Strings

Something enclosed in the " ".

"Strings"

① Chracter Arrays

└ cStrings(chracter Arrays)

we can use them to store the strings.

char str = {'a', 'b', 'c', '\0'}

will convert characters into a ↲
valid string (though char already prints them as the string but '\0' makes the string valid)

like → "abc"

strlen(str) → ③ — will ignore '\0'.

char str[] = "Hello" // String literals

Internally Hello will be stored in contiguous locations

0	1	2	3	4	5
H	E	L	L	O	\0

Input & Output

char str [100]

cin >>str; → Hello world

cout<<str; → Hello // will ignore anything after the space

So, the solution is:

cin.getline(str, len, delim?)

str array

length of input

delimiter (limit the string using specific symbols/anything (By default Enter))

Example

```
cin.getline(str, 100, $)        Hello world$ I am
cout << str → [Hello world]                    Here
```

Output

```
for(i=0 ; i< str[i] = '\0' ; i++){
    cout << str[i];
}
```

Counting the length as well

```
length=0
for(i=0; i<str[i]='\0'; i++){
    length ++;
}
```

for Helloworld → 11

2) → Strings

Inbuilt strings. like strings already exist as the class and then we use strings in the code as objects.

```
string str = "String".
```

→ Dynamic In Nature (means values can be changed, can be resized.) which is also the limitation of character arrays.

→ supports the operators as well.

```
str3 = str1 + str2 (Concatination)
str1 == str2
str1 > str2     [Comparison for the lecigraplical checking]
str1 < str2
```

str.length() → length of strings.

```
                              Hello world
      cin >> str
                            Hello Ignored
      cout <<str
      getline(cin,str)      Hello world
      cout str              Hello world
```

Loops

```
for(i=0 ; i <str.length() ; i++)
        cout <<str[i]
OR
for(char ch : str){
        cout << ch;
```

Reverse a String

For character array

```
st=0    end=n-1
  while(st <end){
  swap(str[st],str[end])
        st++
        end--
      }
```

For Strings

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

Palindrome

Problem.