

```

23 *      *result_count = 5;
24 *
25 *      int *a = malloc(5 * sizeof(int));
26 *
27 *      for (int i = 0; i < 5; i++) {
28 *          *(a + i) = i + 1;
29 *      }
30 *
31 *      return a;
32 * }
33 *
34 */
35 int* reverseArray(int n, int *a, int *rc) {
36     *rc=n;
37     int *b=(int*)malloc(sizeof(int)*n);
38     for(int i=0;i<n;i++)
39     {
40         b[i]=a[n-i-1];
41     }
42     return b;
43 }
44

```

	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! ✓

```

28  */
29  #include<stdlib.h>
30  int cmp(const void*a,const void*b){
31      return(*(int*)a)-*(int*)b);
32  }
33  char* cutThemAll(int n , long *a, long mL) {
34      int s=0;
35      for(int i=0;i<n;i++)
36      {
37          s+=a[i];
38      }
39      long r=s;
40      qsort(a,n,sizeof(long),cmp);
41      for(int i=0;i<n;i++)
42      {
43          if(r==mL){
44              return "Possible";
45          }
46          if(r>mL){
47              r-=a[i];
48          }
49          else{
50              return "Impossible";
51          }
52      }
53      return "Possible";
54  }
55  }
56

```

	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	✓