

# \* Strings

## # declaration.

① `string str;`  $\rightarrow$  s no. of times n  
`string str('s', 'n');`

② `string str = "ApriKaksha";`

## # input

① with spaces

`getline (cin, str);`

② without space

`cin`

$s1[i] \rightarrow$  ye bhi chel jaega.

Page No.

Date :

## # append

```
string s1 = "fam"  
string s2 = "ily";
```

```
s1.append(s2);  $\rightarrow$  OR  $s1 + s2$ ;
```

## # clear $\rightarrow$ puri string wala dega

```
string abc = "sda f g i e j k l m n o p q r s t";  
abc.clear();
```

## # compare

```
string s1 = "abc";  
string s2 = "xyz";
```

```
cout << s2.compare(s1) < endl;
```

output  
23.

## # check if empty or not

```
s1.empty();
```

## # Erase function

```
s1.erase(3, 3);
```

is index se shuru hoga

yahan tak hoga



# to find

`sl.find("com")` → give me index.

# insert

index no.

`sl.insert(2, "lol");`

# length

`sl.size()` OR `sl.length()`

# to get a substring

`sl.substr(1)`

`string s = sl.substr(6, 4);`

position

length

! ko change  
based  
ke seare  
string s  
mein  
copy  
ho jenge.

# ~~string~~ string to integer

`string sl = "786";`

`int x = stoi(sl);`

string to int

# integer to string

`int x = 786`

`sl = to_string(x);`



# // strings

\* strings with comparator.

~~sort~~

```
string str[100];
```

```
for (int i = n; i < n; i++)
```

```
getline(cin, str[i]);
```

sort(str, str+n); → for ~~lexographic~~ lexicographic comparison

OR

sort(str, str+n, comp);

~~string~~ comparison

```
bool comp (string a, string b)
```

```
{
```

```
return a.length() < b.length();
```

length  
wise  
comparison

```
}
```