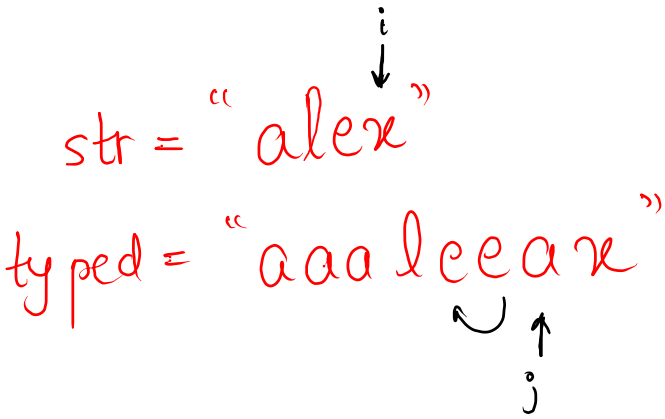
Long Pressed Name



Note:-

↳ sequence should be same

psudo code

1) initialise i at 0 and j at 0
2) if char at i == char at j
   2.1) i++
3) else if prev char != current j
   3.1) return false

j++

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    String str = scn.next();
    String typed = scn.next();
    System.out.println(longPressed(str, typed));
}

public static boolean longPressed(String str, String typed) {
    int i = 0;
    int j = 0;
    while ( j < typed.length() ) {
        if ( i < str.length() && str.charAt(i) == typed.charAt(j) ) {
            i++;
        } else if ( j == 0 || typed.charAt(j) != typed.charAt(j - 1) ) {
            return false;
        }
        j++;
    }
    if (i == str.length())
        return true;
    else
        return false;
}
```

curr

prev.

true

str = "alexx"

typed = "aaaalee x"

i

j

false

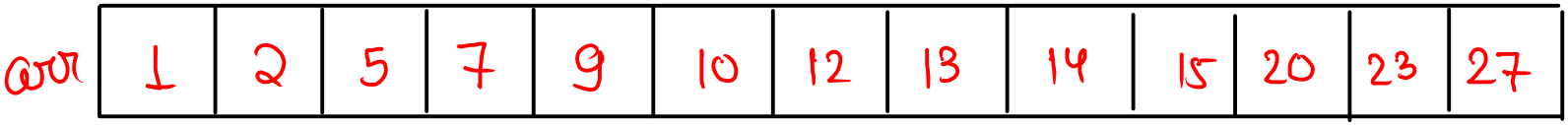
str = "aniket"

typed = "aanikett";

i

j

⇒ Binary Search



tar = 12

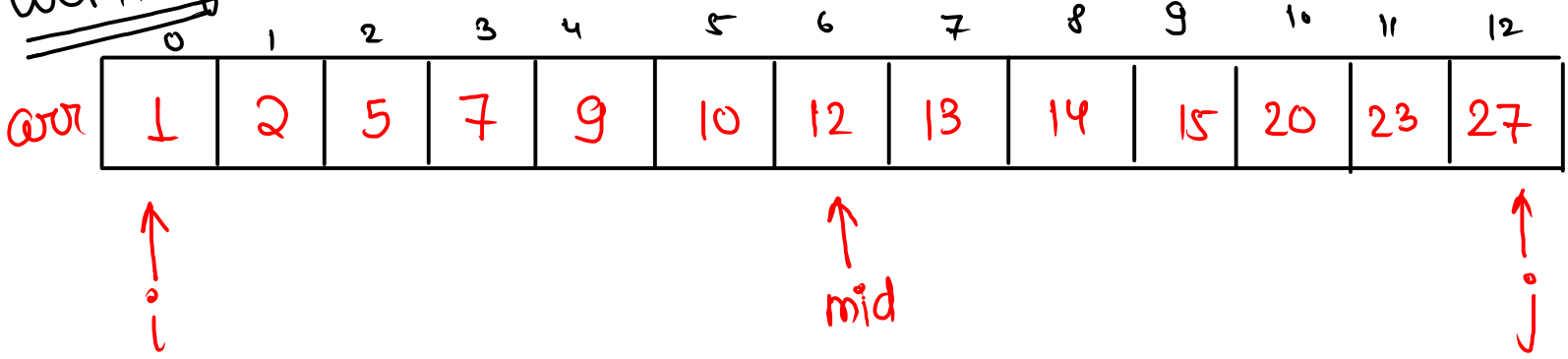
O(N)

note:-

conditions:-

- array need to be sorted
- it will search an element in logN

Working



target = 10

Code

```
while ( i <= j ) {
    mid = (i+j)/2
a   if ( arr[mid] == target) {
        return mid;
b   }else if ( arr[mid] > target) {
        j = mid-1;
    } else if (arr[mid] < target) {
c       i = mid+1;
    }
}
```

assume len = n

$$n + \frac{n}{2} + \frac{n}{4} + \frac{n}{8} + \frac{n}{16} + \cdots\cdots + 1 = \log(N)$$

Binary Search in an Array

n = 7

arr = 1 2 3 4 5 6 7 (3 at index 2)

target = 3

ans = 2

Code

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }
    int target = scn.nextInt();

    System.out.println(binarySearch(arr, n, target));
}

public static int binarySearch(int[] arr, int n, int target) {
    int i = 0;
    int j = n - 1;
    while ( i <= j ) {
        int mid = (i + j) / 2;
        if ( arr[mid] == target ) {
            return mid;
        } else if ( arr[mid] < target ) {
            i = mid + 1;
        } else {
            j = mid - 1;
        }
    }
    return -1;
}
```

Search Character

T.C = O(log N)

n = 7

	0	1	2	3	4	5	6
arr	'a'	'c'	'd'	'f'	'k'	'l'	'm'

ei → 3, si → 4, mid → 4

target = 'i'

```
while( si<=ei)
mid = (si+ei)/2;
a  if( arr[mid] == target){
       si = mid+1;
b  } else if( arr[mid] < target){
       si = mid+1;
c  } else {
       ei = mid-1;
   }
```

Ans = arr[si]

Code

```
public static void searchCharacter(char[] arr, int n, char ch) {
    int si = 0;
    int ei = n - 1;
    while (si <= ei) {
        int mid = (si + ei) / 2;
        if ( arr[mid] <= ch ) {
            si = mid + 1;
        } else {
            ei = mid - 1;
        }
    }
    if (si == n) {
        System.out.println("-1");
    } else {
        System.out.println(arr[si]);
    }
}
```