REFERENCES, POINTERS
PASSING PARAMETERS TO FUNCTIONS

Problem Solving with Computers-I

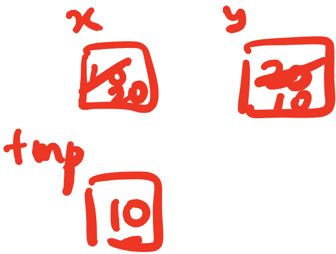
```
#include <iostream>
using namespace std;

int main(){
    cout<<"Hola Facebook\n";
    return 0;
}
```

GitHub

Pass by value

int x = a

```
void swapValue(int x, int y){
    int tmp = x;
    x = y;
    y = tmp;
}
int main() {

    int a=30, b=40;

    cout<<a<<" "<<b<<endl;

    swapValue( a, b);

    cout<<a<<" "<<b<<endl;
    return 0;
}
```

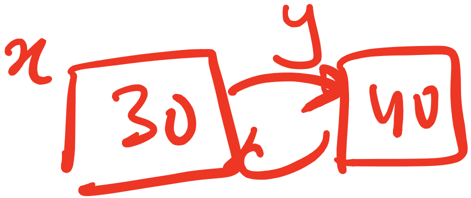


What is printed by this code?

[✓ **A.**]

30 40

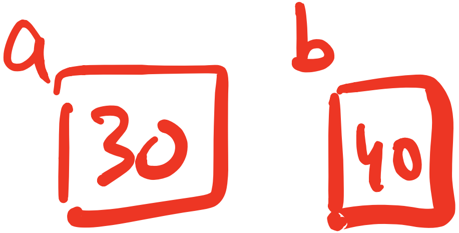
30 40



B.

30 40

40 30

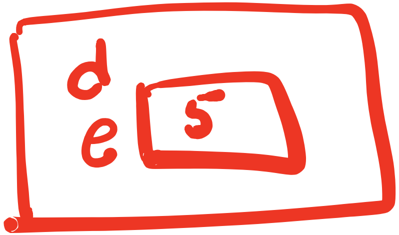


C. Something else

References in C++

A reference in C++ is an alias for another variable

```
int main() {
   int d = 5;
   int &e = d;
}
```



reference

↓

"nickname"

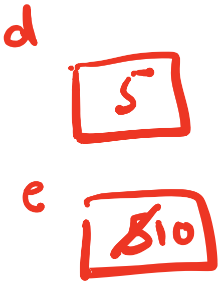
↓

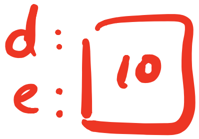
"alias"

e = 10; cout << d;

```
int d = 5;
int e = d;
```

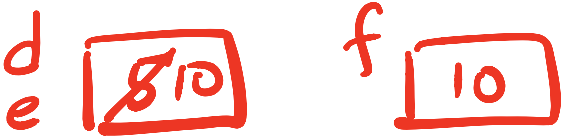
e = 10





e = 10

References in C++



```
int main() {
  int d = 5;
  int &e = d;
  int f = 10;
  e = f;

}
```

How does the diagram change with this code?

[✓ A.]
d:
e: 10

f: 10

B.
d: 5

e:
f: 10

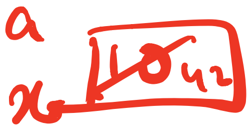
C.
d:
e: 10
f:

D. Other or error

```
int d = 5;
int &e = d;
int &f = d;
```

```
int &f = e;
```

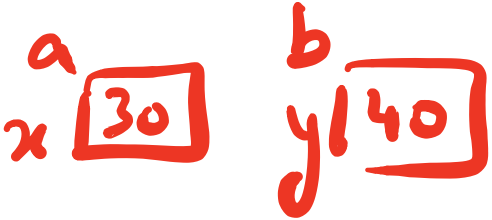
```
void   foo (int& x ) {
          x = 42;

}
int   main() {
        int a = 10;
        foo(a);
}   cout << a;
```



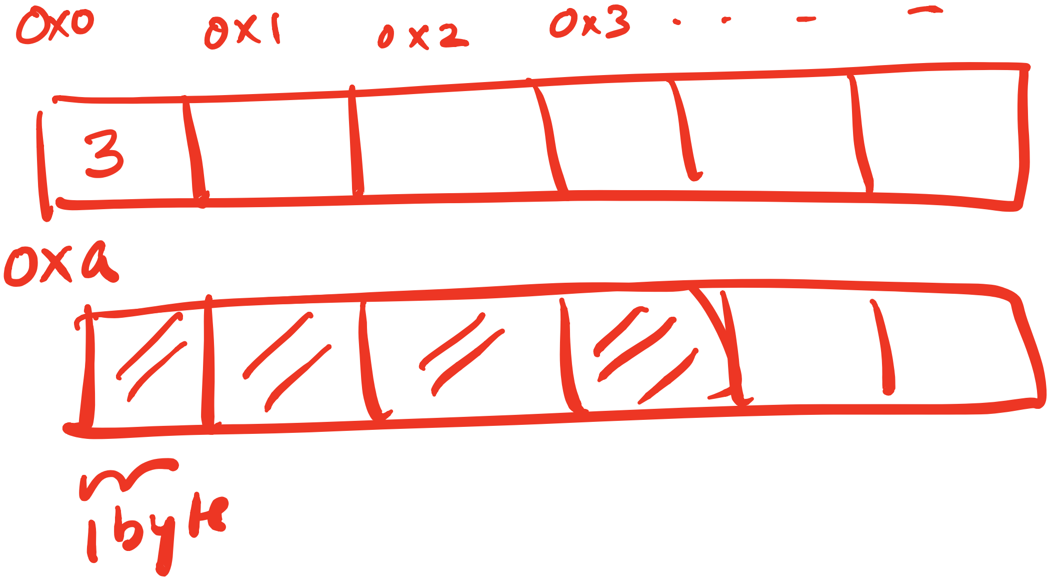
```
int & x = a
```

Passing parameters by reference

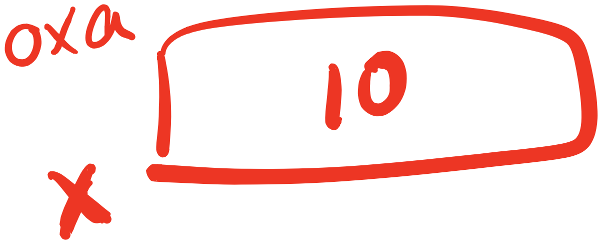
```
 void swapValue(int&x, int&y){
      int tmp = x;
      x = y;
      y = tmp;
 }
int main() {
     int a=30, b=40;
     swapValue( a, b);
     cout<<a<<" "<<b<<endl;
}
```



Memory



int x = 10;



```
cout << &x;
```

↑ get the location & x

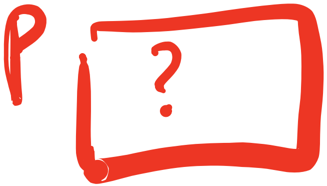
Pointers

- Pointer: A variable that contains the address of another variable
- Declaration: *type ** pointer_name;

```
int* p;
```

int * p;

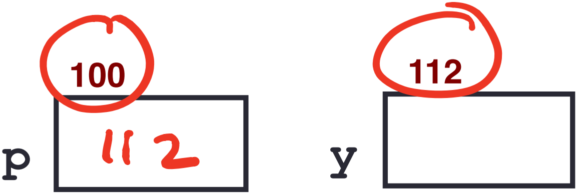
is a pointer = stores the address



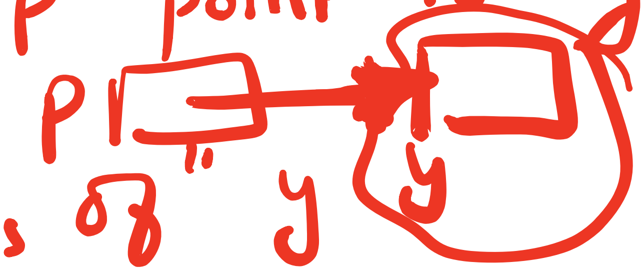
How to make a pointer point to something

int* p;   int *p;

```
int *p;
int y = 3;
```



p "point to" y



p = &y;

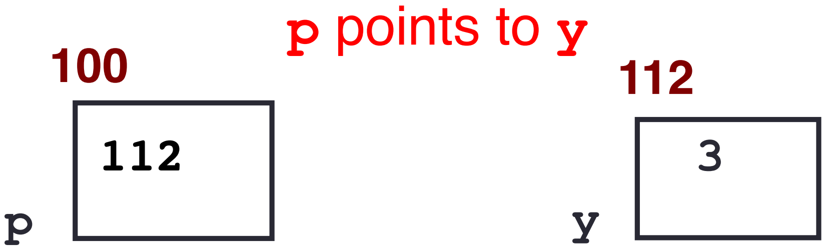
"Address of" y

int* p = &y;

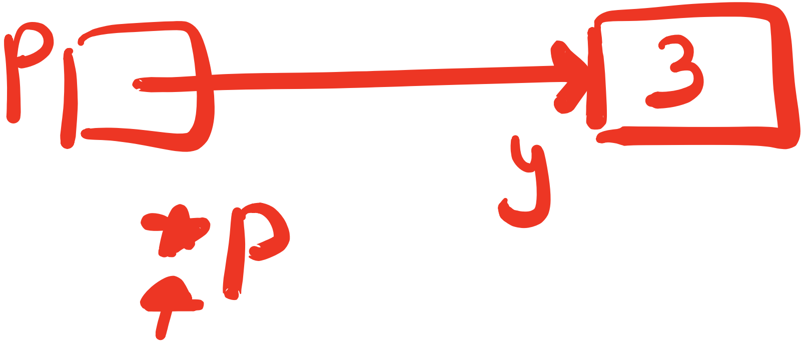
To access the location of a variable, use the address operator '&'

Pointer Diagrams:
Diagrams that show the relationship between pointers and pointees

Pointer: p

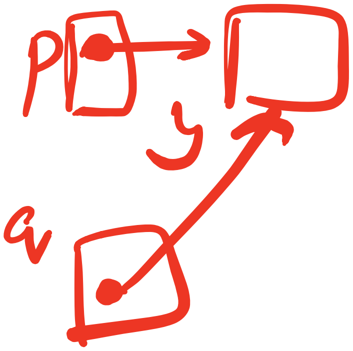


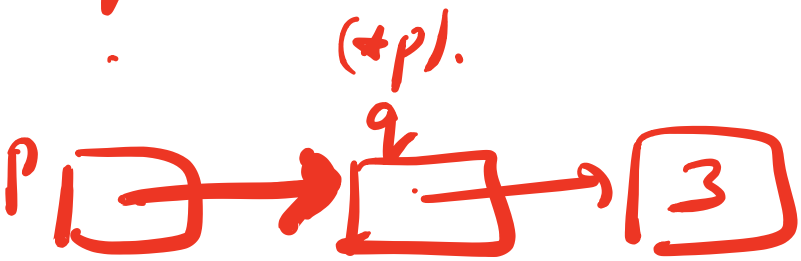
Pointee: y



int * p;

```
int* p;
p = &y;
int* q;

q = p;
```





an

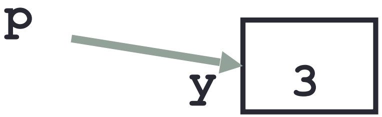
operator (dereferencing p

You can change the value of a variable using a pointer !

```
int *p, y;
y = 3;
p = &y;

*p = 5;
```

Two ways of changing the value of a variable

- Change the value of y directly:

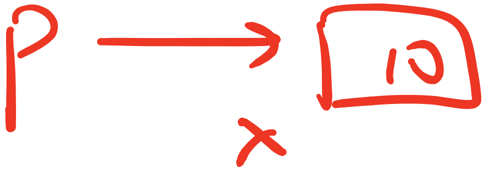


- Change the value of y indirectly (via pointer p):

Tracing code involving pointers

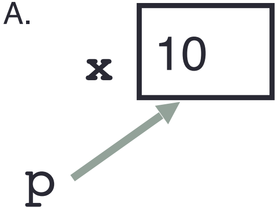
```
int *p;
int x=10;
p = &x;
*p = *p + 1;
```

x = x+1

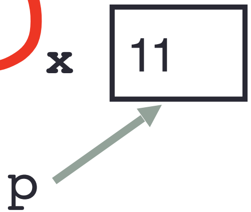


Q: Which of the following pointer diagrams best represents the outcome of the above code?

A.



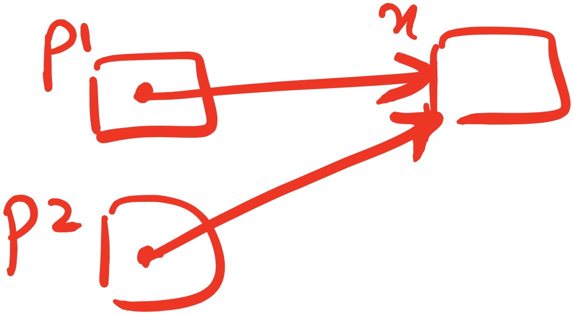
[✓ B.]



C. Neither, the code is incorrect

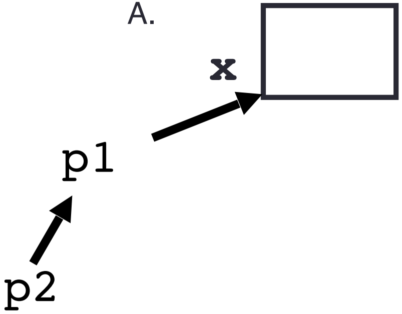
Pointer assignment

```
int *p1, *p2, x;
p1 = &x;
p2 = p1;
```

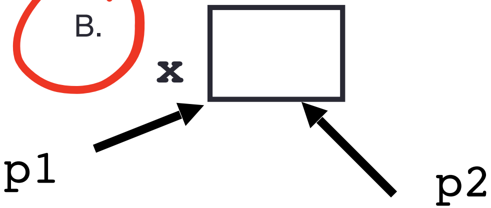


Q: Which of the following pointer diagrams best represents the outcome of the above code?

A.



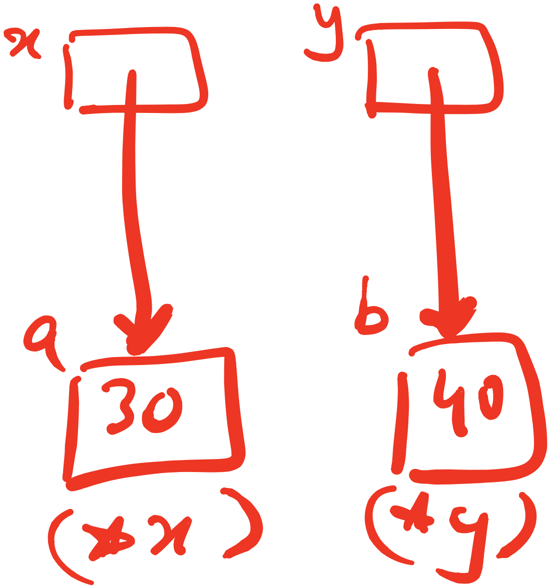
[✓ B.]



C. Neither, the code is incorrect

Passing parameters by address

```
void swapValue(int* x, int* y){
      int tmp = *x;
   *x = *y;
   *y = tmp;
}

int main() {

     int a=30, b=40;

     swapValue(&a,&b);

     cout<<a<<" "<<b<<endl;

}
```



Next time

- Arrays and pointers
- Structs