'''

Students are given T-shirt numbers either 0 or 1 and standing in a line.

The line become circular. We have to group all students with T-shirt No with 1's

together at any location by swapping their locations.

Your task is to return the minimum number of swaps required to group all

students with T-shirt number 1's

A swap is taking two distinct positions in the line and swapping their values.

A line is said to be circular if the first student and last student are adjacent.

Input Format

------------

Line1:An integer N represents no of students

Line2:Space separated 0's or 1's represents T-shirt numbers

Output Format

-------------

An integer represents no of swaps

Sample Input-1:

---------------

7

0 1 0 1 1 0 0

Sample Output-1:

----------------

1

Explanation:

------------

Here are a few of the ways to group all the 1's together:

0 0 1 1 1 0 0 using 1 swap.

0 1 1 1 0 0 0 using 1 swap.

1 1 1 0 0 0 1 using 2 swaps (using the circular property of the array).

There is no way to group all 1's together with 0 swaps.

Thus, the minimum number of swaps required is 1.

Sample Input-2:

---------------

9

0 1 1 1 0 0 1 1 0

Sample Output-2:

----------------

2

Explanation:

------------

Here are a few of the ways to group all the 1's together:

1 1 1 0 0 0 0 1 1 using 2 swaps (using the circular property of the array).

1 1 1 1 1 0 0 0 0 using 2 swaps.

There is no way to group all 1's together with 0 or 1 swaps.

Thus, the minimum number of swaps required is 2.

Sample Input-3:

---------------

5

1 1 0 0 1

Sample Output-3:

----------------

0

Explanation:

------------

All the 1's are already grouped together due to the circular property of the array.

Thus, the minimum number of swaps required is 0.

'''

n = int(input())

l = list(map(int, input().split()))

count=0

for i in l:

if(i==1):

count+=1

mins= len(l)

for i in range(n):

swap= 0

for j in range(i, i + count):

if l[j % n] == 0:

swap+= 1

mins= min(mins, swap)

print(mins)

Sridhar brought his latest Apple iPhone 12 pro. He started his conversation

with one of his friend on WhatsApp with list of words.

Now it’s our task to find and return what are the most common words

in the list of words he used in sorted order based on occurrence from

largest to smallest. If any of words he used are having same occurrence

then consider the lexicographic order.

You will be given a list of words, you need to print top P number of

most common used words as described in the statement.

Input Format:

-------------

Line-1: comma separated line of words, list of words.

Line-2: An integer P, number of words to display.

Output Format:

--------------

Print P number of most common used words.

Sample Input-1:

---------------

ball,are,case,doll,egg,case,doll,egg,are,are,egg,case,are,egg,are,case

3

Sample Output-1:

----------------

[are, case, egg]

Sample Input-2:

---------------

ball,are,case,doll,egg,case,doll,egg,are,are,egg,case,are,egg,are

3

Sample Output-2:

----------------

[are, egg, case]

import java.util.\*;

public class Main{

public static void main(String[] s){

Scanner i=new Scanner(System.in);

String [] inp=i.nextLine().split(",");

int n=i.nextInt();

HashMap<String,Integer> m=new HashMap<>();

List<String> l=new ArrayList<>();

List<String> prin=new ArrayList<>();

for (String j:inp){

if(m.containsKey(j)){

m.put(j,m.get(j)+1);

}

else{

m.put(j,1);

l.add(j);

}

}

Collections.sort(l);

List<Integer> l1=new ArrayList<>();

for(String j:l){

l1.add(m.get(j));

}

while(n>0){

int n1=Collections.max(l1);

for(int i1=0;i1<l.size();i1++){

if(n1==l1.get(i1)){

prin.add(l.get(i1));

l.remove(i1);

l1.remove(i1);

n--;

break;

}

}

}

System.out.println(prin);

}

}

At university of Chicago a Computer Science programing faculty as a part of

teaching passion, in order to make newly joined students more enthusiastic

in learning the subject he will be giving a problem at the first day of semester.

The student who tops they will be awarded with cash prize. In regard to this

he asked the students to work on concept related to strings, he gave a task to

read a word and find the count of all the turn of phrases of the word, and

the phrases should be distinct.

Now it’s time for you to create a method which satisfies the above program.

A turn of phrases of a word is obtained by deleting

any number of characters (possibly zero) from the front of the word and

any number of characters (possibly zero) from the back of the word.

Input Format:

-------------

A single string, the word.

Output Format:

--------------

Print an integer, number of distinct phrases possible.

Sample Input-1:

---------------

aabbaba

Sample Output-1:

----------------

21

Explanation:

-------------

The turn of phrases of the word, which are distinct as follows:

a, b, aa, bb, ab, ba, aab, abb, bab, bba, aba, aabb, abba, bbab, baba,

aabba, abbab, bbaba, aabbab, abbaba, aabbaba

Sample Input-2:

---------------

kmithyd

Sample Output-2:

----------------

28

import java.util.\*;

public class Main{

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

String s=sc.next();

Set<String> hs=new HashSet<String>();

for(int i=0;i<s.length();i++){

for(int j=i+1;j<s.length();j++){

hs.add(s.substring(i,j+1));

}

}

System.out.println(hs.size()+s.length());

// System.out.println(hs);

}

}