Date :

[Arrays & Pointers]

```
int a[10];
cout << a
```

or `cout << &a` — SAME!

Gives address of a[0] & a[0].

`cout << *a`

does not take any additional memory.

a is pointing to a[0] so *a is the value at that address.

+ Don't try to change a.

`Cout << sizeof(a)`

40 bytes

because of symbol table and also 'a' is not given separate memory (4×10)

⇒ 'a' is not allotted any separate memory ∴ you cannot change 'a'. 'a' is pointing to 'a[0]' and remains as it is.

* `int a[6] = {1, 2, 3}`

assigns other elements as 0 by default.