Day-50
Strings

=) We have diff. types of datatypes to store different types of data.

=) int for numbers.
char for characters.
bool for boolean.

=) Now, if we want to store a word then we can use char too. as —

Ex: Apple

A	P	P	L	E

=) For taking i/p from user —

char arr[10];
for(i=0; i< 10;i++)
  cin >> arr[i];

=) Sometimes, user enter ~~enter~~ a ~~stru~~ word less than 10 then —

=) we can use —

char arr[6k];
~~for(i=0; i<~~
cin >> arr;

→ 1B → 1B

A	P	P	L	E	\0	\0	\0	\0	\0

\0 = Null char

=) It will used to terminate the ~~th~~ pointer when print the arr.

=) Chan takes 1 Bytes of space.

=) In this method, there is a problem that we have to give the size of the array.

:) So, to solve this problem, we use String, that uses dynamic memory allocation.

```
String S;
cih >> S;
cout << S;
```

0	1	2	3	4	5

Rohit ⟶ | R | o | h | i | t | \0 |

↳ internal implemen.

=) String S = " Rohit ";

=) If we enter a string like this —

Rohit   Negi

0	1	2	3	4	5	6	7	8	9
R	o	h	i	t		N	e	g	i

↳ O/p → Rohit

=) cin stops the reading whenever ~~the~~ space, Tab or enter comes.

⇒ Instead we will use getline(cin, s);

⇒ Size of string → s.size();

⇒ Combining string
    s1 = "Rohit" , s2 = "Mohit";
    String s3 = s1 + s2
                = "Rohit" + "Mohit"
                = "RohitMohit"

⇒ S. push_back('P');

⇒ S. pop_back ();

⇒ If we want to print a line like this -
    Rohit is a "good" boy.

⇒ String S = "Rohit Negi is a "good" boy";
            It will only print upto this   Here it will give
                                                  error

⇒ To solve this problem, we use -
    Escape char. (\)
    So,
    String S = "Rohit Negi is a \"good\" boy";

* Reverse a string
    start = 0 , end = s. size() - 1;
    while(start < end){
        swap (s[start], s[end]);
        start++ , end--;
    }

*

Calculate size

S = " Rohit";

R	o	h	i	t	\0

size = 0;

while (s[size] != '\0'){

     size++;

}