

Lecture - 19

String
array

"paridhi"

```
String string-name;
cin >> string-name;
```

0	1	2	3	4	5	6
P	a	r	i	d	h	i

string-name[2] = 'r'

S = "Rohit"

M = "Mohit"

$$\textcircled{1} \quad T = S + M$$

output : RohitMohit

$$\textcircled{2} \quad T = s.append(M)$$

output = RohitMohit

S = "10";

T = '20';

S = S + T

Output : 1020

string-name.size();



Output = 7

S = "CoderArmy"

① S.pushback('s'); → Add s at last

S = CoderArmys

② S = S + 's';
S.pop_back();
↳ delete last letter

Sort:

sort(str.begin(), str.end())

str before sort: abchb

str after sort: abbch.

→ Code:

```
String str;  
cin >> str; // Input: paridhi  
cout << str; // Output: paridhi  
cout << str.size(); // Output: 7  
cout << str[1]; // Output: a
```

→ String str1 = "paridhi singh"
cout << str1; // paridhi singh
cout << str1.size(); // 13

→ String str2;
cin >> str2; // paridhi singh
cout << str2; // paridhi

In input if we give space, tab then it can be
terminate.

If you want to consider space also then
use getline :

```
String str;
getline (cin, str);    // Input : Paridhi singh
cout << str;           // Output : paridhi singh
```

```
String str1, str2;
getline (cin, str1);   // Input : Paridhi singh
getline (cin, str2);   // Input : Riya Rani
cout << str1 << " " << str2; // Output : paridhi singh Riya Rani
```

→ string str : "Ram is "a" boy";
↳ Given Error

→ string str : "Ram is \"a\" boy";
↳ Output : Ram is "a" boy.

| → ignore next character and print as written.

→ string str : " Ram is \|a\| boy";
↳ Output : Ram is \a boy.

→ Char c = 'd' + 2;
cout << c; // Output : f

* Sort by Comparing ASCII Values

```

for (int i=0; i<n-1; i++) {
    for (int j=1; j<n; j++) {
        if (s[j] < s[j-1]) {
            swap(s[j], s[j-1]);
        }
    }
}

```

~~O(N²)~~

* Approach - 2

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	
∅	0	0	∅	0	0	∅	∅	0	0	0	0	0	0	∅	0	0	∅	0	∅	∅	0	0	0	0	∅	0
r	1	1	r	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	10	10	11	11	12	12	
s	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3

$s = \text{paridhisinghRiyaRani}$

↳ Assume all are small

P A R I D H I S I N G H R I Y A R A N I

a a a d g h i i i e e n n p r r r s y
↳ sorted.

Code:

```

String s;
cin >> s;
n = s.size();
int count[26];
for (int i=0; i<26; i++)
    count[i] = 0;

```

```

for (int i=0; i<n; i++) {
    int index = s[i] - 'a';
    Count[index]++;
}

for (int i=0; i<26; i++) {
    for (int j=0; j<Count[i]; j++) {
        char c = 'a' + i;
        cout << c;
    }
}

return 0;
}

```

* Reverse a string (GFG) :

P a r i d h i
 ↑ ↑
 start end

Start < end

swap (P, i)

start ++;

end --;

i a r i d h P
 ↑ ↑
 start end

Start < end

swap (a, P)

start ++ ; end -- ;

i h r i d a P
↑ ↑
start end

start < end

swap (r, d)

start ++;

end --;

i b d i r a P
↑
start
end

start < end (false)

Code:

```
int start = 0;  
int end = str.size() - 1;  
  
while (start < end) {  
    swap (str[start], str[end]);  
    start++;  
    end--;  
}  
return str;
```

* Anagram of string :

Example:

$s_1 = cd\textcircled{d}g\textcircled{k}$

$s_2 = gcd$

Output = 2

From s_1 : one d is deleted }
 one k is deleted }
 total 2 operation.

Example:

$s_1 = abcd\textcircled{d}\textcircled{k}\textcircled{f}$

$s_2 = abcd\textcircled{h}$

From s_1 : one d is deleted }
 one k is deleted }
 one f is deleted } 4
 s_2 : one h is deleted }

(S1)

0	1	2	3	4	5	6	7	8	9	10
0	∅	∅	∅	0	∅	0	0	0	0	∅
1	1	1	X ₂	1						1

(S2)

0	1	2	3	4	5	6	7	8	9	10
∅	∅	∅	∅	0	0	0	∅	0	0	0
1	1	1	1				1			

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

0

0

0

1

0

0

1

0

1

0

0

1

Sum = 4

* palindrome string:

Example = a b b a return 1

 True

a b c return 0.

 False

```
int start = 0, end = s.size() - 1;
while (start <= end) {
    if (s[start] != s[end]) {
        return 0;
    start++;
    end--;
}
return 1;
}
```

2nd Approach:

- Reverse string and store in another string.
- Match each character.

```
string k; k = s;
reverse(k.begin(), k.end());
if (k == s)
    return 1;
else
    return 0;
```

* Minimum Number of Flips :

$$S = "001"$$

Alternate me alg alg character hona chahiye.

0 ke bagal se 1 hoga

1 ke bagal me 0 hoga.

Hum ek-ek character ko idhar udhar kar sakte
hai to humko kitna flip flop kar sakte hai.

Minimum flip return karha hai.

$$\rightarrow \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{matrix}$$

1 0101 1 flip

return 1.

$$\rightarrow \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{matrix}$$

$\downarrow \downarrow \times \times \times \times \times \downarrow \times$

8 flips 1 0 1 0 1 0 1 0 1 0

$$\rightarrow \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{matrix}$$

$\downarrow \times \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \times \downarrow$

2 Flips 0 1 0 1 0 1 0 1 0 1

(return 2)

* Merge two string :

S1 : Hello

S2 : Bye

Output : HBeylelD

String S3;

while (i < s1.size() && j < s2.size()) {

 S3 = S3 + s1[i];

 S3 = S3 + s2[j];

 i++;

 j++;

}

while (i < s1.size()) {

 S3 = S3 + s1[i];

 i++;

}

while (j < s2.size()) {

 S3 = S3 + s2[j];

 j++;

}

* To lower case :

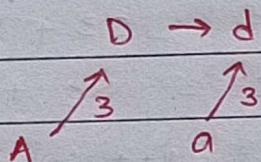
$s1 = "LOVELY"$

Output : lovely

$s2 = "Here"$

Output : here

→ code :



$\text{char } c = s[i] - 'A'$

$c = 'd' + s[i] - 'A';$

if ($s[i] \geq 'A'$ & $s[i] \leq 'Z'$)

* sum of two large Number :

$x = "25"$

$y = "23"$

Output = 48

$x = "2500"$

$y = "27"$

Output = 2527

int num = "7" - '0' = 7

int num = '0' - '0' = 0

carry = 0, num = 0

↳ sum Niklega.

Code:

```
int i = x.size() - 1;
```

```
int j = y.size() - 1;
```

```
string str = " ";
```

```
int num, rem, carry = 0;
```

```
char c;
```

```
while (i >= 0 && j >= 0) {
```

```
    int num1 = x[i] - '0';
```

```
    int num2 = y[j] - '0';
```

```
    num = num1 + num2 + carry;
```

```
    carry = num / 10;
```

```
    rem = num % 10;
```

```
    c = '0' + rem;
```

```
    str = str + c;
```

```
    j--;
```

```
    i--;
```

```
}
```

```
while (i >= 0) {
```

```
    int num1 = x[i] - '0';
```

```
    num = num1 + carry;
```

```
    carry = num / 10;
```

```
    rem = num % 10;
```

```
    c = '0' + rem;
```

```
    str = str + c;
```

i--;

}

while (j >= 0) {

int num1 = y[j] - '0';

num = num1 + carry;

carry = num / 10;

rem = num % 10;

c = '0' + rem;

str = str + c;

j--;

}

if (carry == 1)

str = str + '1';

int size = str.size() - 1;

while (size > 0 && str[size] == '0') {

str.pop_back(); size--;

}

reverse (str.begin(), str.end());

return str;

}

}