Find Unique

(Arrays as hashmap)

str = "152215021";

unique no.

1
5
2
0

4

create freq. array

```
char ch = str.charAt(i);
int idx = ch - '0';
```

	0	1	2	3	4	5	6	7	8	9
freq (int)	1	3	3	0	0	2	0	0	0	0

str = "152215021";

	0	1	2	3	4	5	6	7	8	9
freq (boolean)	T	T	T	F	F	T	F	F	F	F

```
char ch = str.charAt(i);
int idx = ch - '0';
freq[idx] = true;
```

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    String str = scn.nextLine();
    System.out.println(findUnique(str));
}
public static int findUnique(String str) {
    boolean[] check = new boolean[10];
    for (int i = 0; i < str.length(); i++) {
        char ch = str.charAt(i);
        int idx = ch - '0';
        check[idx] = true;
    }
    int count = 0;
    for (boolean i : check) {
        if ( i == true ) {
            count++;
        }
    }
    return count;
}
```

T.C = O(n)

where n is size of string

S.C = O(1)

Is Palindrome (2 pointer)

str = "radar";

i j

psudo code

1) make 2 pointers at 0 and (n-1)
2) loop until i < j
   2.1) check if char at i != char at j then return false
3) return true

code

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    String str = scn.nextLine();

    boolean ans = isPalindrome(str);
    if ( ans == true ) {
        System.out.println("Palindrome");
    } else {
        System.out.println("Not a Palindrome");
    }
}
public static boolean isPalindrome(String str) {
    int i = 0;
    int j = str.length() - 1;
    while ( i < j ) {
        if ( str.charAt(i) != str.charAt(j) ) {
            return false;
        }
        i++;
        j--;
    }
    return true;
}
```

code

```java
public boolean isPalindrome(String s) {
    String str = "";
    for (int i = 0; i < s.length(); i++) {
        char ch = s.charAt(i);
        if ( (ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z') || (ch >= '0' && ch <= '9') ) {
            str = str + ch;
        }
    }
    str = str.toLowerCase();
    return isPalindrome1(str);
}
public static boolean isPalindrome1(String str) {
    int i = 0;
    int j = str.length() - 1;
    while ( i < j ) {
        if ( str.charAt(i) != str.charAt(j) ) {
            return false;
        }
        i++;
        j--;
    }
    return true;
}
```

s = "A man is 012, abc".

str = "Amanis012abc"

a

false

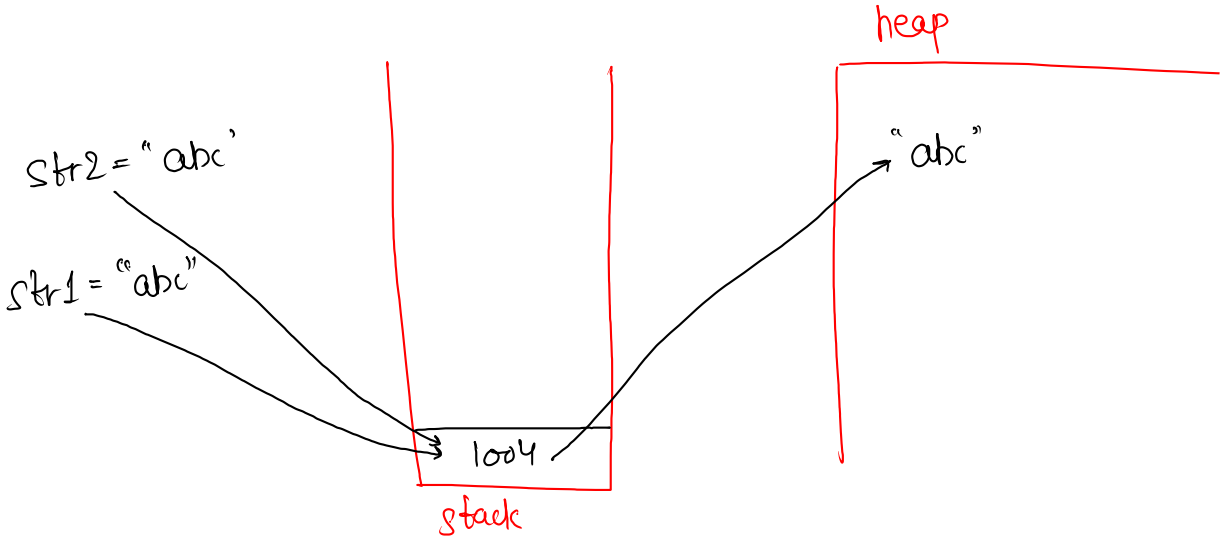
⇒ **String is immutable**

str = "abc"

str1 = "efg"

str2 = str + str1 ;

Q level arch.



Note:- whenever you are comparing 2 strings never use == or != instead use .equals

Locate the Target String

str = "geeksterst";

0 1 2 3 4 5 6 7 8 9

target = "st"

0 1

ans = 4

brute force

↳ generate all substrings and check

str = "geekster" , tar = "st"

g	e	e	k	s
ge	ee	ek	ks	$\boxed{st}$
gee	eek	eks	kst	
geek	eeks	ekst	kste	
geeks	eekst	ekste	kster	
geekst	eekste	ekster		
geekste	eekster			
geekster				

(Too much mehenat)

i

str = "geeksterstep"

0 1 2 3 4 5 6 7 8 9 10 11

target = "step"

0 1 2 3

j

(i+j)

8 + 1 = 9

8 + 2 = 10

8 + 3 = 11

psudo code

1) make 2 pointers
2) loop until i < len of str
   2.1) loop unt j < len of target
      2.1.1) char at i != char at j
         break;
      2.1.2) if j == tar. length
         then return i

str = "abcdabc"

0 1 2 3 4 5 6

error

tar = "abcde"

0 1 2 3 4

j

tar = j

str = i+j

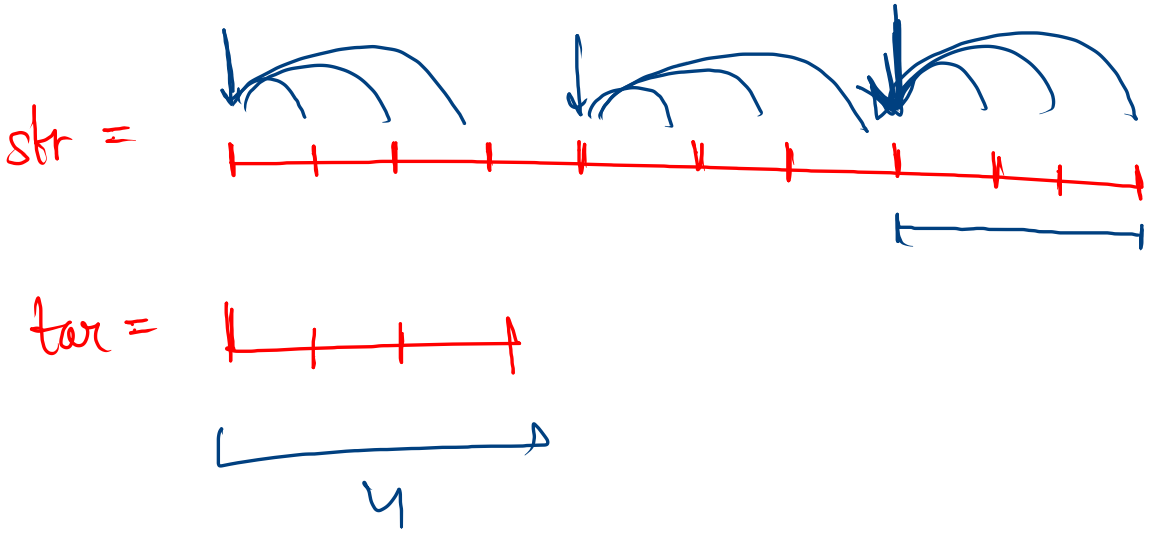
4+1

4+2

4+3

Code

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    String str = scn.nextLine();
    String target = scn.nextLine();
    System.out.println(locateTarget(str, target));
}
public static int locateTarget(String str, String target) {
    for (int i = 0; i <= str.length() - target.length(); i++) {
        for (int j = 0; j < target.length(); j++) {
            if ( target.charAt(j) != str.charAt(i + j) ) {
                break;
            }
            if ( j == target.length() - 1 ) {
                return i;
            }
        }
    }
    return -1;
}
```


str =
tar =
4