String

char array = mutable String = immutable

String str = "sgfghi"

n = str.length()

char c = str.charAt(i)

str[i] = 'z' x not allowed { < get ✓ set x

int x = str.indexOf("fg") { works left to right. first occurrence }

int y = str.lastIndexOf("fg")

Substring

```
for (int i=0 ; i < s.length() ; i++)
{
    for( int j = i; j<=s.length() ; j++)  sop ln (s.substring(i,j
```

delete at i

```
String P1 = S.substring (0, i)
String P2 = S.substring (i+1)
sop ln (P1 + P2)
```

str.startsWith("ab)
str.endsWith ("AB") } case sensitive ✓ returns boolean
str.split.

① String cannot append ∴ harbaar fuddu String kahi ka new String bnata hai

② String bahoot mota hai. kutta bahooot space khata hai

eg: 2222222222 < int = 4 byte
String = 20 byte

2 byte = 16 bits = 2^{16} permutations (store kr skta hai)

If you want to store book on PC

'1 Page = 500×4×2 = 4 KB

1 book = 4000 × 500 = 2 MB

500 book = 500 × 2 = 1 GB

thus bahoot space

'if hum ek word ko store kr le and reoccure hone pe address de de string ki ~~jaha~~ jagah to km space use hoga

Thus if 26 letter store kre and baaki jagah address bhej de to aur kaam space use hogi (= = Trie DS)

(==) check adders

(new) string pool k bhar and heap k andar

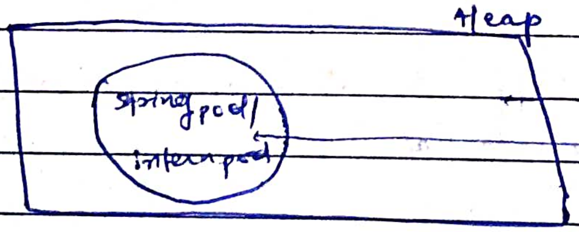
(.equals()) check content

create String

= new String("PPPP")

(store address)

= "PPPP"



ll2 = ll2.intern();

	String	StringBuilder
create	String str = "hello" String str = new String("hello")	String sb = new StringBuilder("hello
print	SOP(str)	SOP(sb)
charAt	str.CharAt(i)	Str.CharAt(i)
Substring	str.substring(i)	str.substring(i)

Add ~~insert~~
insert
	X	sb.append(" ")
		sb.insert(index, " ") (char or string)

update		sb.setCharAt(index, ' ')

remove		sb.deleteCharAt(idx)
		sb.delete(including, excluding)
		sb.toString()

palindrome

```
int start=0, end = s.length()
while (start <end)
{      if (s.charAt(start) != s.charAt(end)) return false;
       start++  end--
}
return true
```

print palindromic substring

Brute force :

```
for (int s=0  s< str.length()  s++)
{
    for (e=s+1 ; e<=str.length() )
    {
        String ss = str.substring(s,e)
        if( isPalindrome (ss) )
              SOPln(ss)
    }
}
```

Bhaiya method :

```
solve ( String str , int L , int R )
{
    while [ str.charAt (L) = str.charAt(R) && L>=0
             && R < str.length() ]
    {
        sopln ( str.substring (L, R+1))
        L --        R++
    }
}

palin ( String str )
{
    for (int c=0 , c < str. length () ; c++ )
    {
        solve (str , c, c )
        solve ( str , c, c+1)
    }
}
```