

Character Array

* Initialization

char a[10];
char a[] = { 'a', 'b', 'c' } ↗ 1 byte
↘ it will be of size 3

also
char a[10] = { 'a', 'b', 'c' } ↗ first 3
↘ or char will be
initial the rest
are
garbage

char a[10] = { 'a', 'b', 'c', '\0' }

Recommended
ASCII code = 0

↳ because ↗

* difference b/w integer array vs character array.

① `int b[] = { 1, 2, 3 }; print`
`cout << b << endl;` → 701568 (Address of b)

`char a[] = { 'a', 'b', 'c', 'd', 'e', '\0' };`

`cout << a << endl;` → abcde (contents, unless it gets null char)
 that is why if is imp to put 0/ \0 at end. otherwise if
`a[] = { 'a', 'b', 'c', 'd', 'e' };`
`cout << a << endl;`
 → abcde~~xyznm~~ garbage.

② 1/ Input

`char s[10];`

`cin >> s;` → direct cin needed for loop
 because when we input

hello

↓
 input
 ↓

hello → stored inside for memory.

~~eg~~ char s1[] = {'h', 'e', 'l', 'l', 'o'};
 char s2[] = "hello";

cout << s1 << " " << sizeof(s1) << endl;
 cout << s2 << " " << sizeof(s2) << endl;

helloabcde 5

hello 6

because

s1 →

h	e	l	l	o	a	b	c	d	e	
---	---	---	---	---	---	---	---	---	---	--

s2 →

h	e	l	l	o	\0
---	---	---	---	---	----

on subjected.

so it is important to end a char array with null character.

* to read paragraph

cin.getline(a, 1000, '\$');

also just
(a, 1000)

↙ since length of array
 start of array

these characters
 when you should ~~stop~~ stop
 processing your
 input.

* Palindrome \rightarrow symm. abt the mid point

a b c b a
 \leftarrow \rightarrow

* trick, \rightarrow `cin.getline(a, 1000);`

of the code

int n;

cin >> n;

no. of strings to take input

for (int i = 0; i < n; i++)
 {

cin.getline(a);
 cout << a << endl;

}

input

3

hello word.

hey b k j

etkkr.

Output

Reads just 2 strings

hello word
 hey b k j.

because in input buffer -

3 \n/ hello word \n/ hey b k j \n/ \n/

to The pointer is at the so our first string that it
 getline taking is empty. , to avoid
 this we can consume this by adding -
`cin.get();`