**package** Question24;

**import** java.util.HashMap;

**class** LargestSubArray

{

**int** maxLen(**int** arr[], **int** n)

{

HashMap<Integer, Integer> hM = **new** HashMap<Integer, Integer>();

**int** sum = 0;

**int** max\_len = 0;

**int** ending\_index = -1;

**int** start\_index = 0;

**for** (**int** i = 0; i < n; i++)

{

arr[i] = (arr[i] == 0) ? -1 : 1;

}

**for** (**int** i = 0; i < n; i++)

{

sum += arr[i];

**if** (sum == 0)

{

max\_len = i + 1;

ending\_index = i;

}

**if** (hM.containsKey(sum))

{

**if** (max\_len < i - hM.get(sum + n))

{

max\_len = i - hM.get(sum + n);

ending\_index = i;

}

}

**else**

hM.put(sum + n, i);

}

**for** (**int** i = 0; i < n; i++)

{

arr[i] = (arr[i] == -1) ? 0 : 1;

}

**int** end = ending\_index - max\_len + 1;

System.***out***.println(end + " to " + ending\_index);

**return** max\_len;

}

**public** **static** **void** main(String[] args)

{

LargestSubArray sub = **new** LargestSubArray();

**int** arr[] = {1, 0, 0, 1, 0, 1, 1};

**int** n = arr.length;

sub.maxLen(arr, n);

}

}