# CF122A: Lucky Division, brute force/number theory, 1000

# <https://codeforces.com/problemset/problem/122/A>

num = int(input())  
if num % 4 and num % 7 and num % 47 and num % 74 and num % 447 \  
 and num % 474 and num % 477 and num % 747 and num % 774:  
 print("NO")  
else:  
 print("YES")

#492B.VanyaandLanterns,binarysearch/implementation/math/sortings, 1200

# <https://codeforces.com/problemset/problem/492/B>

n, l = (int(i) for i in input().split())  
st = [int(i) for i in input().split()]  
st.sort()  
pre, mx = st[0], 0  
for i in st[1:]:  
 if i - pre > mx:  
 mx = i - pre  
 pre = i  
print(max(st[0], mx / 2, l - st[-1]))

# OJ19949: 提取实体v0.3, cs10119 Final Exam, string,

# <http://cs101.openjudge.cn/practice/19949>

sm = 0  
for i in range(int(input())):  
 s = (j for j in input().split())  
 flag = False  
 for k in s:  
 if k.startswith("###") and k.endswith("###"):  
 if flag: sm -= 1  
 sm += 1  
 flag = True  
 else:  
 flag = False  
print(sm)

# OJ25274:赋值、浅拷贝与深拷贝

# <http://cs101.openjudge.cn/practice/25274>

import copy  
lst = [[1, 2, 3], 'abc', [1, 3], 4]  
a = lst  
b = lst[:]  
c = copy.deepcopy(lst)  
lst[0].append(4)  
lst[1] = 'def'  
lst.append(5)  
print(a)  
print(b)  
print(c)

# CF230B: T-primes, binary search/implementation/math/number theory, 1300

# <http://codeforces.com/problemset/problem/230/B>

from math import sqrt  
  
st = {2}  
for i in range(3, 1000000, 2):  
 flag = True  
 for j in range(3, int(sqrt(i)) + 1, 2):  
 if i % j == 0:  
 flag = False  
 break  
 if flag:  
 st.add(i)  
input()  
for i in map(int, input().split()):  
 if sqrt(i) in st:  
 print("YES")  
 else:  
 print("NO")

T-primes（不用筛法）比智能台灯难多了……CF230B这题创下了我单题所花时间的记录。其实算法思路很简单，只是难在时间复杂度的优化上。算法的优化也简单，其实如果代码结构合适的话，步进为一，从1到根号x遍历（比如我的代码）都能满足2秒的时间限制。教训：1.多用pypy，各种版本都试一试2.闫老师说的“因为print函数比较花时间，所以要把中间结果存下来一并输出”的做法不可取，实测是保存中间结果的过程更花时间。（我当时把结果保存为YES\nNO\n字符串，一直超时，直到改成每算一个结果就print一次）

# CF1000B: Light It Up, greedy, 1500

# <https://codeforces.com/problemset/problem/1000/B>

n, m = map(int, input().split())  
prei, lst, s1, s2, f, mx = 0, [], 0, 0, True, 0  
for i in [int(j) for j in input().split()] + [m]:  
 lst.append(i - prei)  
 f = not f  
 if f:  
 s2 += i - prei  
 prei = i  
for i in range(0, n, 2):  
 s1 += lst[i]  
 if lst[i] > 1 or lst[i + 1] > 1:  
 mx = max(mx, s1 + s2 - 1)  
 s2 -= lst[i + 1]  
if not f:  
 mx = max(mx, s1 + lst[n])  
else:  
 mx = max(mx, s1)  
print(mx)