Grind 8

20 May 2025 22:34

A. Bachgold Problem

time limit per test: 1 second

memory limit per test: 256 megabytes

Bachgold problem is very easy to formulate. Given a positive integer n represent it as a sum of **maximum possible** number of prime numbers. One can prove that such representation exists for any integer greater than 1.

Recall that integer k is called prime if it is greater than 1 and has exactly two positive integer divisors — 1 and k.

Input

The only line of the input contains a single integer n $(2 \le n \le 100\,000)$.

Output

The first line of the output contains a single integer k — maximum possible number of primes in representation.

The second line should contain k primes with their sum equal to n. You can print them in any order. If there are several optimal solution, print any of them.

Examples

input
```
5
```
output
```
2
2 3
```
input
```
6
```
output
```
3
2 2 2
```

n ↳ sum of primes }

10

1 +1+1+1+1+1+1+1+1+1

2 2 2 2 2 →

count → max }

sum →

10 ok

↓

sum of maxim

$\underbrace{1+1+1\ldots\ldots,}$ = 10

10 ✓ } NO

~~temp~~ primes → least prime

2

10

2 2 2 2 2 ×5 ✓

11 ↙ 3

3

2 2 2 2 ~~3~~ ~~2~~ } 10 , 11

→ 5+5 → 10 + 1 ok

11

2 2 2 2 3

1 2 3 4 5

no of numbers

5 → n/2

n==11

2 2 2 2 3

n → n/2

10/2 → 5 → 2 . . . .

11/2 → 5 → 2 2 2 2 3

B. Valera and X

time limit per test: 1 second
memory limit per test: 256 megabytes

Valera is a little boy. Yesterday he got a huge Math hometask at school, so Valera didn't have enough time to properly learn the English alphabet for his English lesson. Unfortunately, the English teacher decided to have a test on alphabet today. At the test Valera got a square piece of squared paper. The length of the side equals n squares (n is an odd number) and each unit square contains some small letter of the English alphabet.

Valera needs to know if the letters written on the square piece of paper form letter "X". Valera's teacher thinks that the letters on the piece of paper form an "X", if:

- on both diagonals of the square paper all letters are the same;
- all other squares of the paper (they are not on the diagonals) contain the same letter that is different from the letters on the diagonals.

Help Valera, write the program that completes the described task for him.

Input

The first line contains integer n ($3 \le n < 300$; n is odd). Each of the next n lines contains n small English letters — the description of Valera's paper.

Output

Print string "YES", if the letters on the paper form letter "X". Otherwise, print string "NO". Print the strings without quotes.

Examples

input
```
5
xooox
oxoxo
soxoo
oxoxo
xooox
```
n →, 5 col, 5 rows

output
```
NO
```

input
```
3
wsw
sws
wsw
```

output
```
YES
```

input
```
3
xpx
pxp
xpe
```

output
```
NO
```

Square piece

n →

```
5
xooox
oxoxo
soxoo
oxoxo
xooox
```

1st diagonal
2nd diagonal same

→ same letter
* different from x

NO →

square → ✓
n → n.

* built

* Matrix

$$\begin{bmatrix} x & o & o & o\ x \\ - & - & & - \\ - & - & -- & - \end{bmatrix}$$

next

n×n

int[ ][ ] grid = new int[n] [n];

building the matrix,

(n) times

0,1,2,3,4

for (i → 5) :   i = 0
                 j = 0
    String = 'xooox';
    5×5
    for (j → 5);
        grid[i][j] = string.charAt(j)

i, j = 0,1

diagonal in a matrix

$(0,0) \to$ $\quad$ $(0,0) == (0,1)$

3x3, 4x4, 5x5 $\to$

$0+2 = 2 == n-1$, $3-1$, $2 == 2$

3x3

$$\begin{bmatrix} (0,0) & (0,1) & (0,2) \\ (1,0) & (1,1) & (1,2) \\ (2,0) & (2,1) & (2,2) \end{bmatrix}$$

$2+0=2$

check both dia

i,j

i,j

$(0,0) == (1,1) == (2,2) =$

$(0,2) == (1,1) == (2,0)$

$\hookrightarrow$ 1) diagonal

$(0,1) == (1,2) == (1,0)$

$= (2,1)$

$\boxed{i == j} \to$ diagonal

$i+j = n-1$ $\to$ size of the square

$1+1 = 3-1$, $2 = 2$

$2 = 3-1$, $2 == 2$

$\begin{bmatrix} 0,0 & \\ & \end{bmatrix}$

$1 = 1$

$0+1 = 2-1$ $\quad$ $n=2$

2x2

$$\begin{bmatrix} (0,0) & (0,1) \\ (1,0) & (1,1) \end{bmatrix}$$

$1+0 = 2-1$

$i == j$

$i+j = n-1$

$\underbrace{2\times2,\ 3\times3,\ 4\times4}_{n \times n}$

int nodiagonal = grid[0] [1] ;

1) int diagonal = grid[0] [0]; , Flag = true

0, $\to$ n-1

for (i — n)

if Flag == False break $\quad$ $0 \to n-1$

for (j — n)

if (Flag == False) break

$\to$ if (i == j || i+j = n-1) {

if (grid[i][j] != diagonal) {

sout('NO') ; $\to$ once

Flag = False ;

}

2 when non-diagonal

} else {
if (grid[i][j] != non-diagonal) {
cout ("NO") → once → ✓
Flag = False;
}
}

nth, $n \to 10^{10^6}$, $1 \to 10^9$

C. Lovely Palindromes

time limit per test: 1 second
memory limit per test: 256 megabytes

Pari has a friend who loves palindrome numbers. A palindrome number is a number that reads the same forward or backward. For example 12321, 100001 and 1 are palindrome numbers, while 112 and 1021 are not.

Pari is trying to love them too, but only very special and gifted people can understand the beauty behind palindrome numbers. Pari loves integers with even length (i.e. the numbers with even number of digits), so she tries to see a lot of big palindrome numbers with even length (like a 2-digit 11 or 6-digit 122221), so maybe she could see something in them.

Now Pari asks you to write a program that gets a huge integer n from the input and tells what is the n-th even-length positive palindrome number?

Input

The only line of the input contains a single integer n ($1 \le n \le 10^{100\,000}$).

Output

Print the n-th even-length palindrome number.

Examples

input
```
1
```
output
```
11
```

input
```
10
```
output
```
1001
```

Note

The first 10 even-length palindrome numbers are 11, 22, 33, … , 88, 99 and 1001.

n 4

"1" + "1" ⇒ 11

10^{10^6} → big

int ✗ long ✗

10+01 ⇒ 1001

maths ✓

palindrome 10^{10^6} >>>> 10^9

ABA

ABA

ABA → int

101, 1001, }

222 ✓

1001

11, 12, 13, 4

← Even + Palindrome

n th palindromic number

1 → 11
2 → 22

1 1

2 length

int → ≈ 10^9
long → ≈ 10^{18}

10^{10^6} >>>>

long long double → ✓ → n th palin

{ math direction —
~~int~~ num >>
operation >>> 10^9 → }

math

math { ← pen paper }

→ Pattern

1 → 11
2 → 22
3 → 33
4 → 44

10 → 1001
11 → 1111
12 → 1221
13 → 1331

10 + 01 ⇒ 1001

n + rev(n) → ans

4	44
5	55
6	66
7	77
8	88
9	99

13 → 1331

14 → 1441

14 → 1441

25 → 2552

1 lone

String s = input

String p = s + rev(s) } Ans

D. Vanya and Lanterns

time limit per test: 1 second

memory limit per test: 256 megabytes

Vanya walks late at night along a straight street of length l, lit by n lanterns. Consider the coordinate system with the beginning of the street corresponding to the point 0, and its end corresponding to the point l. Then the i-th lantern is at the point a_i. The lantern lights all points of the street that are at the distance of at most d from it, where d is some positive number, common for all lanterns.

Vanya wonders: what is the minimum light radius d should the lanterns have to light the whole street? 3

Input

The first line contains two integers n, l ($1 \le n \le 1000$, $1 \le l \le 10^9$) — the number of lanterns and the length of the street respectively.

The next line contains n integers a_i ($0 \le a_i \le l$). Multiple lanterns can be located at the same point. The lanterns may be located at the ends of the street.

Output

Print the minimum light radius d, needed to light the whole street. The answer will be considered correct if its absolute or relative error doesn't exceed 10^{-9}.

Examples

input

```
7 15
15 5 3 7 9 14 0
```

output

```
2.5000000000
```

input

```
2 5
2 5
```

output

```
2.0000000000
```

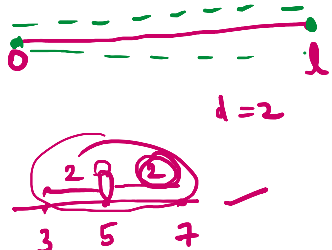
code

Note

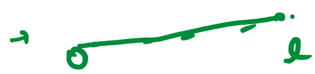
Consider the second sample. At $d = 2$ the first lantern will light the segment $[0, 4]$ of the street, and the second lantern will light segment $[3, 5]$. Thus, the whole street will be lit.

street → l d d

lanters → n, ith → a_i



a_i → position of lan



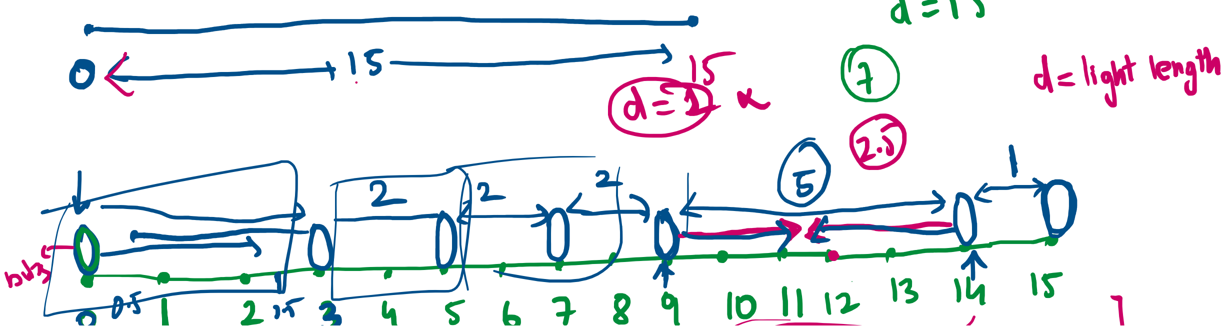
bulbs position

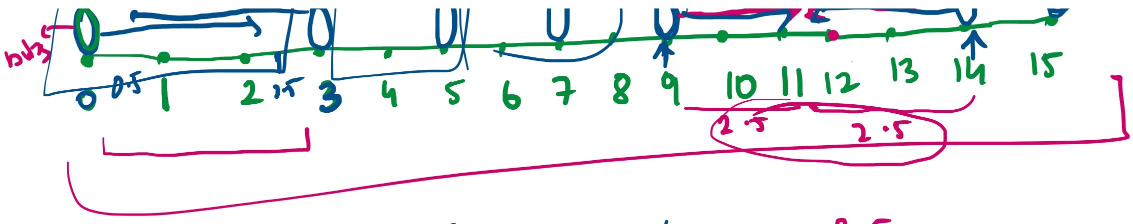
15, 5, 3, 7, 9, 14, 0

d = 15

bulbs ← n 7

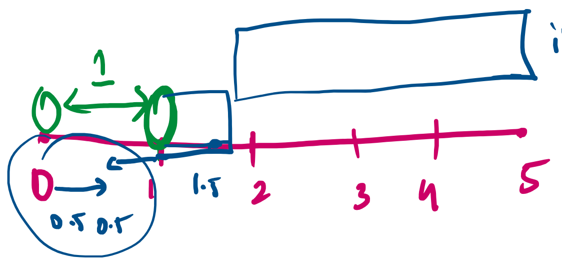
l 15 → end point of the street
→ length





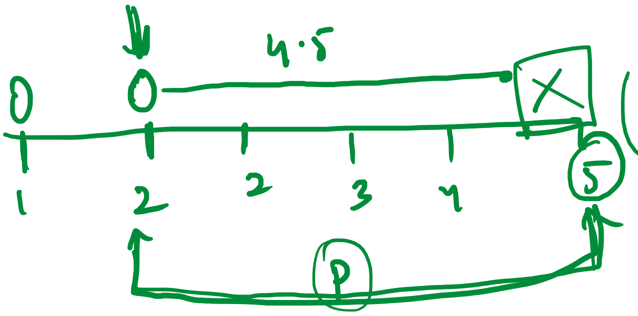
$\frac{5}{2}$ → max / 2 → 2.5

1) Right ans → max/2 − Ans → will it always



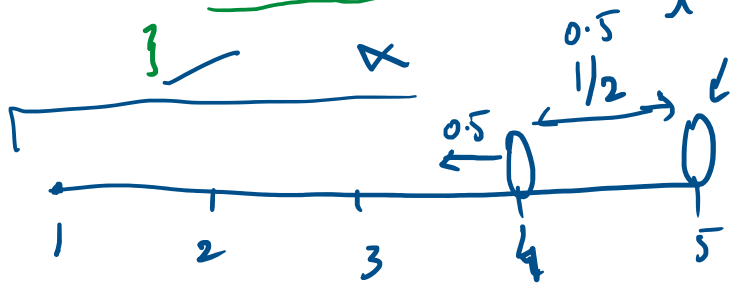
2 bulbs.

ans = 1/2 ⇒ 0.5



(l − last lantern) → max

if (p > max) {
max = p;
}



l − last → 0

5 − 5 = 0

first lanter − 0 ⇒ first lanter > max

error → 10^{-9}

$\frac{5}{2}$ → 2.5000000000

~~2.5~~ 2

2.4999999999 — 2.500000000 → ✓

10^{-9} → ter ✓

$< 10^{-9}$

↳ part ✓

BS → decimal

---

m fruits

| seller item

E. Fruits

time limit per test: 1 second

memory limit per test: 256 megabytes

The spring is coming and it means that a lot of fruits appear on the counters. One sunny day little boy Valera decided to go shopping. He made a list of m fruits he wanted to buy. If Valera want to buy more than one fruit of some kind, he includes it into the list several times.

When he came to the fruit stall of Ashot, he saw that the seller hadn't distributed price tags to the goods, but put all price tags on the counter. Later Ashot will attach every price tag to some kind of fruits, and Valera will be able to count the total price of all fruits from his list. But Valera wants to know now what can be the smallest total price (in case of the most «lucky» for him distribution of price tags) and the largest total price (in case of the most «unlucky» for him distribution of price tags).

Input

The first line of the input contains two integer number n and m $(1 \le n, m \le 100)$ — the number of price tags (which is equal to the number of different kinds of fruits that Ashot sells) and the number of items in Valera's list. The second line contains n space-separated positive integer numbers. Each of them doesn't exceed 100 and stands for the price of one fruit of some kind. The following m lines contain names of the fruits from the list. Each name is a non-empty string of small Latin letters which length doesn't exceed 32. It is guaranteed that the number of distinct fruits from the list is less of equal to n. Also it is known that the seller has in stock all fruits that Valera wants to buy.

Output

Print two numbers a and b $(a \le b)$ — the minimum and the maximum possible sum which Valera may need to buy all fruits from his list.

Examples

input Copy

```
5 3
4 2 1 10 5
apple
orange
mango
```

fruits

output Copy

```
7 19
```

min cost   max cost

n → price tags , m → fruits

[4 2 1 10 5] } price tags

apple
orange
mango

1₹ Apple
2₹ orange
4₹ mango

min ⇒ 7₹

apple → 10₹
orange → 5₹
mango → 4₹
↓
19₹

[1, 2, 4, 5, 10]

ASC order

```
6 5
3 5 1 6 8 1
peach
grapefruit
```

price tags

3 5 1 6 8 1

5×2 = 10

P B → orange → 2×8

```
peach
grapefruit
banana
orange
orange

output

11 30
```

price tag

₹ price tag

$5 \times 2 = 10$

p 8 → orange → 2×8 → 16

$8 + 6 \Rightarrow 14$

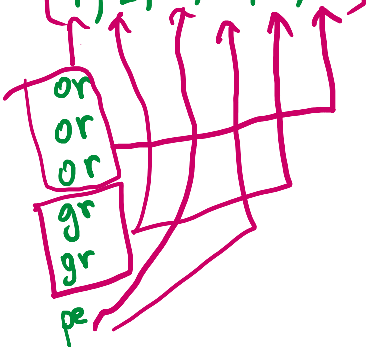
[1, 1, 3, 5, 6, 8]

₹1 × 2 + ₹1 + ₹3 + ₹5

→ 11

min → $1 \times 3 + 2 \times 2 + 3 \times 1$

[1, 2, 3, 4, 5, 6]



min

get the idea

nil

imp

* code ✓ → hints

* Hashmap → col frequencies / freq array

sort the frequencies

While

price tag = [3, 4, 5, 7, 8]

[1, 1, 2, 3]

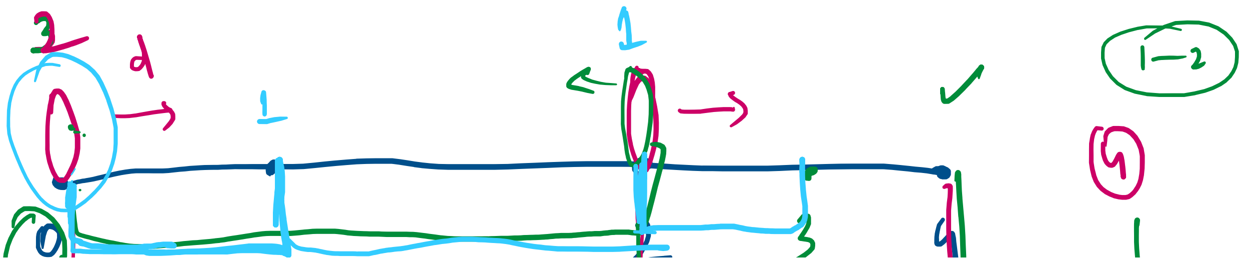
dis

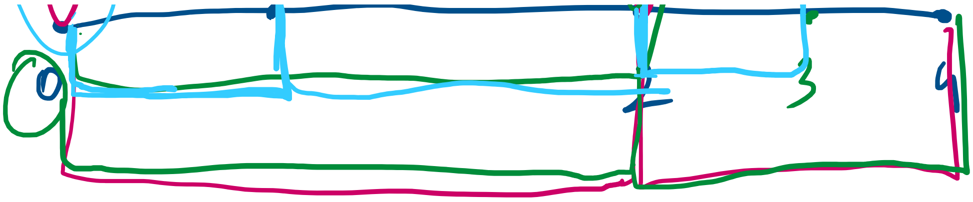
∅ [3, 2, 1, 1] →

1200 rated ✓

min →

max →





2^{9} ← street } M

(3,4)
(4,5)
(4,6)
(8,10)

[(3,4), (4,5)(4,6), (8,10)]

$O(n^2)$ →

~~3~~ 4 - ~~22~~

LB = 3

UB = 4

[(3,4), (4,5), (4,6), (8,10)]

[(3,4), (4,5), (4,6), (8,10)] 9

(8,~~10~~)

(3,4)
(4,5)
(4,6)