```
// ***
// *** You MUST modify this file
// ***

#include <stdio.h>
#include <stdbool.h>
#include <stdlib.h>
#include <string.h>

#ifdef TEST_ELIMINATE
// 100% of the score
void eliminate(int n, int k)
{
  // allocate an arry of n elements
  int * arr = malloc(sizeof(* arr) * n);
  // check whether memory allocation succeeds.
  // if allocation fails, stop
  if (arr == NULL)
    {
      fprintf(stderr, "malloc fail\n");
      return;
    }
  for(int j = 0; j < n; j++)
  {
   arr[j] = j;
  }
  int i = 0; //index counter
  int k_final = k;
  int k_initial = 1;
  int g = 0; //count number of elements changed
  //k_initial = k;
  //initialize all elements

k = 1;
//for (i = 0; i <= n; i++)
while (i < n)
  {
   if (k != k_final)
   {
    if(arr[i] != 'X')
    {
     k++;
    }
   }
   else if (k == k_final)
   {
    if(arr[i] != 'X')
    {
     printf("%d\n", arr[i]);
     arr[i] = 'X';
```

```
      k = k_initial;
      g++;
     }
    }
    i++;
    if (i == n)
    {
     i = 0;
    }
    if ( g == n)
    {
     break;
    }
   }

for(i = 0; i < n; i++)
{
 if (arr[i] != 'X')
 {
  printf("%d\n", arr[i]);
 }
}

   //while(k < k_final){
         //ensures that k is passed through completely

   // counting to k,
   // mark the eliminated element
   // print the index of the marked element
   // repeat until only one element is unmarked




   // print the last one




   // release the memory of the array
   free (arr);
}

#endif
```