

```

32  * }
33  *
34  */
35
36  #include<stdio.h>
37  #include<stdlib.h>
38  int* reverseArray(int arr_count, int *arr, int *result_count) {
39      *result_count=arr_count;
40      int *reversed=(int *)malloc(arr_count * sizeof(int));
41      if(reversed == NULL)
42      {
43          exit(1);
44      }
45      for(int i=0;i<arr_count;i++){
46          reversed[i]=arr[arr_count-1-i];
47      }
48      return reversed;
49  }
50
51  }
52

```

	Test	Expected	Got	
✓	<pre> 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)); </pre>	<pre> 5 4 2 3 1 </pre>	<pre> 5 4 2 3 1 </pre>	✓

Passed all tests! ✓

```

26  * }
27  *
28  */
29  #include<stdio.h>
30  char* cutThemAll(int lengths_count, long *lengths, long minLength) {
31  long totalLength=0;
32
33  for(int i=0;i<lengths_count;i++){
34      totalLength+=lengths[i];
35  }
36  long currentLength=0;
37  for(int i=0;i<lengths_count-1;i++){
38      currentLength+=lengths[i];
39      long remainingLength=totalLength-currentLength;
40      if(remainingLength>=minLength){
41          return "Possible";
42      }
43  }
44  return "Impossible";
45  }
46

```

	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! ✓