Print Indices of Vowels

str = "bcaufguio" ;

$$\begin{array}{ccccccccc} b & c & a & u & f & g & u & i & o \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array}$$

ans = 2 3 6 7 8

$\rightarrow$ abc efg

scn.next() ; $\longrightarrow$ "abc"

scn.nextLine() ; $\longrightarrow$ "abc efg"

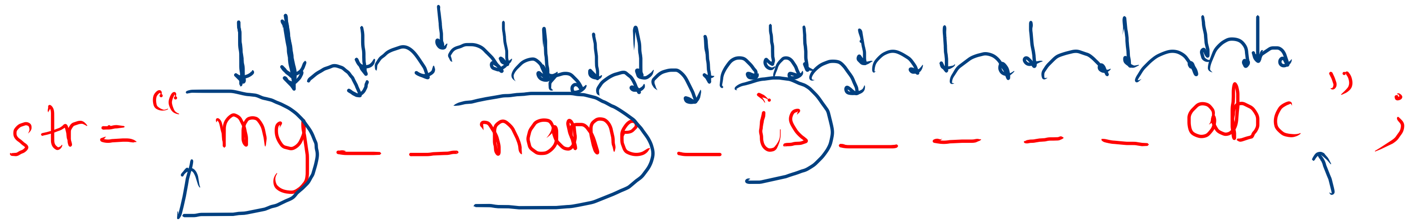
Code

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


    int n = str.length();
    for (int i = 0; i < n; i++) {
        if ( isVowel( str.charAt(i) ) ) {
            System.out.print(i + " ");
        }
    }
}

public static boolean isVowel(char c) {
    // if ( c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u' ) {
    //     return true;
    // } else {
    //     return false;
    // }

    return c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u';
}
```

Count Words

str = "It_ is _ a _ sentense";

ans = 4



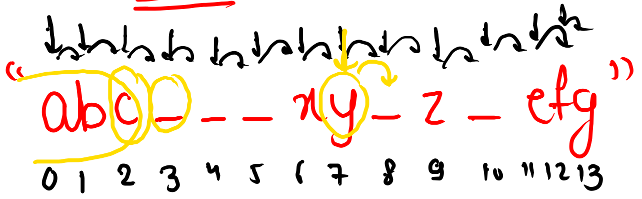
str = "my__name_is____abc";

str.charAt(i) != ' ' && str.charAt(i+1) == ' '

code

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


    int n = str.length();
    int count = 0;
    for (int i = 0; i < n - 1; i++) {
        char curr = str.charAt(i);
        char next = str.charAt(i + 1);
        if ( curr != ' ' && next == ' ' ) {
            count++;
        }
    }
    System.out.println(count + 1);
}
```

str



count = 0

i = 0, ✗

i = 1, ✗

i = 2, ✓

count = 1

i = 3, ✗

i = 4, ✗

i = 5, ✗

i = 6, ✗

i = 7, ✓

count = 2

i = 8, ✗

i = 9, ✓

count = 3

i = 10, ✗

i = 11, ✗

i = 12, ✗

→ **Inbuilt function** (String[] arr = str.split(" ");)

str = " This _ is _ a _ sentence" ;

arr (String)

→ str.split(" ");

"This"	"is"	"a"	"sentence"

→ str.split("i");

"Th"	"s_"	"s_a_sentence"

str = "_abc _ _ ab _ cdabxyz"

→ str.split("ab");

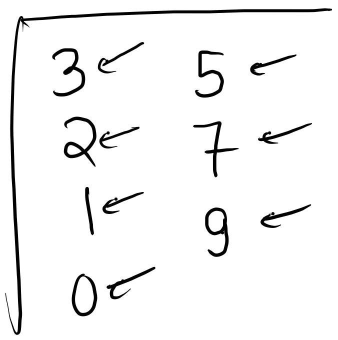
_	c_ _	_cd	xyz
0	1	2	3

Find Unique

str = "32120013579332";

ans = 7

which topic :- arrays as hashmap



boolean

0	1	2	3	4	5	6	7	8	9
T	T	T	T	F	T	F	T	F	T

Code

```java
public static int findUnique(String str) {
    boolean[] check = new boolean[10];
    int n = str.length();
    for (int i = 0; i < n; i++) {
        char ch = str.charAt(i);   // '5'
        int idx = ch - '0';
        check[idx] = true;
    }
    int count = 0;
    for (int i = 0; i < 10; i++) {
        if ( check[i] == true ) {
            count++;
        }
    }
    return count;
}
```

10 → `boolean[] check = new boolean[10];`

1 → `int n = str.length();`

$O(N)$ — first for loop

1 → `int count = 0;`

$O(10)$ — second for loop

$T.C = O(N+10)$

$\cong O(N)$

where, N is size of string

$S.C = O(12)$

$\cong O(1)$

Note:-

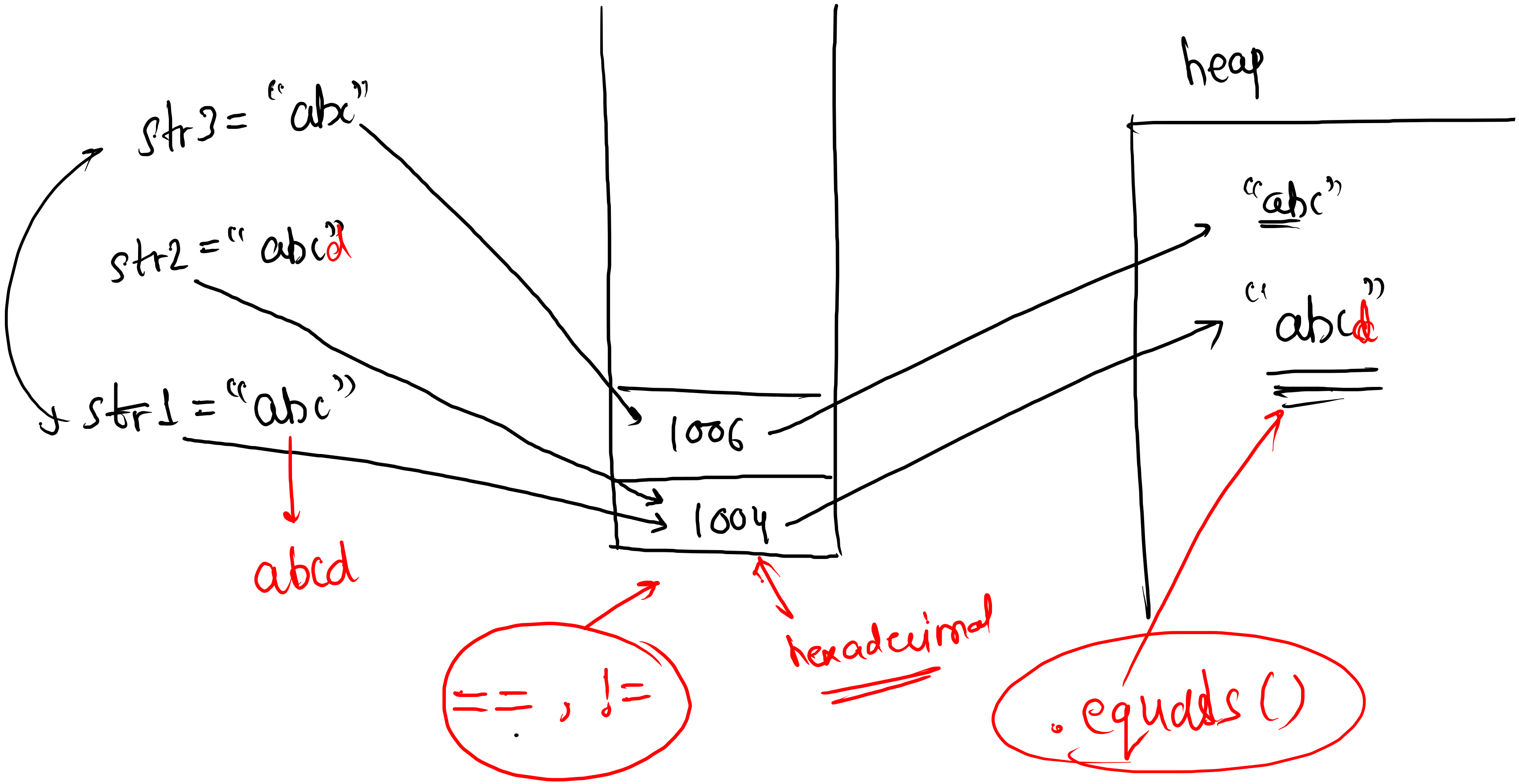
$\boxed{2 \times N, \quad \underline{N^2}}$

T.C ⇒ no. of operations
(how many times you have visited each element)

S.C ⇒ no. of memory addresses you have consumed.

⇒ String is immutable

2 level arch.



str3 = "abc"
str2 = "abcd"
str1 = "abc"
abcd
1006
1004
heap
"abc"
"abcd"
== , !=
hexadecimal
.equals()