

Complete the 'balancedSum' function below.

The function is expected to return an integer.

The function accepts INTEGER_ARRAY arr.

```
int balancedSum(int arr_count, int* arr) {  
    long long total=0;  
    for(int i=0;i<arr_count;i++){  
        total+=arr[i];  
    }  
    long long left=0;  
    for(int i=0;i<arr_count;i++){  
        long long right=total-left-arr[i];  
        if(left==right){  
            return i;  
        }  
        left+=arr[i];  
    }  
    return 1;  
}
```

```
#include <stdlib.h>
```

```
int cmp(const void *a, const void *b){  
    return (*(int*)a - *(int*)b);  
}
```

```
int minDiff(int arr_count, int* arr)
```

```
{  
    qsort(arr, arr_count, sizeof(int), cmp);
```

```
    int sum=0;
```

```
    for(int i=0; i<arr_count-1; i++){
```

```
        sum+=abs(arr[i]-arr[i+1]);
```

```
    }
```

```
    return sum;  
}
```

```
/*  
 * Complete the 'arraySum' function below.  
 *  
 * The function is expected to return an INTEGER.  
 * The function accepts INTEGER_ARRAY numbers as parameter.  
 */
```

```
int arraySum(int numbers_count, int *numbers)  
{  
    long long sum=0;  
    for(int i=0;i<numbers_count;i++){  
        sum+=numbers[i];  
    }  
    return sum;  
}
```