**Assignment 3, Question 1**

**Group 24**

**Gorantla Thoyajakshi – 21CS10026**

**Ashwin Prasanth – 21CS30009**

**Algorithm Used:**

* There are 2 options: either the maximum sum is without wrapping (1), or there is some wrapping (2)
  + If it is case 1, then maximum circular sum is just the maximum linear subarray sum
  + If it is case 2, we can find it by subtracting the minimum linear subarray sum from the total sum of array
  + To find the absolute maximum, we find the maximum between these two cases
* So finally, maximum circular subarray sum = Max(max\_linear\_subarray\_sum, total\_sum – min\_linear\_subarray\_sum)
* **Case 1:**
  + We keep two variables – max sum here (mh), max so far (msf) – both initialised with arr[0]
  + We start from index 1, and check whether it is better to consider both elements at index 0 and 1, or just the element at index 1 for max sum
  + This is done by doing: mh = max(mh, mh+arr[i])
  + Then max so far is just obtained as max(msf, mh)
  + This is repeated until we get the max contiguous sum
* **Case 2:**
  + Same thing is repeated, but with min instead of max in all places