Given a number **N**. The task is to find the N^th catalan number.
The first few Catalan numbers for **N = 0, 1, 2, 3,** ... are **1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, ...**
Note: Positions start from 0 as shown above.

Catalan numbers:

$$C_n = \binom{2n}{n} - \binom{2n}{n+1} \qquad n \geq 0$$

bardziej dynamicznie:

$$\begin{cases} C_{n+1} = \sum_{i=0}^{n} C_i C_{n-i} & n \geq 0 \\ C_0 = 1 \end{cases}$$

lub

$k = n+1$

$k - 1 = n$

$C_0 = 1$

$$C_{n+1} = \frac{2(2n+1)}{n+2} C_n \qquad n \geq 0$$

$$C_k = \frac{2(2(k-1)+1)}{k+1} C_{k-1} = \frac{2(2k-1)}{\cancel{2}\, k+1} C_k = \frac{4k-\cancel{2}}{k+1} C_k$$

Given a number N. Find the minimum number of operations required to reach **N** starting from **0**. You have 2 operations available:

- Double the number
- Add one to the number

Example 1:

```
Input:
N = 8
Output: 4
Explanation: 0 + 1 = 1, 1 + 1 = 2,
2 * 2 = 4, 4 * 2 = 8
```

â€‹**Example 2:**

```
Input:
N = 7
Output: 5
Explanation: 0 + 1 = 1, 1 + 1 = 2,
1 + 2 = 3, 3 * 2 = 6, 6 + 1 = 7
```

start = 0

$f(n)$ - min. liczba operacji, aby uzyskać n

$$f(n) = \min\left(f(n-1)+1,\ f\left(\frac{n}{2}\right)+1 \right)$$

$\frac{n}{2} \in \mathbb{N}$

$$n-1 = \frac{n}{2} \quad / \cdot 2$$

$$2n - 2 = n$$

$$n = 2$$

Given string **s** containing characters as integers only, the task is to delete all characters of this string in a minimum number of steps wherein one step you can delete the substring which is a palindrome. After deleting a substring remaining parts are concatenated.

Example 1:

```
Input: s = "2553432"
Output: 2
Explanation: In first step remove "55",
then string becomes "23432" which is a
palindrome.
```

Example 2:

```
Input: s = "1234"
Output: 4
Explanation: Remove each character in
each step
```

$f(i, j)$ - min liczba kroków, aby usunąć $s[i, \ldots, j]$

1° $f(i+1, j)$ - usuwamy 1 literkę (jest palindromem)

2° $f(i+1, k-1) + f(k+1, j) \mid s[i] == s[k]$

$k \in [i+2, j]$

3° $f(i+2, j)$

Minimum number of Coins

Easy Accuracy: 76.4% Submissions: 5120 Points: 2

Given an infinite supply of each denomination of Indian currency { 1, 2, 5, 10, 20, 50, 100, 200, 500, 2000 } and a target value N.
Find the minimum number of coins and/or notes needed to make the change for Rs N.

Example 1:

```
Input: N = 43
Output: 20 20 2 1
Explaination:
Minimum number of coins and notes needed
to make 43.
```

Example 2:

```
Input: N = 1000
Output: 500 500
Explaination: minimum possible notes
is 2 notes of 500.
```

<= Na ćwiczeniach

schody omówione =>

A frog jumps either 1, 2, or 3 steps to go to the top. In how many ways can it reach the top. As the answer will be large find the answer modulo 1000000007.

Example 1:

```
Input:
N = 1
Output: 1
```

Example 2:

```
Input:
N = 4
Output: 7
Explanation:Below are the 7 ways to reach
4
1 step + 1 step + 1 step + 1 step
1 step + 2 step + 1 step
2 step + 1 step + 1 step
1 step + 1 step + 2 step
2 step + 2 step
3 step + 1 step
1 step + 3 step
```

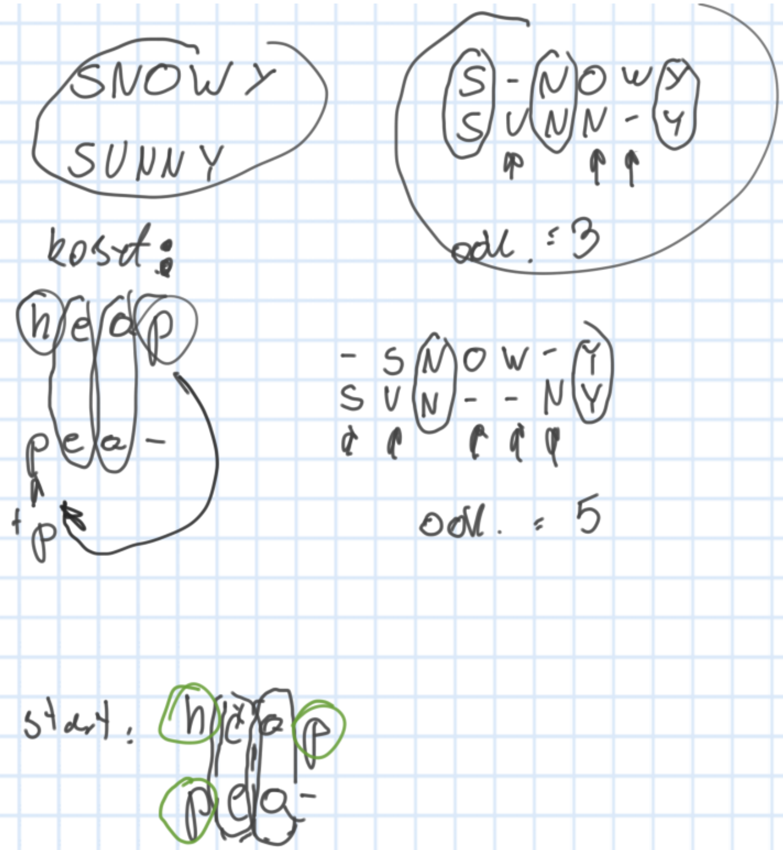
Given two strings **str1** and **str2**. The task is to remove or insert the minimum number of characters from/in **str1** so as to transform it into **str2**. It could be possible that the same character needs to be removed/deleted from one point of str1 and inserted to some another point.

Example 1:

```
Input: str1 = "heap", str2 = "pea"
Output: 3
Explanation: 2 deletions and 1 insertion
p and h deleted from heap. Then, p is
inserted at the beginning One thing to
note, though p was required yet it was
removed/deleted first from its position
and then it is inserted to some other
position. Thus, p contributes one to the
deletion_count and one to the
insertion_count.
```

Example 2:

```
Input : str1 = "geeksforgeeks"
str2 = "geeks"
Output: 8
Explanation: 8 deletions
```



Minimum four sum subsequence

Easy Accuracy: 50.3% Submissions: 2533 Points: 2

Given an array A[] of positive integers. The task is to complete the function which returns an integer denoting the minimum sum subsequence from the array such that at least one value among all groups of four consecutive elements is picked.

Examples:

```
Input: A[] = {1, 2, 3, 4, 5, 6, 7, 8}
Output: 6
6 is sum of output subsequence {1, 5}
Following 4 length subarrays are possible
(1, 2, 3, 4), (2, 3, 4, 5), (3, 4, 5, 6),
(4, 5, 6, 7), (5, 6, 7, 8)
Here, Our subsequence {1, 5} has an element from all
above groups of four consecutive elements.
```

```
Input: A[] = {1, 2, 3, 3, 4, 5, 6, 1}
Output: 4
The subsequence is {3, 1}. Here we consider
second three.
```

$f(i)$ = najlepsza suma dla $A[0, \dots, i]$, pod warunkiem, że bierzemy $A[i]$

$$f(i) = A[i] + \min\big(f(i-1), f(i-2), f(i-3), f(i-4)\big)$$

wynik: $\min\big(f(n-1), f(n-2), f(n-3), f(n-4)\big)$

Given a NxN matrix of positive integers. There are only three possible moves from a cell **Matrix[r][c]**.

1. Matrix [r+1] [c]
2. Matrix [r+1] [c-1]
3. Matrix [r+1] [c+1]

Starting from any column in row 0 return the largest sum of any of the paths up to row N-1.

Example 1:

```
Input: N = 2
Matrix = {{348, 391},
          {618, 193}}
Output: 1009
Explaination: The best path is 391 -> 618.
It gives the sum = 1009.
```

Example 2:

```
Input: N = 2
Matrix = {{2, 2},
          {2, 2}}
Output: 4
Explaination: No matter which path is
chosen, the output is 4.
```

ruchy :

$f(i, j)$ - max. suma aby dostać się na $M[i][j]$

$$f(i, j) = M[i][j] + \max\big(f(i-1, j), f(i-1, j-1), f(i-1, j+1)\big)$$

Optimal Strategy For A Game

Medium Accuracy: 52.29% Submissions: 20192 Points: 4

You are given an array **A of size N**. The array contains integers and is of **even length**. The elements of the array represent N **coin** of **values $V_1, V_2, \ldots V_n$**. You play against an opponent in an **alternating** way.

In each **turn**, a player selects either the **first or last coin** from the **row**, removes it from the row permanently, and **receives the value** of the coin.

You need to determine the **maximum possible amount of money** you can win if you **go first**.
Note: Both the players are playing optimally.

Example 1:

```
Input:
N = 4
A[] = {5,3,7,10}
Output: 15
Explanation: The user collects maximum
value as 15(10 + 5)
```

Example 2:

```
Input:
N = 4
A[] = {8,15,3,7}
Output: 22
Explanation: The user collects maximum
value as 22(7 + 15)
```

$A: [8, 15, 3, 7]$

$f(i, j)$ - max wygrana na tablicy $A[i, \ldots, j]$ zakładając że ruch jest zajebisty

$f(i,j) = \max\big(A[i] + \min(f(i+2, j), f(i+1, j-1)),$
$A[j] + \min(f(i+1, j-1), f(i, j-2))\big)$

1° biorę z lewej
rywal
biorę z lewej
biorę z prawej

2° biorę z prawej
rywal
z lewej
z prawej

Given an array **arr[]** of size **N**, check if it can be partitioned into two parts such that the sum of elements in both parts is the same.

Example 1:

```
Input: N = 4
arr = {1, 5, 11, 5}
Output: YES
Explaination:
The two parts are {1, 5, 5} and {11}.
```

Example 2:

```
Input: N = 3
arr = {1, 3, 5}
Output: NO
Explaination: This array can never be
partitioned into two such parts.
```

$f(i, j)$ - czy istnieje podzbiór $A[0, \dots, i]$ do sumy j

1° $A[i]$ bierzemy

2° $A[i]$ nie bierzemy

$$f(i, j) = f(i-1, j) \text{ or } f(i-1, j - A[i])$$

Given a value N, find the number of ways to make change for N cents, if we have infinite supply of each of S = { S_1, S_2, .. , S_M } valued coins.

Example 1:

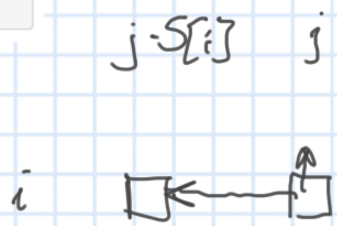
```
Input:
n = 4 , m = 3
S[] = {1,2,3}
Output: 4
Explanation: Four Possible ways are:
{1,1,1,1},{1,1,2},{2,2},{1,3}.
```

Example 2:

```
Input:
n = 10 , m = 4
S[] ={2,5,3,6}
Output: 5
Explanation: Five Possible ways are:
{2,2,2,2,2}, {2,2,3,3}, {2,2,6}, {2,3,5}
and {5,5}.
```

$f(m, n)$ - liczba sposobów, aby wydać n centów wśród monet $S[1, \dots, m]$

$$f(i, j) = f(i-1, j) + f(i, j - S[i])$$



Given a string **A** and a dictionary of n words **B**, find out if A can be segmented into a space-separated sequence of dictionary words.

Note: From the dictionary **B** each word can be taken any number of times and in any order.

Example 1:

```
Input:
n = 12
B = { "i", "like", "sam",
"sung", "samsung", "mobile",
"ice","cream", "icecream",
"man", "go", "mango" }
A = "ilike"
Output:
1
Explanation:
The string can be segmented as "i like".
```

$f(i,j)$: czy $s[i, \ldots, j]$ może być rozdzielone

$$f(i,j) : f(i + len(k), j) \mid \begin{matrix} k \in A \\ s[i, i + len(k)] == k \end{matrix}$$

Given **N** dice each with **M** faces, numbered from 1 to M, find the number of ways to get sum **X**. X is the summation of values on each face when all the dice are thrown.

Example 1:

```
Input:
M = 6, N = 3, X = 12
Output:
25
Explanation:
There are 25 total ways to get
the Sum 12 using 3 dices with
faces from 1 to 6.
```

Example 2:

```
Input:
M = 2, N = 3, X = 6
Output:
1
Explanation:
There is only 1 way to get
the Sum 6 using 3 dices with
faces from 1 to 2. All the
dices will have to land on 2.
```

$f(n, x)$ - na ile sposobów można uzyskać liczby ~~xn~~ używając n kostek

$$f(n, x) = \sum_{k=1}^{M} f(n-1, x-k)$$

You are given **N** identical eggs and you have access to a **K**-floored building from **1** to **K**.

There exists a floor **f** where **0 <= f <= K** such that any egg dropped at a floor higher than **f** will break, and any egg dropped **at or below** floor **f** will **not break**. There are few rules given below.

- An egg that survives a fall can be used again.
- A broken egg must be discarded.
- The effect of a fall is the same for all eggs.
- If the egg doesn't break at a certain floor, it will not break at any floor below.
- If the eggs breaks at a certain floor, it will break at any floor above.

Return the minimum number of moves that you need to determine with certainty what the value of **f** is.

For more description on this problem see wiki page

Example 1:

```
Input:
N = 1, K = 2
Output: 2
Explanation:
1. Drop the egg from floor 1. If it
   breaks, we know that f = 0.
2. Otherwise, drop the egg from floor 2.
   If it breaks, we know that f = 1.
3. If it does not break, then we know f = 2.
4. Hence, we need at minimum 2 moves to
   determine with certainty what the value of f is.
```

$O(n * k^2)$

$f(i,j) =$ min. liczba operacji aby przy pomocy i jajek zbadać f pięter

$$f(i,j) = 1 + \max\left(f(i-1, j-1), f(i, j-x)\right)$$
$$x \in [1, j]$$