int& a = b is setting a's ADDRESS to b's ADDRESS

f(int ,int&, int\*);

int -> We should pass the value (Call by value)

int& -> It takes address(&) of the parameter

int\* -> It is an integer pointer which points to an integer variable

**in main():**

a=1,b=2,c=3,d[2]={4,5}

c = myfcn(a,b,d);

**in myfcn():**

myfcn(int x, int& y,int \*z)

int x; -> call by value, we can’t change the value

int& y-> call by address, we can change the value

int\* z-> call by reference, we can change the value

Here

x=a

address of b = address of y

z is pointing to base address of array d[2]

Int a=16,b=17; -> No effect variables in main becasue these are local variables

x=13; -> No effect on a in main() becasue it is call by value. So x=a=1

y=14; -> Here is an effect on b, We are assigning address of 14 to address of y. So b=14

z[1]=15; -> Here is an effect on array d[2], int \*z is call by reference so we can change the value.

z[1] is nothing but d[1]. So d[1]=15. So d[] = {4,15}

return x; -> x has value 13, myfcn() returning value 13 which is stored in c becasue we are assigning return value of mufcn() to c as c = myfcn(a,b,d); so c=13

**Final solution:**

a=1,b=14,c=13,d[0]=4,d[1]=15