```
 *     return a;
 * }
 *
 */
int* reverseArray(int arr_count, int *arr, int *result_count) {
    int*result=(int*)malloc(arr_count*sizeof(int));
    if(result==NULL){
        return NULL;
    }
    for(int i=0;i<arr_count;i++){
        result[i]=arr[arr_count-i-1];
    }
*result_count=arr_count;
return result;
}
```

GOUTHAM
R
ECE B
240801088

	Test	Expected	Got	

✓	int arr[] = {1, 3, 2, 4, 5};
	int result_count;
	int* result = reverseArray(5, arr, &result_count);
	for (int i = 0; i < result_count; i++)
	printf("%d\n", *(result + i));
		5
		4
		2
		3
		1
			5
			4
			2
			3
			1
				✓

Passed all tests! ✓

GOUTHAM
R
ECE B
240801088

```
29 char* cutThemAll(int lengths_count, long *lengths, long minLength) {
30     long t=0,i=1;
31     for(int i=0;i<=lengths_count-1;i++){
32         t+=lengths[i];
33     }
34     do{
35         if(t-lengths[lengths_count-1]<minLength){
36             return "Impossible";
37         }i++;
38     }while(i<lengths_count-i);
39     return "Possible";
40
41 }
42
```

GOUTHAM
R
ECE B
240801088

	Test	Expected	Got	
✓	`long lengths[] = {3, 5, 4, 3};` `printf("%s", cutThemAll(4, lengths, 9))`	`Possible`	`Possible`	✓
✓	`long lengths[] = {5, 6, 2};` `printf("%s", cutThemAll(3, lengths, 12))`	`Impossible`	`Impossible`	✓

Passed all tests! ✓

GOUTHAM
R
ECE B
240801088