```
167. Two sum II. Input away is sorted.
TP: numbers = [2,7,11,15] taget = 9
ofp: [1,2]

Explanation: The sum of 2 and 7 2s q. Therefore,

Index, = 1, index<sub>2</sub> = 2 we return [1,2].
Brute force: (Beginner):
Class Salution {

public Int [] toossum [int [] numbers int target) {
        for lint i = 0; 1 < numbers length, itt) of
          for (int ) = i +1; j < numbers - length; j++) {
              if [numbers [i] + numbers [i] ==target) [
                   return new ind[][i+1,j+1];
                         : (i) enner = (i) encer
      return new int[][];
                                       thi mostor
   Time \rightarrow o(n^2) space \rightarrow o(1)
optimal approach (Two pointers)
                                        [c. 1, 1] & corner
   n left -) stead of owney
                                       brands toof ) 1 = 3
    * right -) end of array
                                     1 . [i] eruss ( + ) .
  ogic:

1. If numbers [left] + numbers [sight] = = target -> answer
found
  2. If the sum is too big, nove right leftward
  3. If the own is too small, move left rightward.
    10 c 2092
```

(en) 9 - 2091"

dar Solution & public int [] seasum [int [] numbers, int seaget)? get that the int right - numbers, longth -1; while (deft a right) of and own a humbers [left] a numbers [right] if (sum = = larget) f. networ new Int [] { Left +1, right +1}; //1 - indered. get if (sum c target) } left ++; called at abundan some was the page of Jelse } right -- ; was officer endling grademan ins I twinted to moderny . The of grideresse sure return new int [] {}; meda salul est of string in fahruntees minteel s Dry Run 1 numbers = (2,7,11, 45) target = 9 1. left = 0(2), sight = 3(15) -> sum = 17 >9 -> move right -> 2 2. left = 0(2), right = 2 (11) -> sum = 13 > 9 -> more sight -) 3. left = 0(2), right = 1 (7) -> sum = 9 -> 1 consues = [1,2].