**Kadane’s Algorithm:**

*The idea of****Kadane’s algorithm****is to maintain a variable****max\_ending\_here****that stores the maximum sum contiguous subarray ending at current index and a variable****max\_so\_far****stores the maximum sum of contiguous subarray found so far, Everytime there is a positive-sum value in****max\_ending\_here****compare it with****max\_so\_far****and update****max\_so\_far****if it is greater than****max\_so\_far****.*

[**Pseudocode**](https://www.geeksforgeeks.org/what-is-pseudocode-a-complete-tutorial/)**:**

*Initialize:  
    max\_so\_far = INT\_MIN  
    max\_ending\_here = 0*

*Loop for each element of the array*

*(a) max\_ending\_here = max\_ending\_here + a[i]  
  (b) if(max\_so\_far < max\_ending\_here)  
            max\_so\_far = max\_ending\_here  
  (c) if(max\_ending\_here < 0)  
            max\_ending\_here = 0  
return max\_so\_far*

*CODE*

*// C++ program to print largest contiguous array sum*

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

*using namespace std;*

*int maxSubArraySum(int a[], int size)*

*{*

*int max\_so\_far = INT\_MIN, max\_ending\_here = 0;*

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

*max\_ending\_here = max\_ending\_here + a[i];*

*if (max\_so\_far < max\_ending\_here)*

*max\_so\_far = max\_ending\_here;*

*if (max\_ending\_here < 0)*

*max\_ending\_here = 0;*

*}*

*return max\_so\_far;*

*}*

*// Driver Code*

*int main()*

*{*

*int a[] = { -2, -3, 4, -1, -2, 1, 5, -3 };*

*int n = sizeof(a) / sizeof(a[0]);*

*// Function Call*

*int max\_sum = maxSubArraySum(a, n);*

*cout << "Maximum contiguous sum is " << max\_sum;*

*return 0;*

*}*