String

It is an object that represents a sequence of chars.

java.lang

Strings are immutable.

String → class → object → non-primitive data type

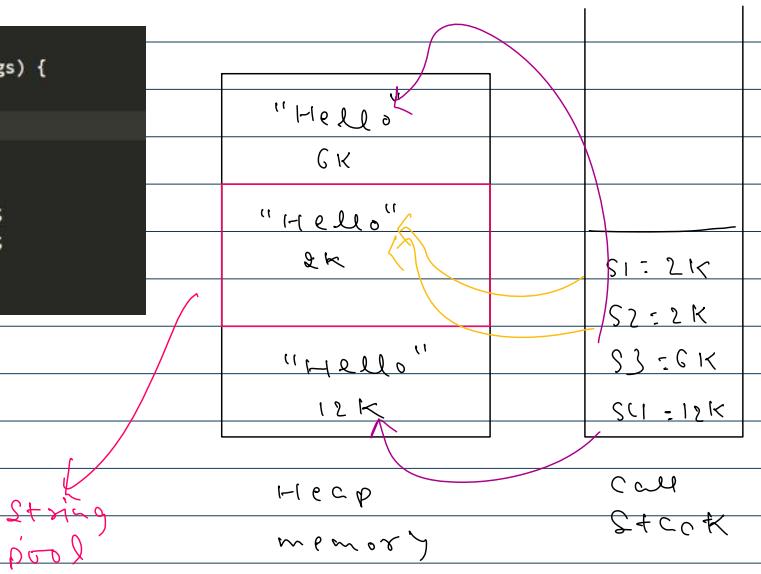
↓

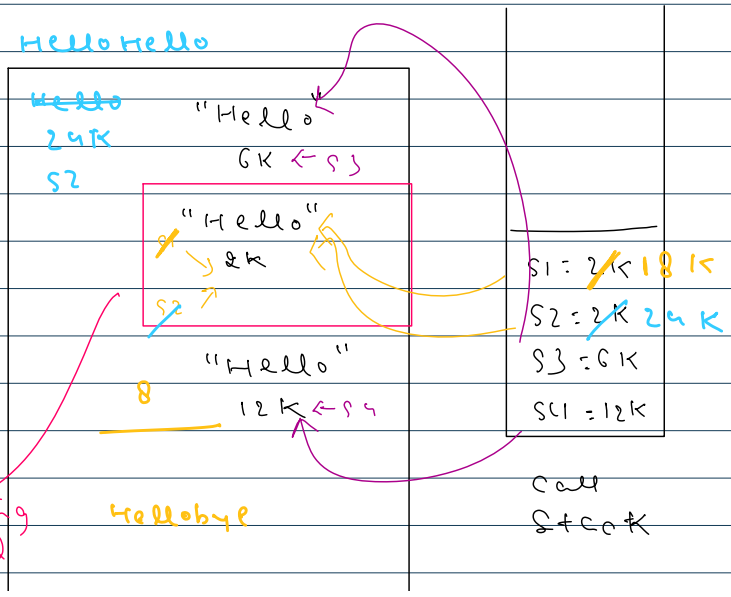
Heap memory

1K
"Hello"
S1

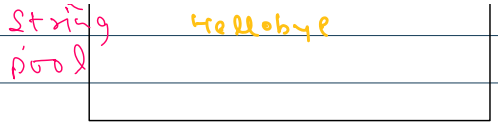
String pool (intern pool)

```java
public class Main {
    public static void main(String[] args) {

        String s1 = "Hello";
        String s2 = "Hello";

        String s3 = new String("Hello");
        String s4 = new String("Hello");
    }
}
```






```
S1 = S1 + "bye";
S2 = S2 + S3;
S3 = S3 + S4;
```



Stack

Heap memory

```
public class Main {
    public static void main(String[] args) {

        String s1 = "Hello";
        String s2 = "Hello";
        String s5 = "Akarsh";

        String s3 = new String("Hello");
        String s4 = new String("Hello");

        System.out.println(s1);

        System.out.println(s1 == s2);
        System.out.println(s3 == s4);
        System.out.println(s1 == s3);

        System.out.println(s5 == s2);

        int[] arr = new int[7];
        System.out.println(arr.length);
        System.out.println(s1.length());

        s1 = s1 + "bye";
        System.out.println(s1);

        s1 = s1.concat("bye");
        System.out.println(s1);

        String name = "Akarsh";

        name = name + "Jaiswal" + "lives in Noida";
        System.out.println(name);

        System.out.println("Hey" + 10 + 20 + "Bye");
        System.out.println("Hey" + (10 + 20) + "Bye");
        System.out.println(10 + 20 + "Hey" + "Bye");
        System.out.println("10" + "20" + "Hey" + "Bye");

    }
}
```

```
public class Main {
    public static void main(String[] args) {

        String s = "akarsh";

        System.out.println(s.charAt(3));
        System.out.println(s.charAt(s.length()-1));
    }
}
```

```
public class Main {
    public static void main(String[] args) {

        String s1 = "akarsh";
        String s2 = new String("akarsh");
        s2 = "akarsh";

        String s3 = "akar" + "sh";

        System.out.println(s1==s2);
        System.out.println(s3==s2);

        String s4 = "akar";// + new String("sh");
        System.out.println(s4==s2);
        System.out.println(s1.equals(s2));

    }
}
```

```
public class Main {

    public static void main(String[] args) {

        String s1 = "akarsh";
        String s2 = new String("akarsh");
        System.out.println(s1.equals(s2));
        System.out.println(equals(s1, s2));

    }

    public static boolean equals(String s1, String s2){

        if(s1 == s2){
            return true;
```

```
        }

        if(s1.length() != s2.length()){
            return false;
        }

        for(int i=0; i<s1.length(); i++){
            if(s1.charAt(i) != s2.charAt(i)){
                return false;
            }
        }
        return true;
    }
}
```

Shinchan shero

i > e => Shinchan

compareTo

Return type => 0 → same (s1 == s2)

-ve → s1 < s2

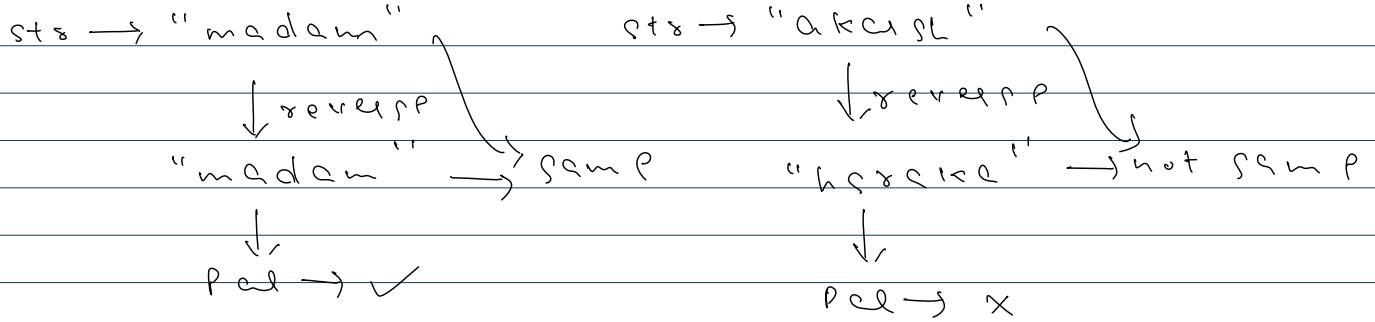
+ve → s1 > s2

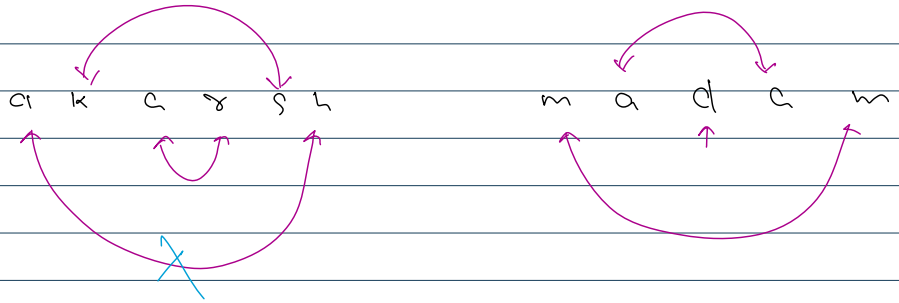
ankit ankit@ => s1.length() - s2.length() => 1

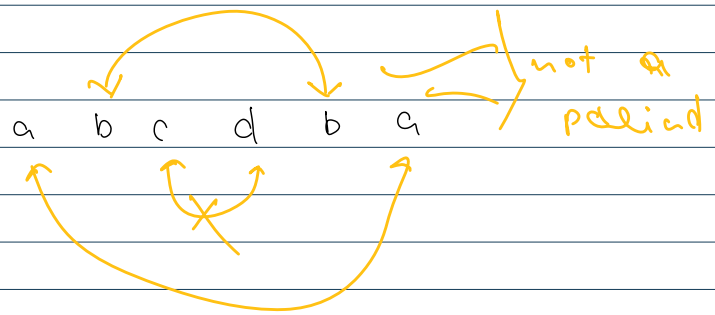
```
public class Main {

    public static int compareTo(String s1, String s2){
        if(s1 == s2){
            return 0;
        }
        int n = Math.min(s1.length(), s2.length());

        for(int i=0; i<n; i++){
            if(s1.charAt(i) != s2.charAt(i)){
                return s1.charAt(i) - s2.charAt(i);
            }
        }
```

```
        return s1.length() - s2.length();
    }

    public static void main(String[] args) {

        String s1 = "Shinchan";
        String s2 = "Sheero";
        System.out.println(s1.compareTo(s2));
        System.out.println(compareTo(s1, s2));

        s1 = "ankit";
        s2 = "ankitan";
        System.out.println(s1.compareTo(s2));
        System.out.println(compareTo(s1, s2));

        s1 = "ankit";
        s2 = "ankit";
        System.out.println(s1.compareTo(s2));
        System.out.println(compareTo(s1, s2));

        s1 = "Sheero";
        s2 = "Shinchan";
        System.out.println(s1.compareTo(s2));
        System.out.println(compareTo(s1, s2));

        s1 = "2";
        s2 = "136";
        System.out.println(s1.compareTo(s2));
        System.out.println(compareTo(s1, s2));

    }

}
```

Check if a string is palindrome








```
public class Main {

    public static void main(String[] args) {

        String s = "madam";
        System.out.println(isPalindrome(s));

        s = "akarsh";
        System.out.println(isPalindrome(s));

        s = "abcdba";
        System.out.println(isPalindrome(s));
    }

    public static boolean isPalindrome(String s){
        int i = 0;
        int j = s.length() - 1;

        while(i < j){
            if(s.charAt(i) != s.charAt(j)){
                return false;
            }
            i++;
            j--;
        }
        return true;
    }

}
```

Substring

It is a contiguous sequence of characters in a string

akarsh

a	k	a	r	s h
a k	k a	a r	r s	s h
a k a	k a r	a r s	r s h	
a k a r	k a r s	a r s h		
a k a r s	k a r s h			
a k a r s h				

substring

extract a portion of the string.

```
substring(start Index);
substring(start Index, end Index);
```

inclusive exclusive

```
public class Main {

    public static void main(String[] args) {

        String s = "akarsh";
        System.out.println(s.substring(1));
        System.out.println(s.substring(1, 4));
        // System.out.println(s.substring(1, 10));

    }

}
```

Print all substrings

0-1 a 1-2 k 2-3 a 3-4 r 4-5 s 5-6 h

0-2 ak 1-3 ka 2-4 ar 3-5 rs 4-6 sh

0-3 aka 1-4 kar 2-5 ars 3-6 rsh

0-4 akar 1-5 kars 2-6 arsh

0-5 akars 1-6 karsh

0-6 akarsh

```
for(int i:0 ; i< n ; i++ )
    }
      for (int j:i+1 ; j<=n ;j++ ) {
          substring(i,j) ;
    } }
```

[0 - n-1] => [0,1,2,3,4,5]

[0 - n] => [0,1,2,3,4,5,6]

```
public class Main {

    public static void main(String[] args) {

        String s = "akarsh";
        printAllSubstrings(s);

    }

    public static void printAllSubstrings(String s){

        for(int i=0; i<s.length(); i++){
            for(int j=i+1; j<=s.length(); j++){
                System.out.println(s.substring(i, j));
            }
            System.out.println("_____");
        }

    }

}
```

Character frequency

str → "character"

lowercase string

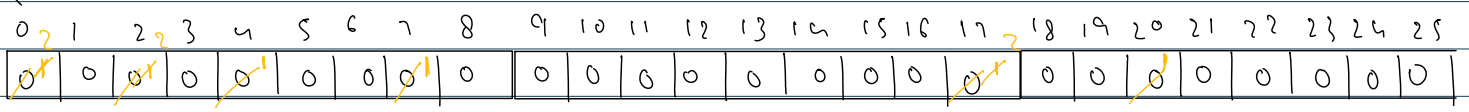
↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑

c → 99 ? 97

c → 99
a → 97
b → 98

- 97

c - 2
h - 1
a - 2
r - 2
t - 1
e - 1




```
public class Main {

    public static void main(String[] args) {

        String s = "akarsh";

        int[] freq = new int[26];
        for(int i=0; i<s.length(); i++){
            int idx = s.charAt(i) - 97;
            freq[idx] = freq[idx] + 1;
        }

        for(int i=0; i<freq.length; i++){
            System.out.print(freq[i]+" ");
        }
        System.out.println();

        for(int i=0; i<freq.length; i++){
            char ch = (char) (i+ 97);
            if(freq[i] > 0){
                System.out.println(ch + " -> " + freq[i]);
            }
        }

    }

}
```