

i) Enumeration :-

Syntax :- enum Var_Name { Low,
Medium,
High } |
Low = 47,
Medium = 50,
High = 53

Notes :-

i) Upper Case (Proper) }

↳ lower Case ✓)

x { Low = 6,
Medium,
High }

ii)

```
int main() { Enum Var_name Var_name1 = MEDIUM;
```

Cout << Var_name1 ; -> // number

Switch ()

{ Case - .

Case - .

} default :

return 0;

}

*Pointers :- A kind of Variable use to store the address of any Variable however you need to make sure that the datatype should match with the target variable.

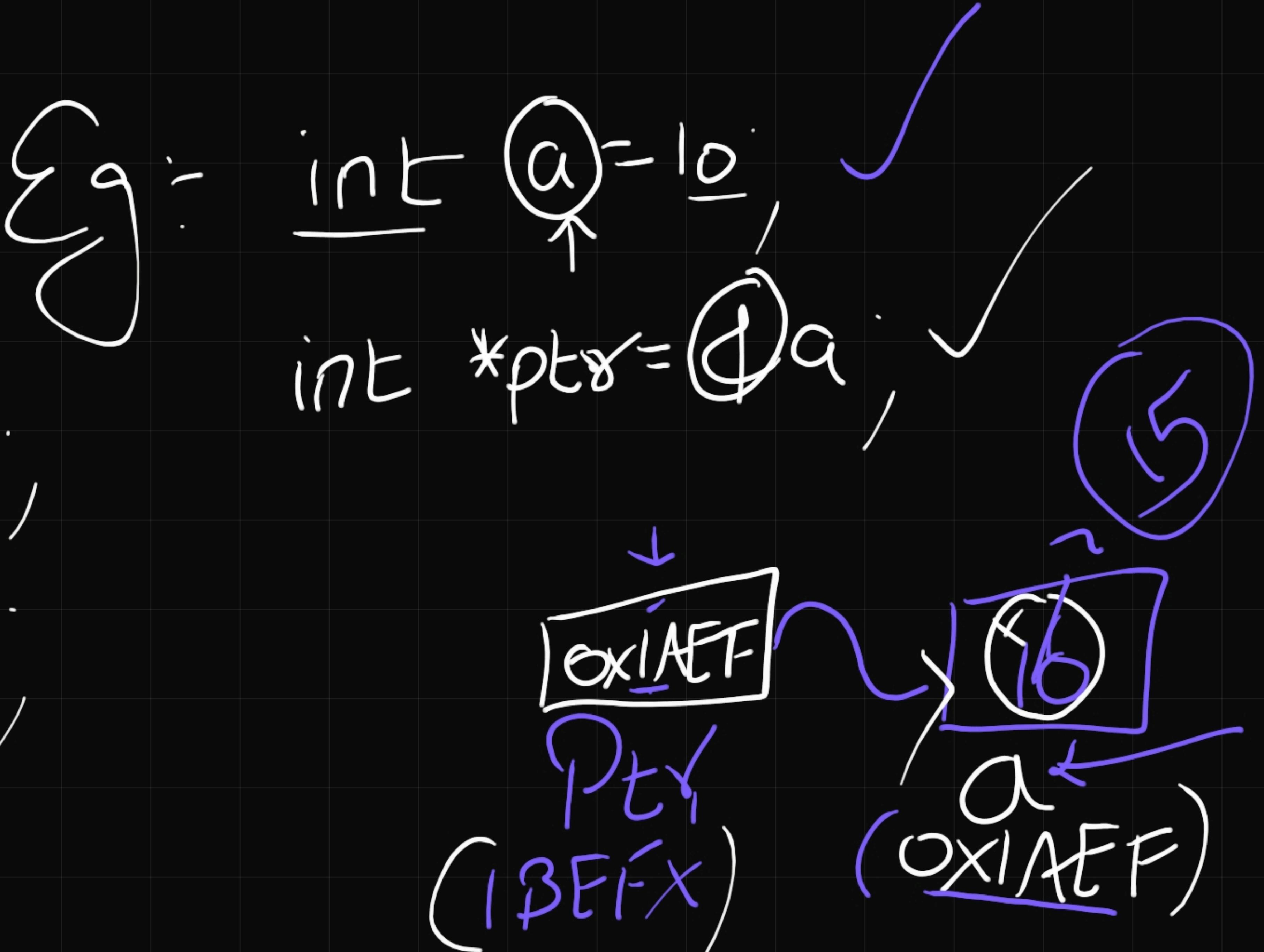
① Normal Variable:-

```
Cout << Pt << endl;
```

```
Cout << *Pt << endl;
```

```
Cout << a
```

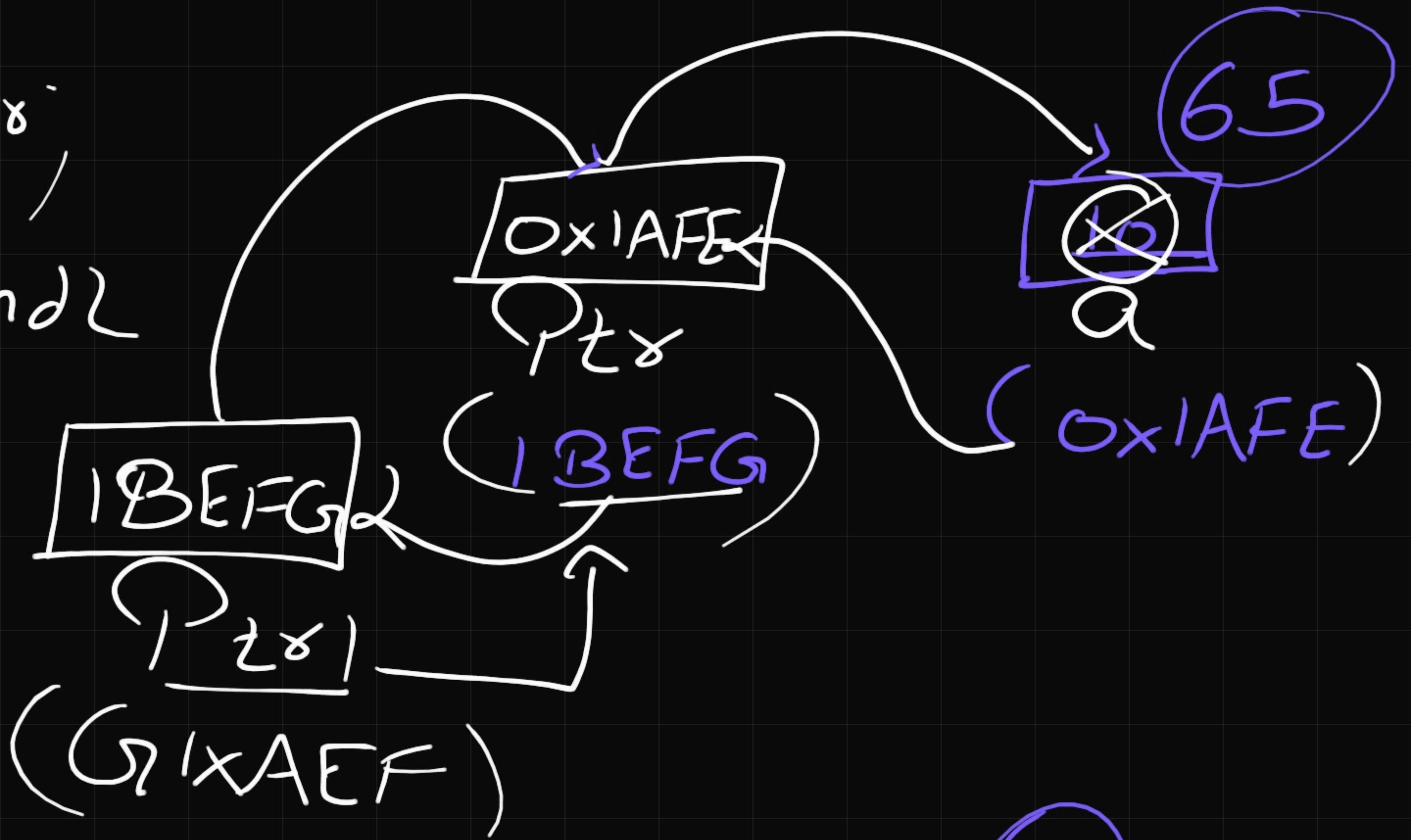
```
*Pt = 15;
```



M/S
M/E

```
int a = 10;  
int* pEx = &a; ←  
int** pEx1 = &pEx;  
cout << *pEx1 << endl;  
(*(pEx1)) = 65;
```

*
PEx1



Notes:- i) Pointer handle

ii) Pointer **ptr**

→ int arr[5];

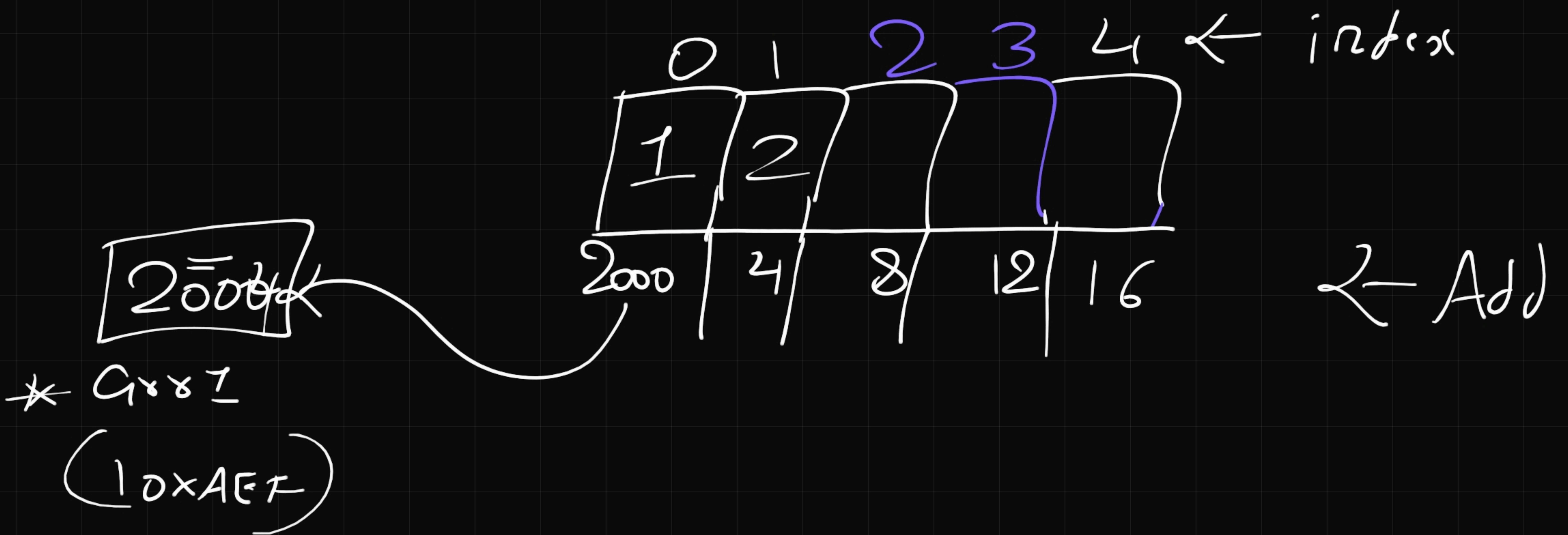
int* pEx = arr;

Cout << "Enter total 5 numbers" << endl;

for (int i=0; i < 5; i++)
{
 Cin >> pEx[i];
}

for (int i=0; i < 5; i++)
{
 Cout << "Current index is: " << i << " "
}

" " << pEx[i] << endl;



4x8I [0]
[1]
[2]
[3]

Memory Management - (Memory Utilisation)

↳ Efficiently

$$\text{App1}^n = 15 \text{ Bytes} - 4 = 15$$

int a=10; (4) ~~X~~

↳ Unnecessary waste of memory
↳ App1's downside.

- ① new
- ② delete

Normal Variable | Array

* Normal Variable :-

int *px = new int (n)
*px = 36;

delete px;

*Array :- int *arr = new [size]
Delete arr;

Normal Variable

int *ptr = new int

Delete ptr;

Array

int *arr = new [15]

Delete [arr];