#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#define MAX\_SIZE 100

int numbers[MAX\_SIZE];

int numCount;

int largestEven = -1;

int smallestOdd = -1;

pthread\_mutex\_t mutex = PTHREAD\_MUTEX\_INITIALIZER;

void \*findLargestEven(void \*arg) {

for (int i = 0; i < numCount; i++) {

if (numbers[i] % 2 == 0) {

pthread\_mutex\_lock(&mutex);

if (numbers[i] > largestEven) {

largestEven = numbers[i];

}

pthread\_mutex\_unlock(&mutex);

}

}

pthread\_exit(NULL);

}

void \*findSmallestOdd(void \*arg) {

for (int i = 0; i < numCount; i++) {

if (numbers[i] % 2 != 0) {

pthread\_mutex\_lock(&mutex);

if (smallestOdd == -1 || numbers[i] < smallestOdd) {

smallestOdd = numbers[i];

}

pthread\_mutex\_unlock(&mutex);

}

}

pthread\_exit(NULL);

}

int main() {

printf("Enter the number of elements: ");

scanf("%d", &numCount);

printf("Enter the elements:\n");

for (int i = 0; i < numCount; i++) {

scanf("%d", &numbers[i]);

}

pthread\_t evenThread, oddThread;

pthread\_create(&evenThread, NULL, findLargestEven, NULL);

pthread\_create(&oddThread, NULL, findSmallestOdd, NULL);

pthread\_join(evenThread, NULL);

pthread\_join(oddThread, NULL);

if (largestEven == -1) {

printf("No even numbers found.\n");

} else {

printf("Largest even number: %d\n", largestEven);

}

if (smallestOdd == -1) {

printf("No odd numbers found.\n");

} else {

printf("Smallest odd number: %d\n", smallestOdd);

}

return 0;

}