**Batch: C3 Roll No.: 16010123217**

**Experiment No. 0**

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Q.1 Three partitions with equal sum**  
**Code:**  
#include<bits/stdc++.h>

using namespace std;

int countWays(const vector<int> &arr) {

int totalSum = 0;

for (auto num : arr) {

totalSum += num;

}

if (totalSum % 3 != 0) {

return 0;

}

int targetSum = totalSum / 3;

int currentSum = 0;

int countFirstPart = 0;

int countWays = 0;

for (int i = 0; i < arr.size(); ++i) {

currentSum += arr[i];

if (currentSum == targetSum) {

countFirstPart++;

}

if (currentSum == 2 \* targetSum) {

countWays += countFirstPart;

}

}

return countWays;

}

int main() {

vector<int> arr = {1,2,3,0,3};

int result = countWays(arr);

cout << "Number of ways to split into 3 subarrays with equal sum: "

<< result << endl;

return 0;

}  
  
**Output:**  
**1) For arr= {1,2,3,0,3}**



**2) For arr = {1,2,3,-4,4,0,3}**



**3) For arr = {7,7,7,7,7,7}**  


**4) For arr = {1,2,4,0,2,-4,4,1}**

****

**Q.2 Increasing triplet with maximum sum**

**Code:**  
#include <bits/stdc++.h>

using namespace std;

void findMaximumTriplet(int arr[], int n) {

// Initialize variables

int max\_i = -1, max\_j = -1, max\_k = -1;

int max\_sum = 0;

// Iterating over all possible triplets

for (int i = 0; i < n - 2; i++) {

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

for (int k = j + 1; k < n; k++) {

//Applying given condition

if (arr[i] < arr[j] && arr[j] < arr[k]) {

int current\_sum = arr[i] + arr[j] + arr[k];

if (current\_sum > max\_sum) {

max\_sum = current\_sum;

//Storing the index

max\_i = i;

max\_j = j;

max\_k = k;

}

}

}

}

}

if (max\_i != -1 && max\_j != -1 && max\_k != -1) {

cout<< "Triplet: " << arr[max\_i] << ", " << arr[max\_j] << ", " << arr[max\_k] << endl;

cout<<"The maximum sum is: "<<max\_sum<<endl;

} else {

cout<<"No such triplet found.\n";

}

}

int main() {

int arr[] = {1,2,3,0,3};

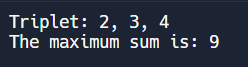
int n = sizeof(arr) / sizeof(arr[0]);

findMaximumTriplet(arr, n);

return 0;

}

**Output:**  
**1) For arr = {1,2,3,0,3}**  


**2) For arr = {1,2,3,-4,4,0,3}**

**3) For arr = {7,7,7,7,7,7}**



**4) For arr = {5,4,3,2,1}**

**5) For arr = {7,1,5,3,6,4}**

