


Char arrays

char arr[10];

→ Symbols

→ 'A' → 'Z'

→ 'a' → 'z'

→ 0 → 9



Char → 1 byte → 8 bit

$2^8 \rightarrow \underline{256 \text{ char}}$

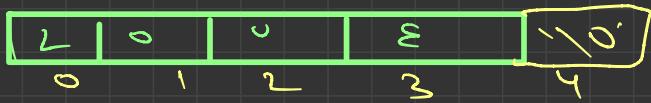
How to take input?

`Cin >> ch;` // Address

How to print?

`Cout << ch;` // Address

Char ch[5];



IP \Rightarrow LOUE \Leftarrow Sequence of characters
↓ ↓ ↓ ↓
0 1 2 3

* Null string \rightarrow termination of string

' \0 ' \rightarrow ASCII value \rightarrow 0

How to verify null char ?

Char temp = ch[6];
int value = (int)temp;
cout << value;

ch[10] =
A D I T Y A '\0' | | |
0 1 2 3 4 5 6 7 8 9

cin >> ch; // ADITYA

cout << ch \Rightarrow ADITYA

NULL char

0 output

int temp = ch[6] \Rightarrow int val = (int)temp;

```
char ch[100];
```

```
cin >> ch; // ADITYA JAIN  
cout << ch => ADITYA
```

* Clear away me input left to space like
bad sink Jayega

delimiter

grvc
jayega {
'n' → newline
't' → tab
'-' → space

How to solve this?

char ch[100];
cin.getline(ch, 100); } ← newline
{
 enter
 yeha jukega

length of string ?

Length

A	/	S		1		T		4		A		'\0'		
0	1	2	3	4	5	6								

↳
↳ 'j' is null char
↳ jayega sink
↳ jayga

```
int leng = findlength(ch, 100);
```

```
int findLength ( ch , size ) {  
    for ( int i = 0 ; i < size ; i ++ )  
        {  
            if ( ch [ i ] == '\0' )  
                break ;  
            else  
                count ++ ;  
        }  
    return count ;  
}
```

Predefined Stl

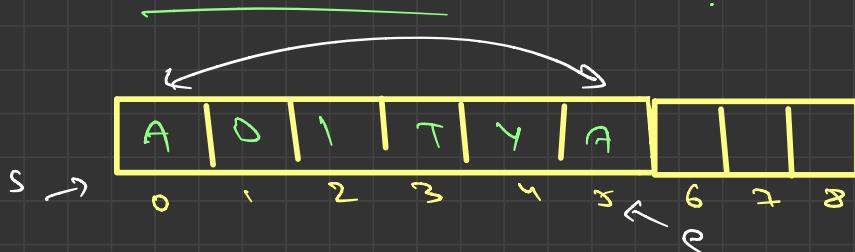
~~if include < string.h >~~

strlen (ch) ;

using
while
loop

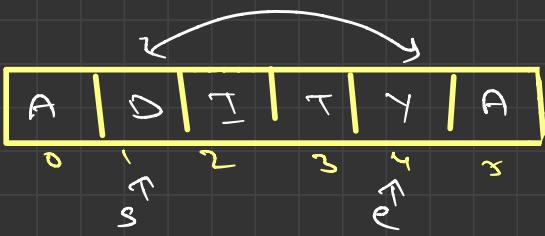
```
int len = 0 ;  
int index = 0 ;  
while ( ch [ index ] != '\0' ) {  
    index ++ ;  
}  
return index ;
```

How to reverse string? → using 2 pointers

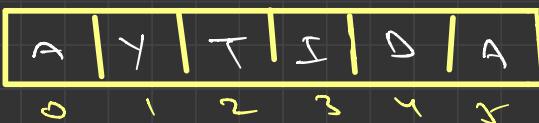
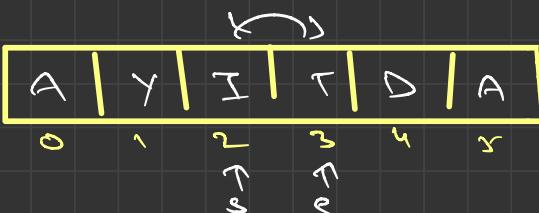


↓
swap

↓
make sure
we pass
the length



↓
don't pass
full array
size



⇒ O/P
=====

Lowercase \rightarrow uppercase

'a' \rightarrow 'A'

'a' \rightarrow ~~'z'~~ - ~~'A'~~ + 65 \rightarrow 'A'

formula

L.C. - 'a' + 'A' \rightarrow uppercase

formula

lc \rightarrow uc

ch - 'a' + 'A' \rightarrow lowercase

```
int index = 0;
```

```
while (ch[index] != '\0') {
```

```
if (ch[index] >= 'a' && ch[index] <= 'z')
```

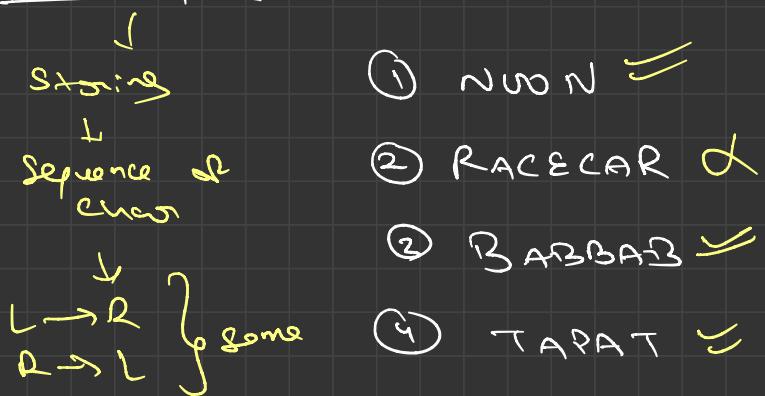
```
{
```

$$ch[index] = ch[index] - 'a' + 'A';$$

```
}
```

```
index += 1;
```

Check Palindrome



ch ⇒

B	A	B	B	A	B
---	---	---	---	---	---

 s → 0 1 2 3 4 5 re

while (s <= e)
 {

 if (ch[s] != ch[e])

 return false;

}

else

 { s++;

 e--;

}

return true;

String → Runtime change
size dynamically

Creation

String name;
← datatype ← var-name

I/O ⇒ cin >> name ;

O/P ⇒ cout << name ;

Printing first char name[0] ;

int index = 0 ;
while (name[index] != '\0')
{
 cout << name[index] ;
 index++;
}
cout << name[6] ;
int val = (int) name[6] ;
cout << val ; ⇒ 0

upto we don't get null char

Checking null char

How to store full name with space?

```
getline ( cin , name ) ;
```

Predefined Functions → cplusplus.com / String

String name ;

```
cin >> name ;
```

Empty string

```
String temp = " " ;
```

name.length() → length of String

temp.empty() → String is empty or not

name.at(0) → char at pos 0

name.front() → front char

name.back() → back char

String.append(string) → append a string

name.push_back('R') → insert at last index

name.pop_back() → delete from last

~~erase~~

T	H	I	S		I	S		A		C	A	R	‘\0’	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

desc. `erase(4, 3);`

`String name = "Aditya Jain";`

`String middle = "kumar";`

`name.insert(6, middle)`

→ Aditya kumar Jain

`String name = "Aditya kumar Jain";`

`String name2 = "kumar";`

`if (name.find(name2) == name::npos)`

`cout << "not found";`

`else`

`cout << "found";`

without using
this it will give
starting index

String::npos

It holds
the max
possible value
for an ele
of container
give & it is
used to denote the
end of the
string or sub-string

Compare

String ch1 = 'Bobbaa' ;

String ch2 = 'Bobbaa' ;

if (ch1.compare(str2) == 0)

{ cout << "match" ;

}

else

cout << "not match" ;

desc. substr(\overrightarrow{i} , \overrightarrow{j})
from which index
upto how many