last(int arr[], int n, int target) {
    if (n == -1)
```

```
        return -1;

    if (arr[n] == target) {
        return n;
    }

    return last(arr, n - 1, target);
}

int main() {
    int n;
    cin >> n;
    int arr[100000];

    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }

    int target;
    cin >> target;

    cout << last(arr, n, target);

    return 0;
}
```

All Indices

```
#include<iostream>
using namespace std;

void print(int arr[], int i, int n, int target) {
    if (i == n) return;

    if (arr[i] == target)
        cout << i << " ";

    print(arr, i + 1, n, target);
}

int main() {
    int n;
    cin >> n;

    int arr[n];
    for(int i = 0; i < n; i++)
        cin >> arr[i];

    int target;
    cin >> target;

    print(arr, 0, n, target);

    return 0;
}
```

Replace 0’s with 5’s

```
#include<iostream>
using namespace std;

int to5(int n) {
    if (n == 0)
        return 0;

    int c = n % 10;
    if (c == 0)
        c = 5;

    return to5(n / 10) * 10 + c;
}

int main() {
    int n;
    cin >> n;

    if (n == 0)
        return 5;
    else
        cout << to5(n);

    return 0;
}
```