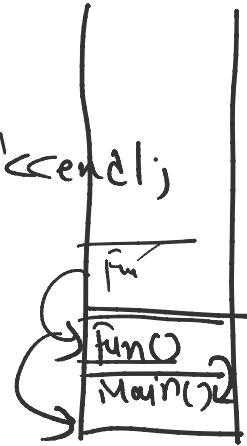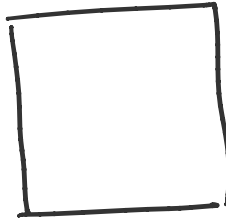# functions:- ⟶ Agent

void fun() { cout << "function called" << endl;

⟶ int main() {

→ | fun ( ); |

return 0;
}

---

return_type   function_name (arguments) {
_____
_____
_____
return —
}

Add 2 numbers
a, b

void   Addnum(int x, int y) {
     cout << "sum of two number :-" << x+y << endl;
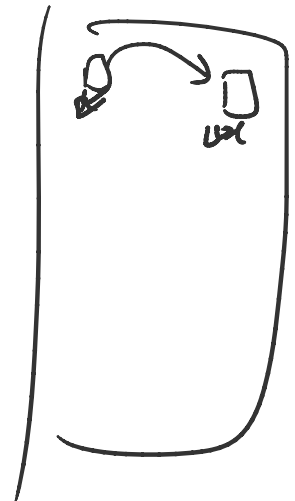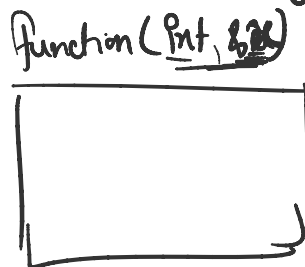}

int main() {

```cpp
int main() {
    int a,b;
    cin>>a>>b;
    Addnum(a,b);
    return 0;
}
```
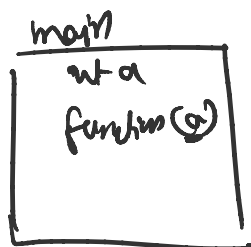
return → void

return → int, char, double, float, string,

argument →
- Pass by value
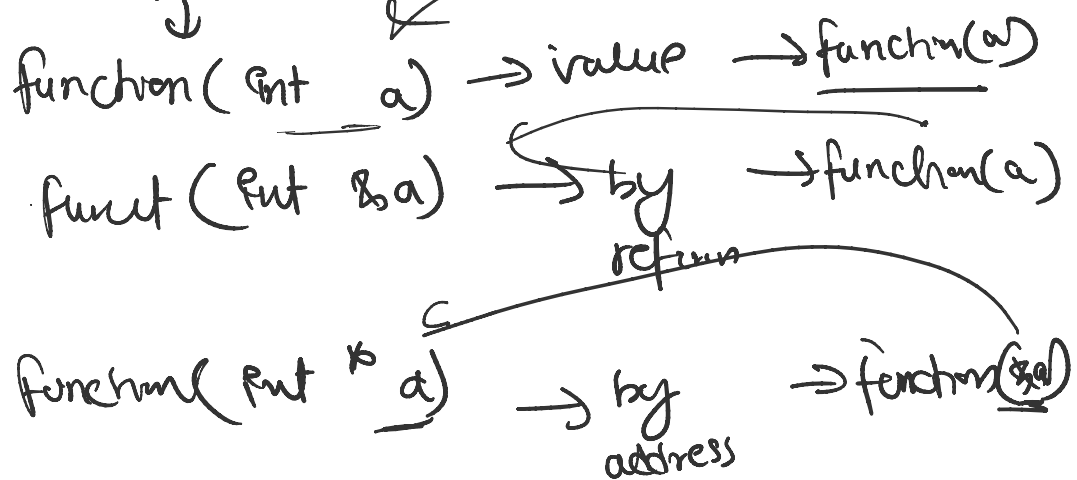- pass by reference
- pass by address

main
w a
function (a)

function (int, &a)

pass by value → copy of argument

pass by Reference → give you a Reference

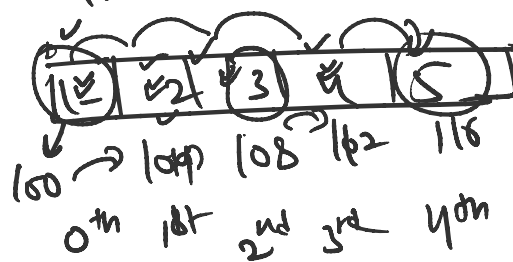pass by Address → define

Mam

pass by __Address__ → __Main__ ↓

define
↓

funchon (int _a_) → ivalue → funchon(a)

funct (int &a) → by → funchon(a)

reterr

funchm(int * _a_) → by → funchon(&a)

address

int a,b,c,d,e;

$\begin{cases} a=0 \Rightarrow 100 \\ b=3 \Leftarrow 3000 \\ c=2 \Rightarrow 800 \\ d=10 \Rightarrow 200 \\ e=5 \Rightarrow 100000 \end{cases}$ 2

syntax

int arr[5];

int arr[] = {1,2,3,4,5};

int arr[5] = { n };

100 → 104   108   112   116

0th   1st   2nd   3rd   4th

idx : 0, 1, 2, 3, 4

arr[2] = 3

arr[4]

```
for( int i=0 ; i< 5 ; i++ ) {

    cout << arr[i] << endl;
}
```

i=0 ; i<5
    arr[0];

i=1 ; i<5;
    arr[1];

i=2
    arr[2]
    .
    .
    arr[4]

arr[] => { ...... } (n)

arr[]

total => m/4

int * arr ⟶ {1,2,3,4,5}

arr[]