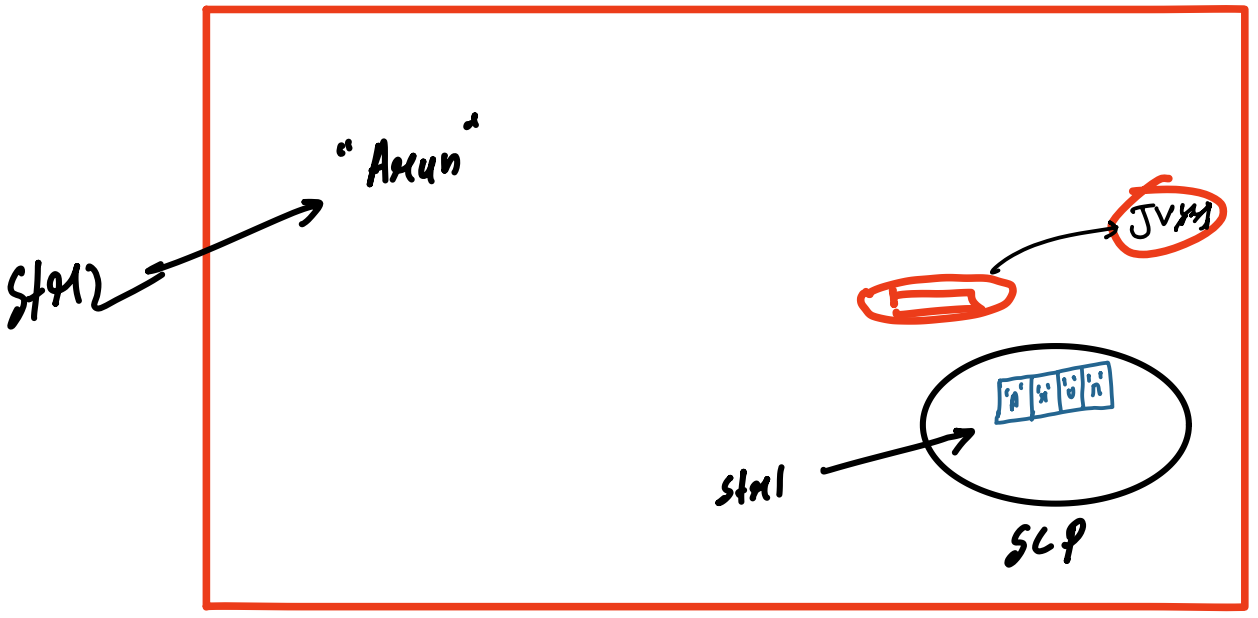
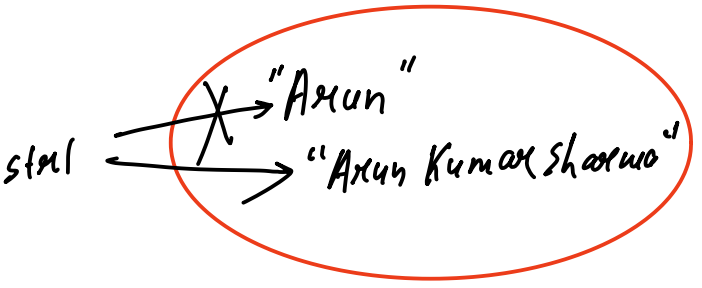
String



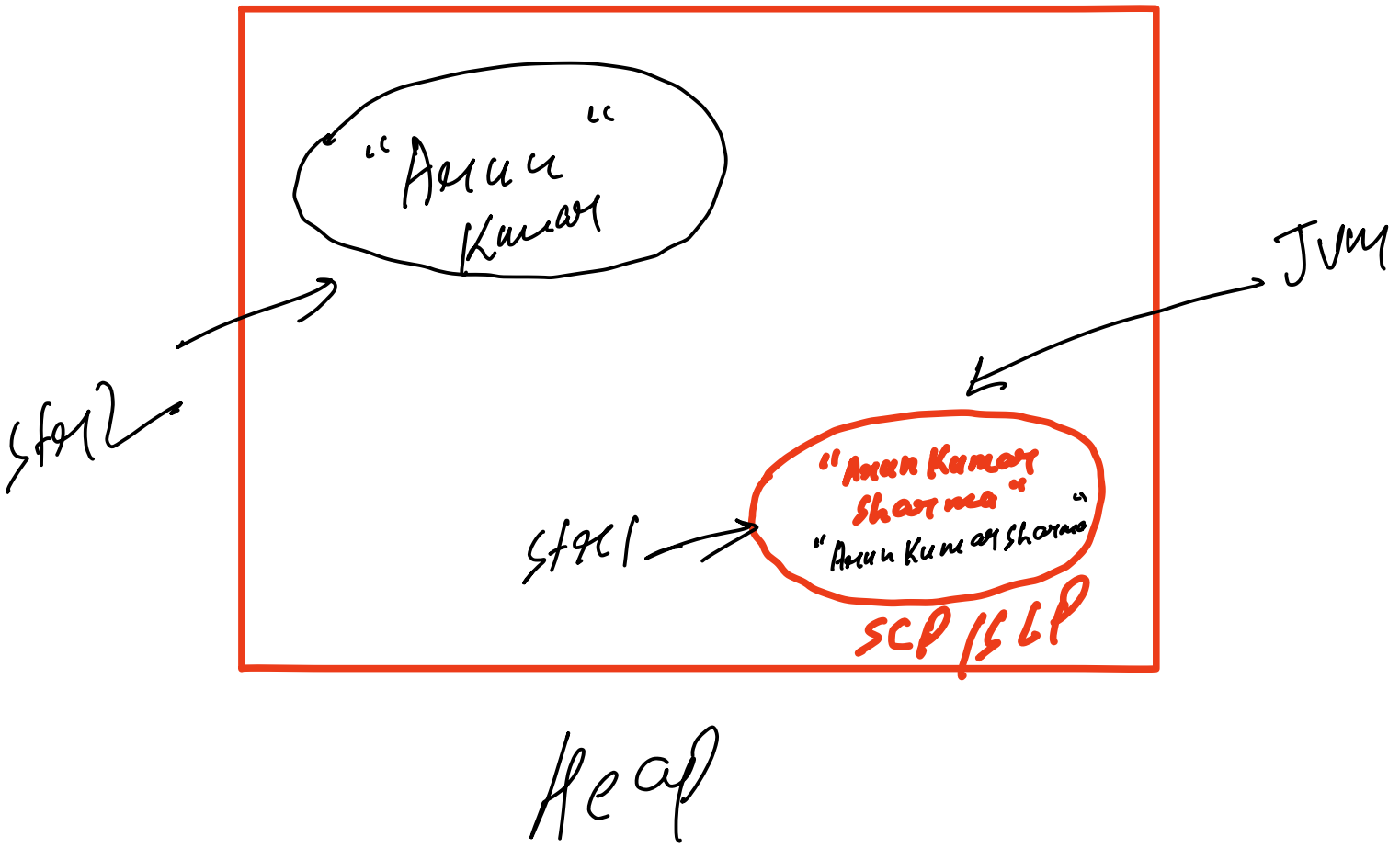
Heap Memory

arr.length;

str.length();

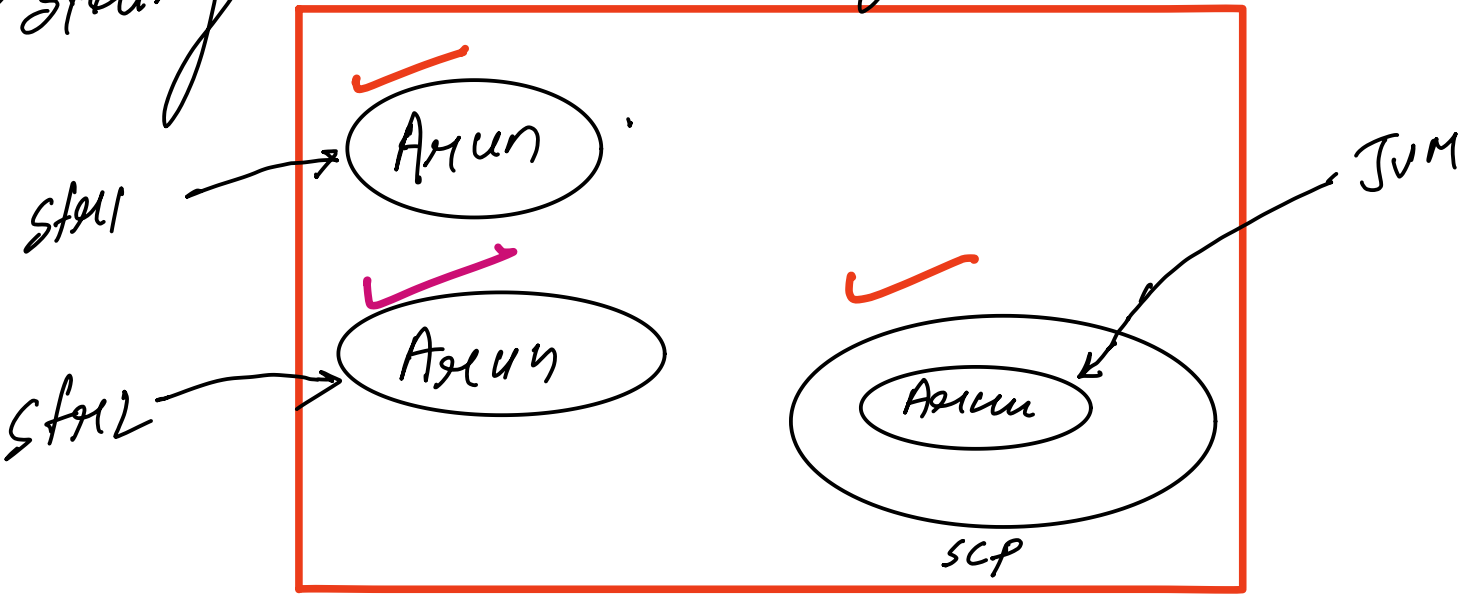


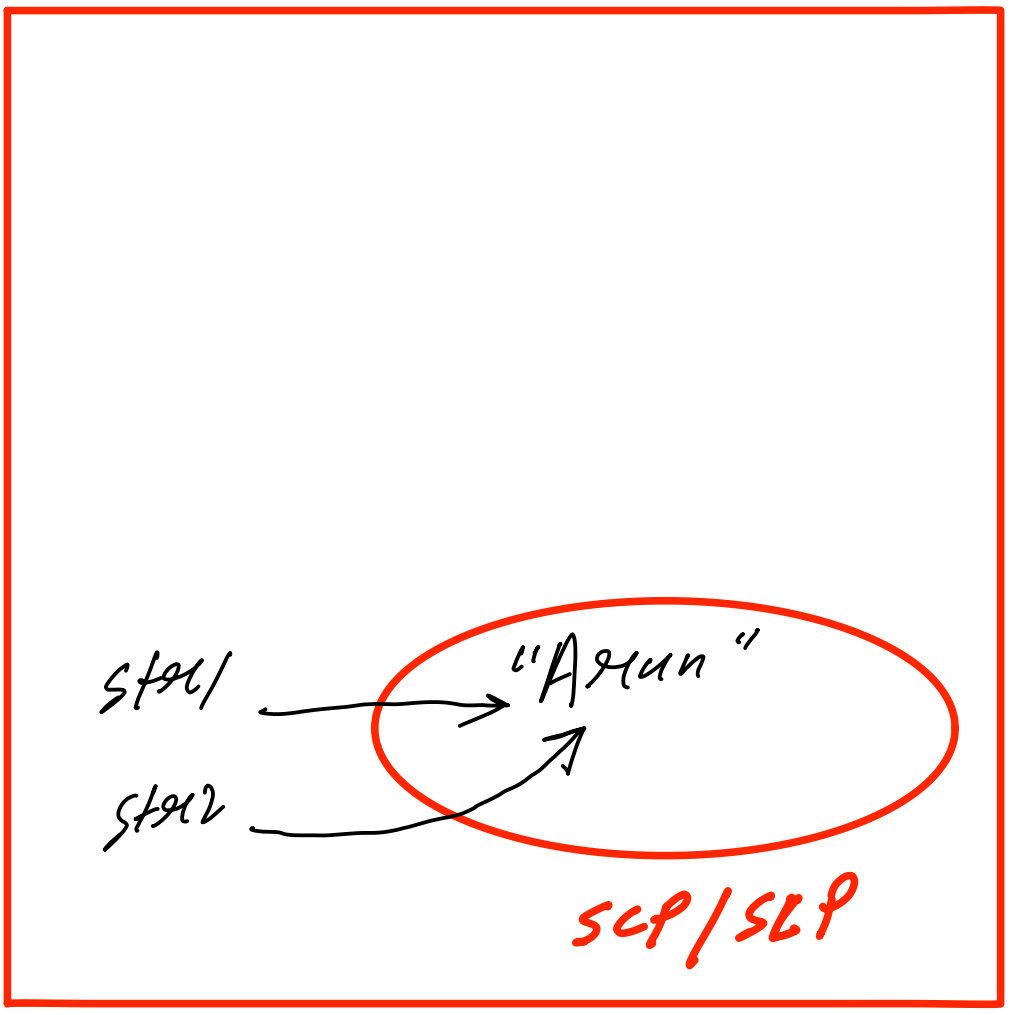
str1 + "Kumar Sharma";



Heap

```
String str1 = new String("Amit");  //2 obj
String str2 = new String("Amit");  //1
```





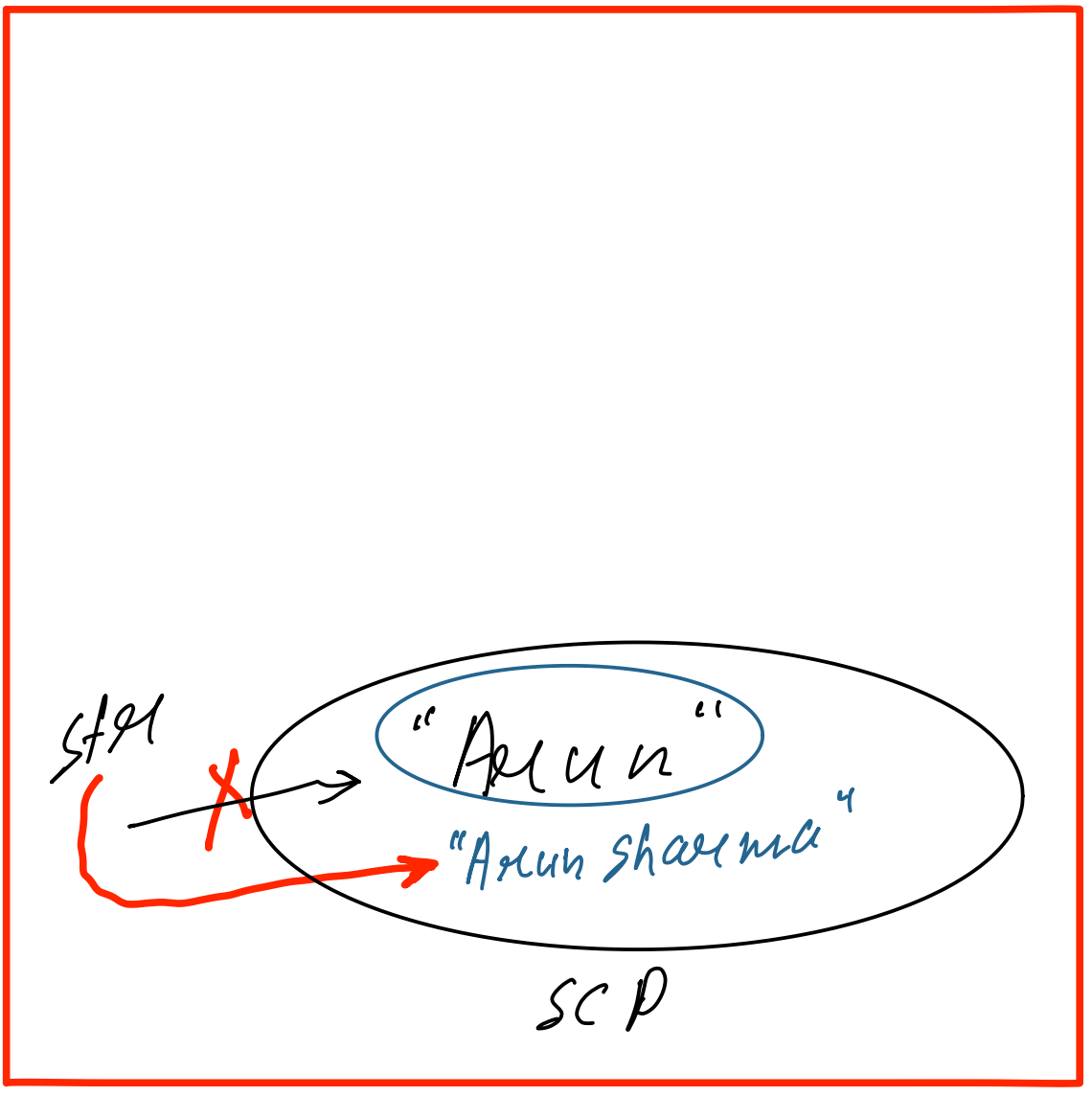
⇒ String str1 = "Arun";

⇒ String str2 = "Arun";

Why string objects are Immutable ?

* String is Not immutable, String objects are immutable

```
String str = "Arun";
s.o.p(str + "sharma");
str = str + "sharma";
```



str = str + "Sharma"

Q: Why String class is final ?

# String str = new String("Arun");
S.O.println(str);

String Methods

** (1) length()
(2) concat()
* (3) trim()
* (4) isEmpty()
* (5) equals
* (6) compareTo()
(7) subString()
* (8) replace()
* (9) contains()
(10) lastIndexOf()
** (11) charAt(i);
(12) startsWith()
(13) endsWith()
(14) toUpperCase()
(15) toLowerCase()
⇒ (16) split()

Substring

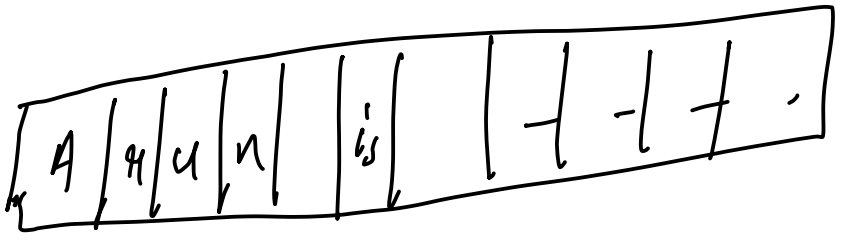
A
Ar
Aru
Arun
run
un
n

subsequence

A
Ar
Aru
Arun
Aun
An
Arun
Aun
Arn
An

String str = "Arun is teaching java";

char [] c = str.toCharArray();

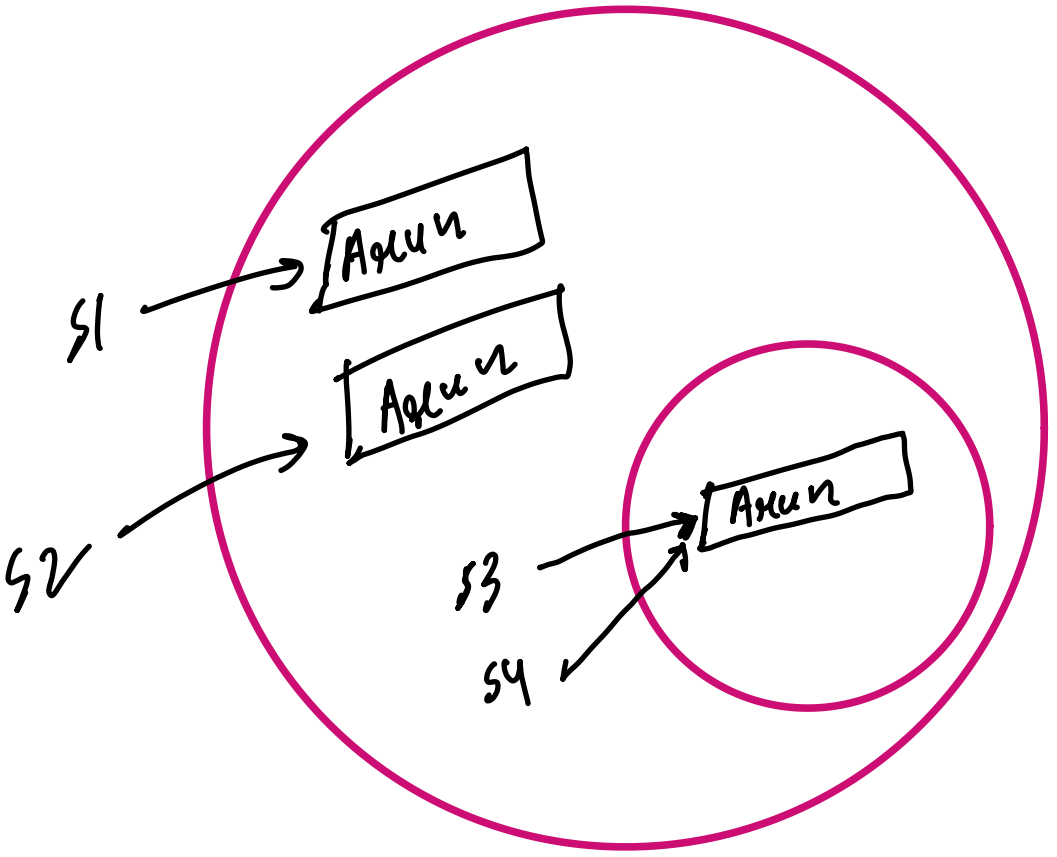


StringBuffer

Diff b/w equals() & ==

```
String s1 = new String("Arun");
String s2 = new String("Arun");
String s3 = "Arun";
String s4 = "Arun";

S.o.p(ln (s1.equals(s2)) // T
         (s1.equals(s3) // T
         (s1 == s2) // F
         (s1 == s3) // F
         (s3 == s4) // T
         (s3.equals(s4)) // T
```



equals() ⟶ Content

== ⟶ reference

Q: WAP to reverse the String

s = "Arun Kumar Sharma"

o/p = "amrahs ramuK nuRA"

str = "Arun"
↑

```
String res = "";
N = str.length();
for(i = N-1 ; i>=0 ; i--) {
        res = res + str.charAt(i);
}
Print res;
```

Soln 2: StringBuilder SB = new StringBuilder(str)

```
SB.reverse();
// S.o.p(SB);
S.o.p(SB.toString());
```

WAP to check weather a String is palindrome or Not

```
Str = "racecar";
```

racecar
↑ ↑

```
int i = 0
int j = N - 1
boolean ispalindrome = true;
while( i < j)
    if(str.charAt(i) != str.charAt(j)){
        ispalindrome = false
        break j
    }
    i++ ;
    j-- ;
}
```

```
if(isPalindrome){
    print("yes");
}
else{
    print("No");
}
```

2nd approach

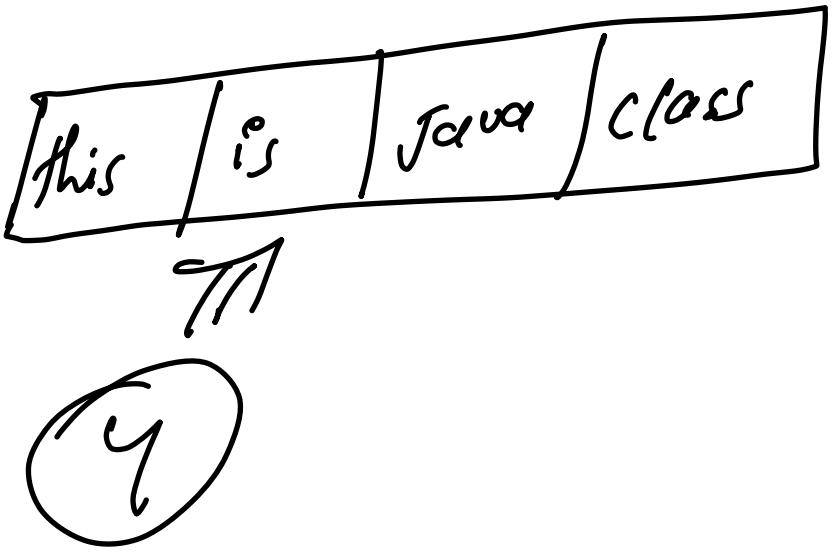
① str = "race car"
rev = "race car"

```
if(str.equals(rev){
    //
}
```

Count the no. of words in a String

String str = "This is Java class";

o/p = 4



String s3[] arr = str.split(" ");

s.o.println(arr.length);

Q. String str = "abcdbba";

a → 2

b → 3

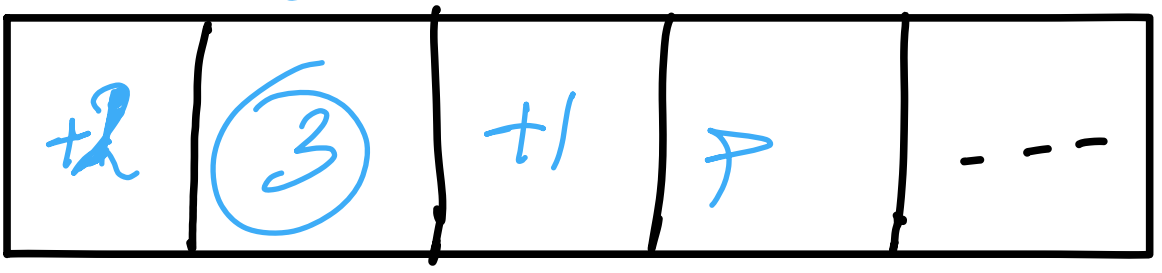
c → 1

d → 1

'a' - 97 ⇒ 0

b - 97 ⇒ 1

z - 97 = 25



(1+97) →

```
String str = "abbacddbq";
int N = str.length();
int[] arr = new int[26];
for(i=0; i<N; i++){
    char c = str.charAt(i); //a, b, b
    int index = c-97; //0, 1, 1
    arr[index] = arr[index] + 1
}
for(i=0; i<=25; i++){
    if(A[i] > max){
        index j = i
        max = A[i];
    }
}
```

print(char(i+97));

I/P ⇒ "aranp rnb"

↳ str.removeAll(" ", "");

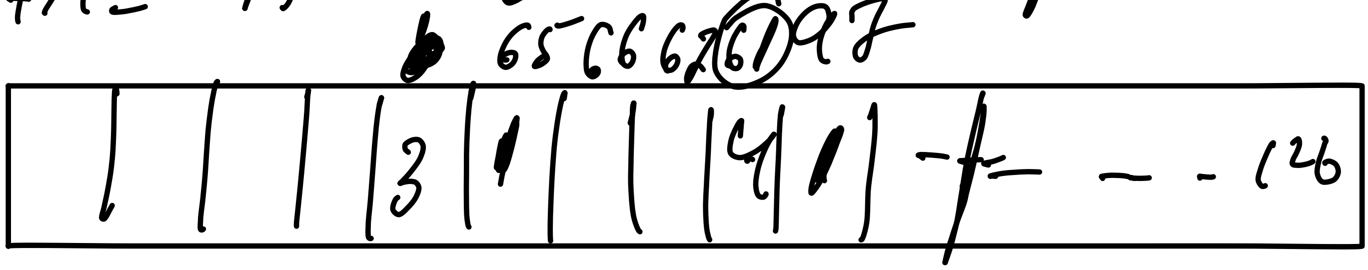
0 to 126

int[] arr = new int[126];

A → 65

a → 97

str = "AbcDZcDDDab"



string str = "abcdefg";

int N = str.length();

arr[126]

for(i=0; i<N; i++){

char ch = str.charAt(i)

arr[ch] = arr[ch]+1;

}

int max = -1;

char maxchar = ' ';

```
for(i=0; i<126; i++){
    if(arr[i] > max){
        max = arr[i];
        maxchar = (char)i;
    }
}
Print(maxchar);
```

WAP to find duplicate character in a string.

str = "abcbdad";

o/p = a b d ⇒ a → 2
b → 2
d → 3

```
psvm ( ) {
    string str = "abdcbabe"
    str = str.replaceAll("\\s", "");
    int[] arr = new arr[126];
    for(i=0; i<N; i++) {
        char ch = str.charAt(i);
        arr[ch] = arr[ch] + 1;
    }
    for( i=0; i<126; i++) {
        if(A[i] > 1) {
            Print ((char)i + "repeated" +
                arr[i] + "times")
        }
```