

Unidad 5: Punteros

<http://bit.ly/2HRBWgq>

Profesores:

Ernesto Cuadros- Vargas, PhD.

ecuadros@utec.edu.pe

María Hilda Bermejo, M. Sc.

mbermejo@utec.edu.pe

Telegram:

1. Configurar tu cuenta

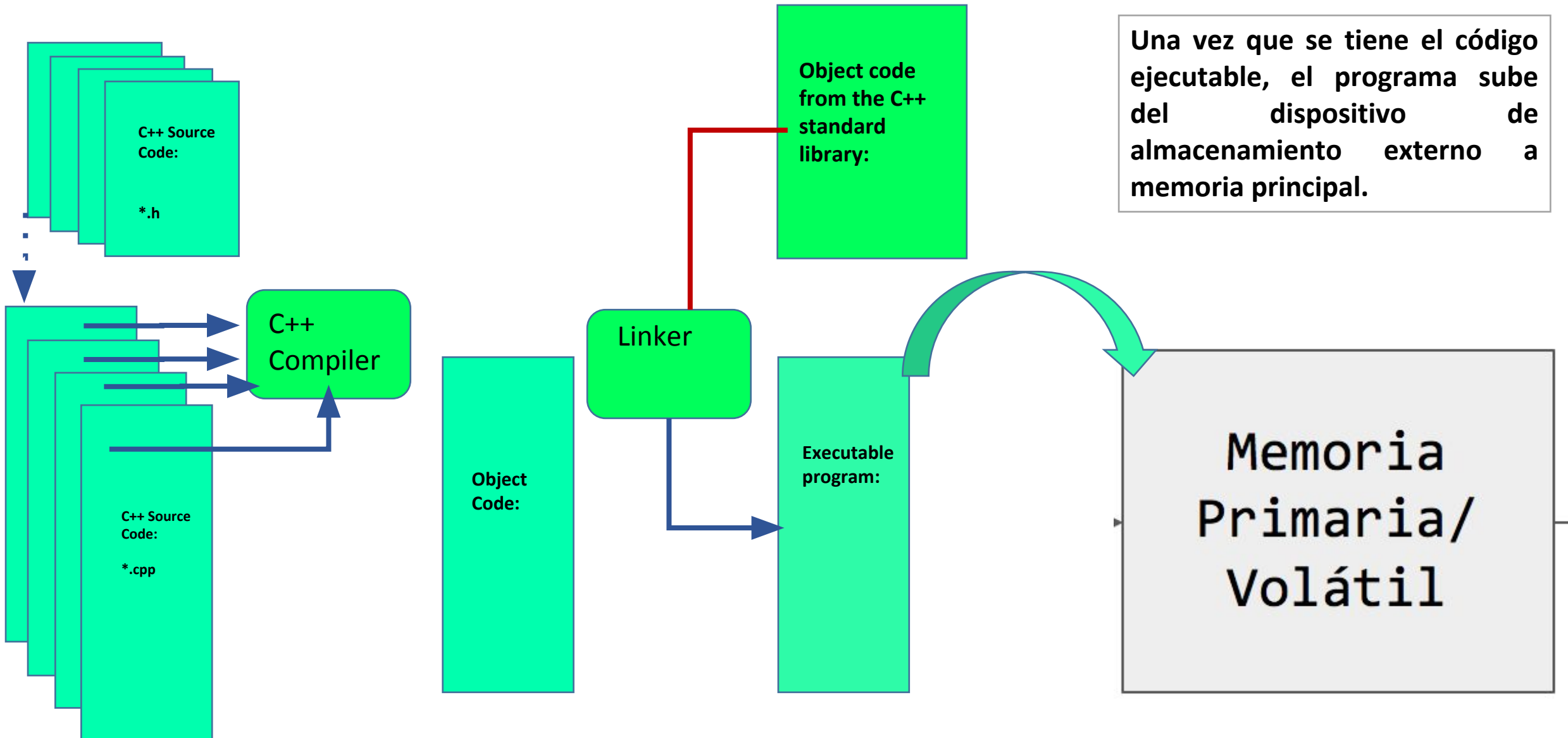
2. <http://bit.ly/2TJnwBq>

Logro de la sesión:

Al finalizar la sesión, los alumnos desarrollan sus programas utilizando punteros.

Punteros

¿Qué ocurre con el código desde que se escribe hasta que se ejecuta?



Uso de la memoria primaria en C++

Segmento de Información Externa

Argumento externos (argc, argv)

Pila (Stack)

Memoria automática donde se asigna las variables estáticamente.



Montón (Heap)

Memoria donde se asigna variables dinámicamente.

Segmento Estático y Global

Memoria donde se asigna variables globales y tipo static.

Segmento de Código

Memoria donde se guarda el segmento de código

Representación simplificada de la memoria

- Cada Byte tiene su propia dirección

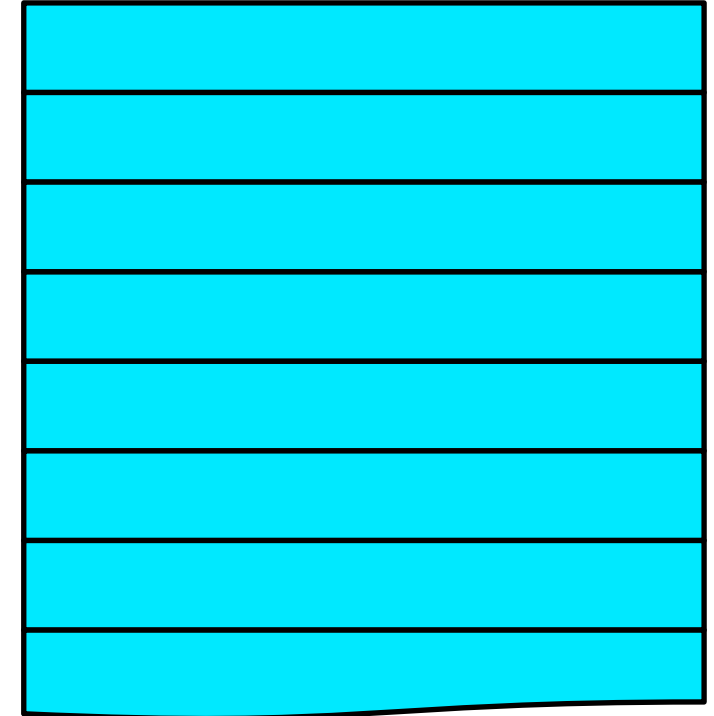
1 Byte

0	1	0	1	0	0	1	1
---	---	---	---	---	---	---	---

Dirección

0x1008
0x1009
0x100A
0x100B
0x100C
0x100D
0x100F
0x1010

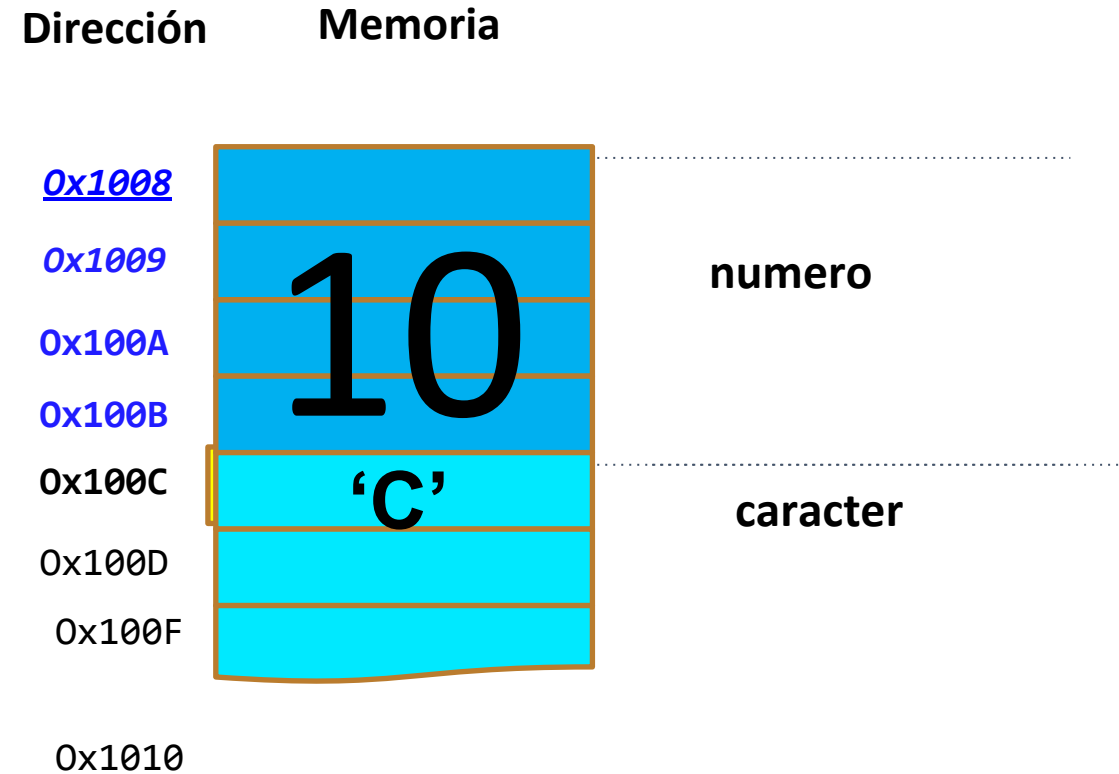
Memoria



Representación de una variable

```
int numero = 10;  
char caracter = 'C';
```

- Los **tipos de datos**: tamaño específico y conjunto de reglas para cada tipo.
- Ejemplo: **char** es un byte
- El tipo **int**: 4 bytes en 32 bits, 8 bytes en 64 bits.



¿Cuál es la dirección de memoria de la variable número?

Dirección de una variable

La forma explícita de obtener la dirección de una variable es por medio del operador **&**

Ejemplo:

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int numero = 10;
7      float x=12.75;
8
9      cout << "Numero = " << numero << "\n";
10     cout << "La direccion de la variable numero es " << &numero << "\n\n";
11
12     cout << "x = " << x << "\n";
13     cout << "La direccion de la variable x es " << &x;
14
15     return 0;
16 }
```

```
Numero = 10
La direccion de la variable numero es 0x7fff599c7a98

x = 12.75
La direccion de la variable x es 0x7fff599c7a94
Process finished with exit code 0
```

¿Qué es un puntero? y ¿Cómo se define?

Un puntero, es un tipo de componente que apunta a otro tipo.

Se utilizan para acceder indirectamente a otros objetos.

El puntero es un objeto que puede ser asignado y copiado.

Se define así:

```
int ival= 42;  
int *p = &ival; // p contiene la dirección de ival; se dice que p apunta a ival
```

La segunda instrucción define **p** como un puntero a un **int** e inicializa a **p** apuntando a un objeto **int** llamado **ival**

¿Cómo se define?

El tipo del puntero y el objeto al cual apunta deben coincidir.

double dval;

double *pd = &dval; // **pd** se inicializa con la dirección de un double

int *pi = pd; // error **pi** y **pd** difieren en el tipo

pi = &dval; // error se asigna la dirección de un double a un puntero a un entero.

El valor de un puntero puede:

1. Apuntar a un objeto ó
2. Puede ser ***nullptr***, que indica que el puntero no ha sido ligado a ninguna variable/objeto.

Usando un puntero para acceder a un objeto

Cuando un puntero apunta a un objeto, se puede utilizar el "dereference operator" (the * operator) para acceder al objeto.

```
int ival = 42;
```

```
int *p = &ival; // p contiene la dirección de ival; p es un puntero a ival
```

```
cout << *p; // * se accede al objeto al cual apunta p; se imprime 42
```

Desreferenciando un puntero se accede al objeto que es apuntado por el puntero.

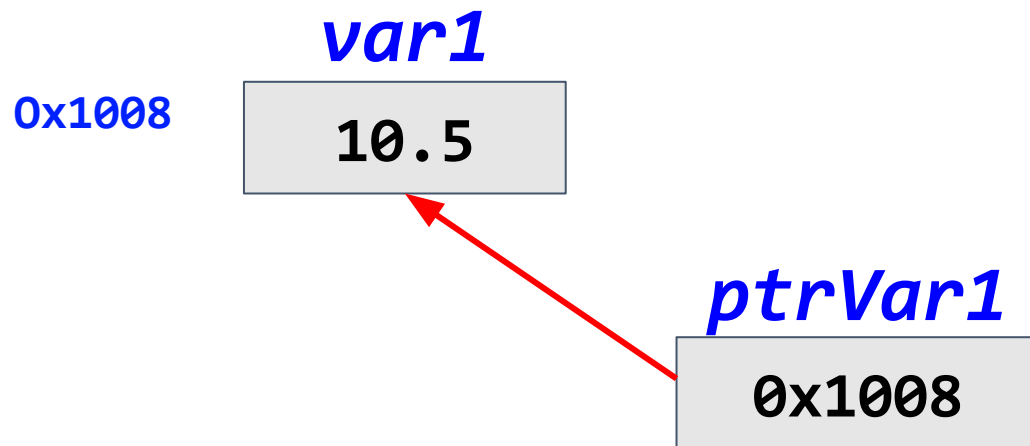
```
*p = 0; // * se accede al objeto, se asigna un nuevo valor a ival a través de p
```

```
// Cuando se asigna a *p, estamos asignando el valor 0 al objeto al cual p apunta
```

```
cout << *p; // se imprime 0
```

Veamos lo que podría ocurrir en la memoria

```
double    var1 = 10.5;  
double*   ptrVar1 = &var1;
```



Dirección

Memoria

0x1008

0x1009

0x100A

0x100B

0x100C

0x100D

0x100E

0x100F

0x1010

0x1011

0x1012

0x1013

0x1014

0x1015

0x1016

0x1017

10.5

var1

0x1008

ptrVar1

Un puntero en Clion utiliza 8 bytes.

Null Pointers

Un *nullptr* no apunta a ningún objeto.

```
int *pi = nullptr;    // es equivalente a  int *pi=0;  
int *p2 = 0;         // inicializa p2 con la constante literal 0
```

```
int zero=0;  
pi = zero;  // error, no se puede asignar un int a un puntero
```

Mas sobre punteros

```
int i = 42;
```

```
int *pi = 0;    // pi es inicializado pero no tiene la dirección de ningún objeto
```

```
int *pi2 = &i;  // pi2 se inicializa y contiene la dirección de i
```

```
int *pi3;       // pi3 no está inicializado
```

```
pi3 = pi2;      // pi3 y pi2 tiene la dirección del mismo objeto en el ejemplo i;
```

```
pi2 = 0;        // pi2 ahora no apunta a ningún objeto.
```

```
int ival = 72;
```

```
pi = &ival;    // El valor de pi cambia, ahora pi apunta a ival
```

```
*pi = 0;       // el valor en ival cambió a cero, pi no cambió
```

Mas sobre punteros:

```
int ival = 1024;
```

```
int *pi = 0;    // pi es válido, pi tiene asignado nullptr;
```

```
int *pi2 = &ival; // pi2 es un puntero válido, tiene la dirección de ival
```

```
if(pi) // pi tiene el valor 0, entonces si se evalúa la condición es falsa
```

```
if(pi2) // pi2 es un puntero a ival, entonces no vale cero, si se evalúa la condición  
        // es verdadera
```


Ejercicios:

1. ¿ Qué hace el siguiente código?:

```
int i = 42;
```

```
int *pi = &i;
```

```
*pi = *pi * *pi;
```

2. Indica si hay definiciones ilegales y ¿por qué?.

```
int i =0;
```

a) double *p = &i;

b) int *pi =i;

c) int *p=&i;

Ejercicios:

3. ¿Por qué la inicialización de **p** es legal y la de **lp** es ilegal?

```
int i = 42;
```

```
void *p= &i;
```

```
long *lp=&i;
```

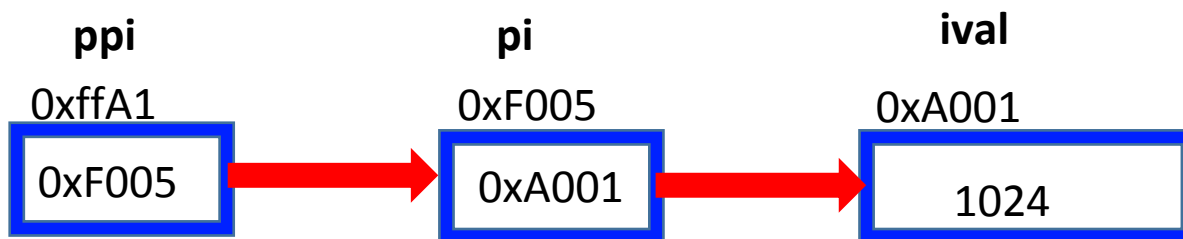
Punteros a punteros:

Un puntero es un objeto en memoria, entonces como cualquier objeto tiene una dirección. Por lo tanto se puede asignar la dirección de un puntero en otro puntero.

```
int ival = 1024;
```

```
int *pi = &ival; // pi es un puntero a un int
```

```
int **ppi = &pi; // ppi es un puntero a un puntero de un int
```



```
cout << "Los valores de ival \n";
```

```
cout << "Valor directo      : " << ival << "\n";
```

```
cout << "Valor indirecto     : " << *pi << "\n";
```

```
cout << "Doble valor indirecto : " << **ppi;
```

Referencia: Es un alias

```
int ival= 1024;
```

```
int &refVal = ival; // refVal refers to ival (es otro nombre de ival)
```

```
refVal = 2; // asigna 2 al objeto al que se refiere refVal, es decir asigna 2 a ival
```

```
int &refVal2; // error: una referencia debe ser inicializado
```

Comparando Transferencia de parametros

```

int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    return 0;
}

```

```

int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{ int temporal = a;
  a=b;
  b=temporal;
}

void swap(int *p1, int *p2)
{ int t = *p1;
  *p1 = *p2;
  *p2 = t;
}

```

Se va analizar paso a paso este programa y se muestra lo que se imprime en la pantalla.

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    return 0;
}
```

oX7fff530d2a58
x 5

oX7fff530d2a54
y 4

oX7fff530d2a48
p

oX7fff530d2a40
q

oX7fff530d2a38
pp

```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{ int temporal = a;
  a=b;
  b=temporal;
}

void swap(int *p1, int *p2)
{ int t = *p1;
  *p1 = *p2;
  *p2 = t;
}
```

Pantalla:

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x = " << x << " y = " << y << "\n\n";

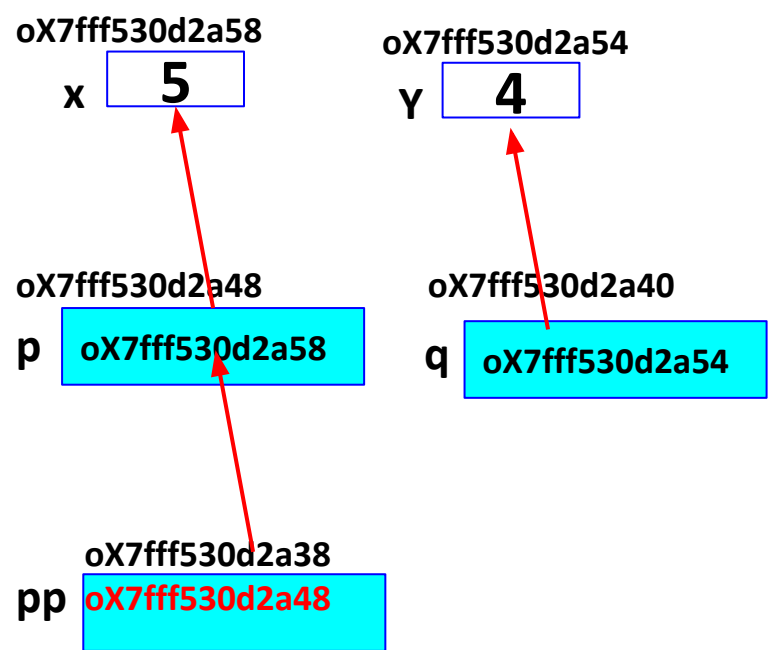
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    return 0;
}
```



```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{
    int temporal = a;
    a=b;
    b=temporal;
}

void swap(int *p1, int *p2)
{
    int t = *p1;
    *p1 = *p2;
    *p2 = t;
}
```

Pantalla:


```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x = " << x << " y = " << y << "\n\n";

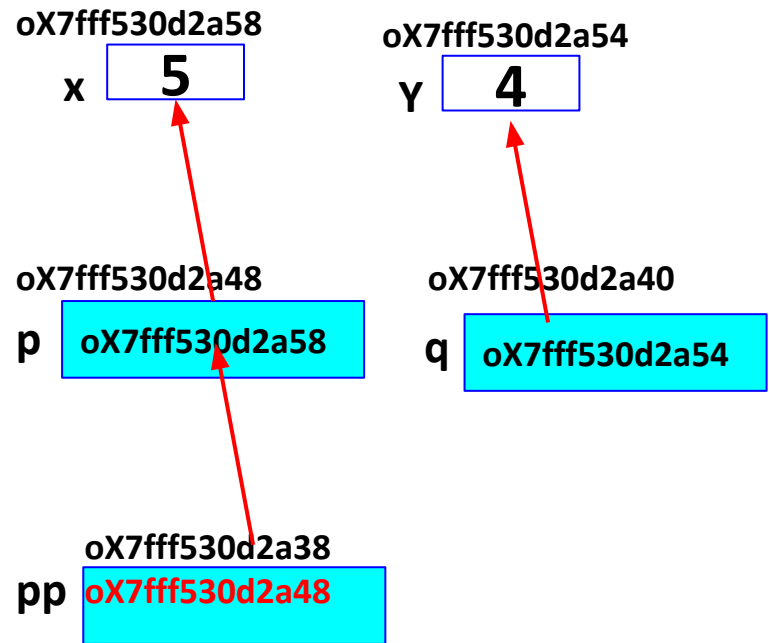
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    return 0;
}
```



```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{
    int temporal = a;
    a=b;
    b=temporal;
}

void swap(int *p1, int *p2)
{
    int t = *p1;
    *p1 = *p2;
    *p2 = t;
}
```

Pantalla:

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x = " << x << " y = " << y << "\n\n";

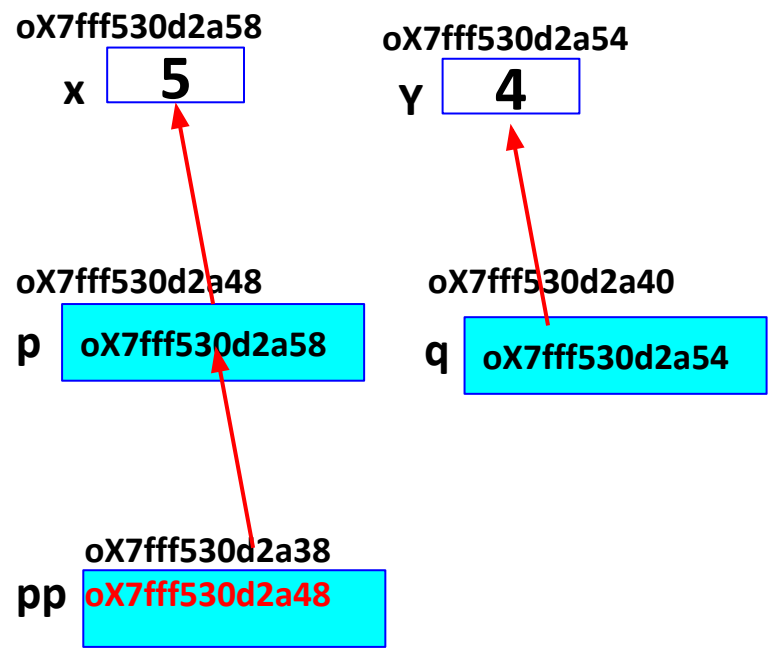
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    return 0;
}
```



a
5

b
4

```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{
    int temporal = a;
    a=b;
    b=temporal;
}

void swap(int *p1, int *p2)
{
    int t = *p1;
    *p1 = *p2;
    *p2 = t;
}
```

Pantalla:
Suma = 9

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x = " << x << " y = " << y << "\n\n";

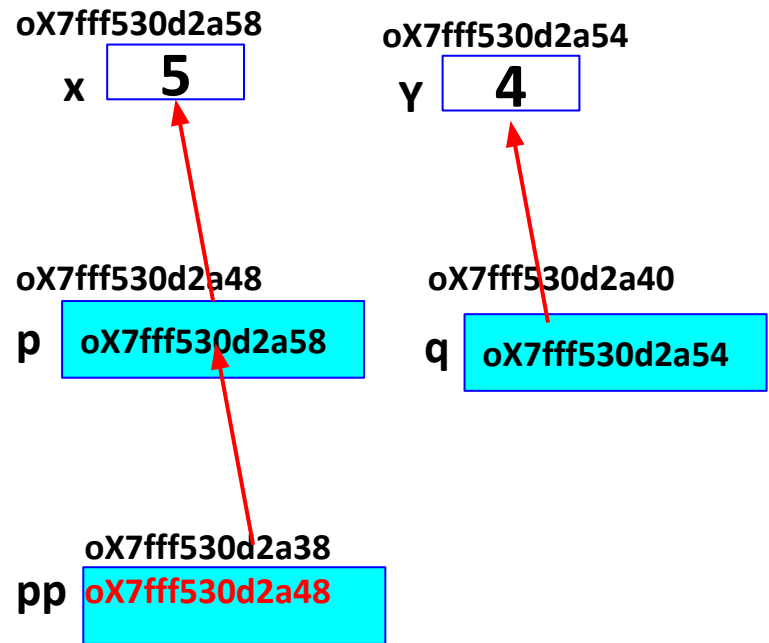
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    return 0;
}
```



```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{
    int temporal = a;
    a=b;
    b=temporal;
}

void swap(int *p1, int *p2)
{
    int t = *p1;
    *p1 = *p2;
    *p2 = t;
}
```

Pantalla:
Suma = 9

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x = " << x << " y = " << y << "\n\n";

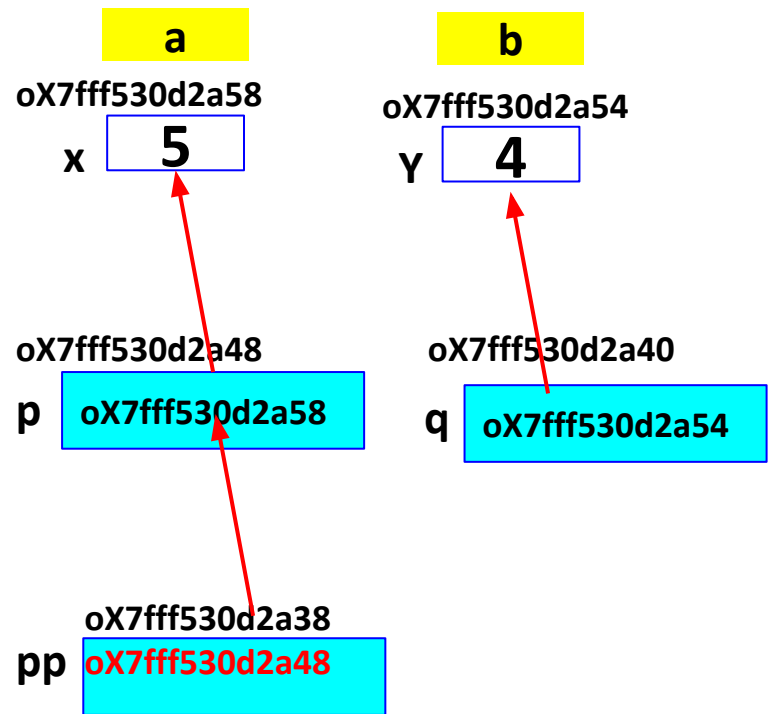
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    return 0;
}
```



temporal

```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{ int temporal = a;
  a=b;
  b=temporal;
}

void swap(int *p1, int *p2)
{ int t = *p1;
  *p1 = *p2;
  *p2 = t;
}
```

Pantalla:
Suma = 9
Primer swap

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x = " << x << " y = " << y << "\n\n";

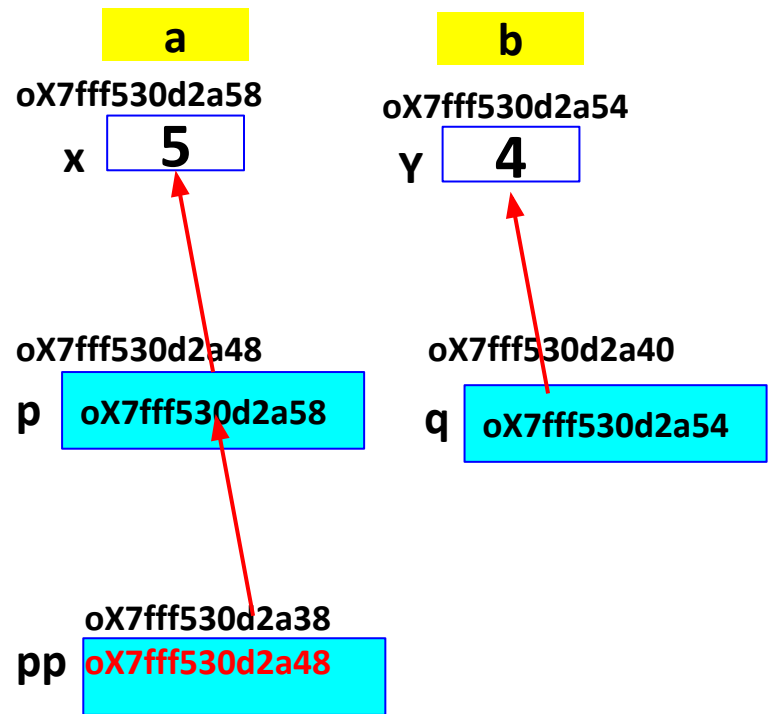
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    return 0;
}
```



temporal
5

```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{ int temporal = a;
  a=b;
  b=temporal;
}

void swap(int *p1, int *p2)
{ int t = *p1;
  *p1 = *p2;
  *p2 = t;
}
```

Pantalla:
Suma = 9
Primer swap

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x = " << x << " y = " << y << "\n\n";

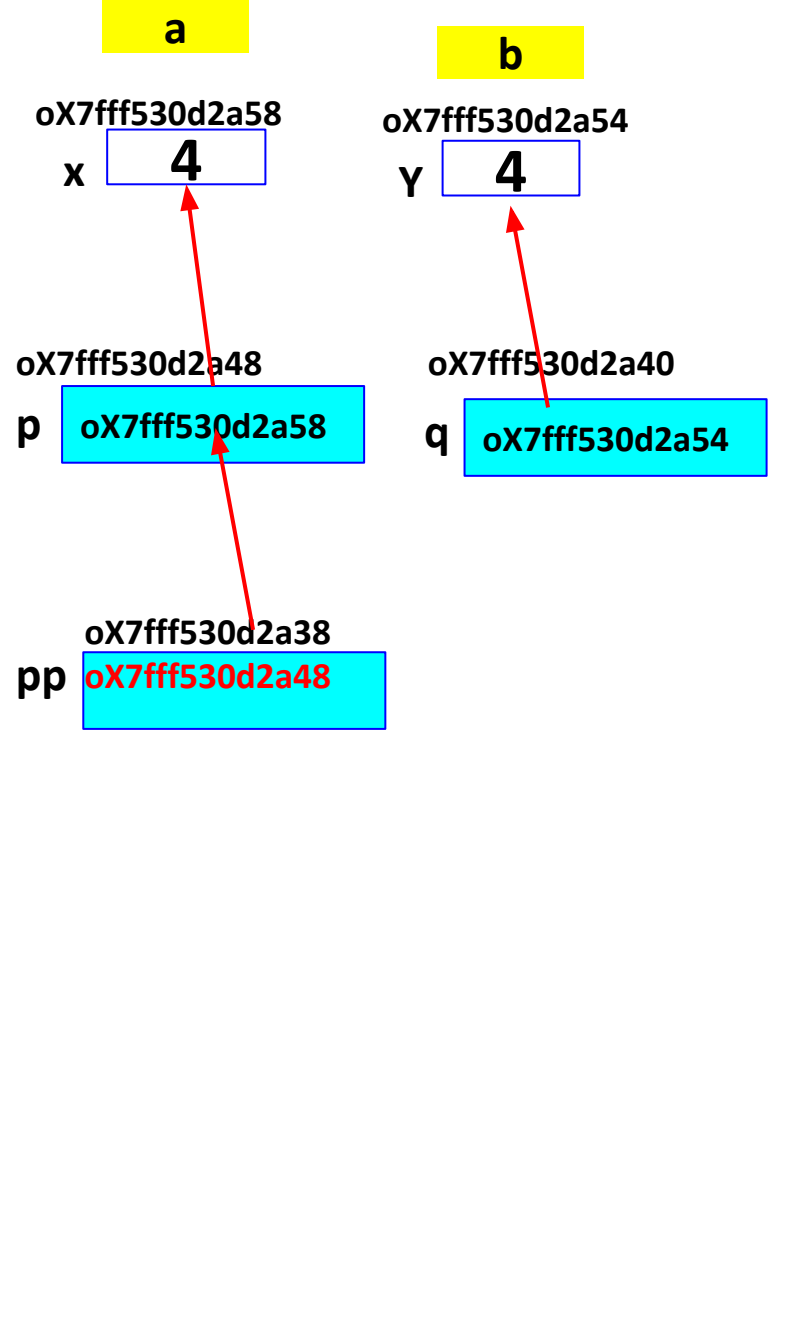
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    return 0;
}
```



temporal
5

```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{ int temporal = a;
  a=b;
  b=temporal;
}

void swap(int *p1, int *p2)
{ int t = *p1;
  *p1 = *p2;
  *p2 = t;
}
```

Pantalla:
Suma = 9
Primer swap

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x = " << x << " y = " << y << "\n\n";

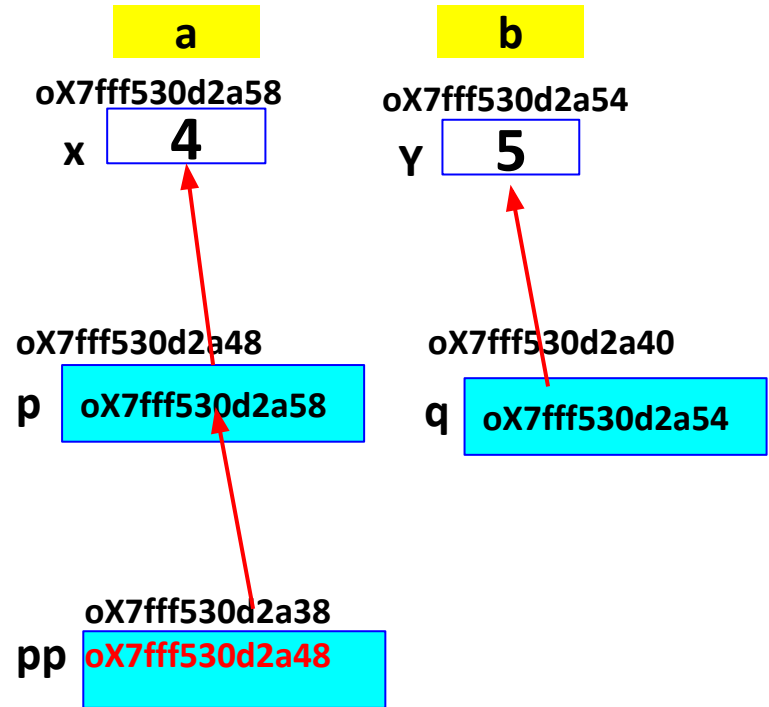
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    return 0;
}
```



temporal
5

```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{ int temporal = a;
  a=b;
  b=temporal;
}

void swap(int *p1, int *p2)
{ int t = *p1;
  *p1 = *p2;
  *p2 = t;
}
```

Pantalla:
Suma = 9
Primer swap

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x = " << x << " y = " << y << "\n\n";

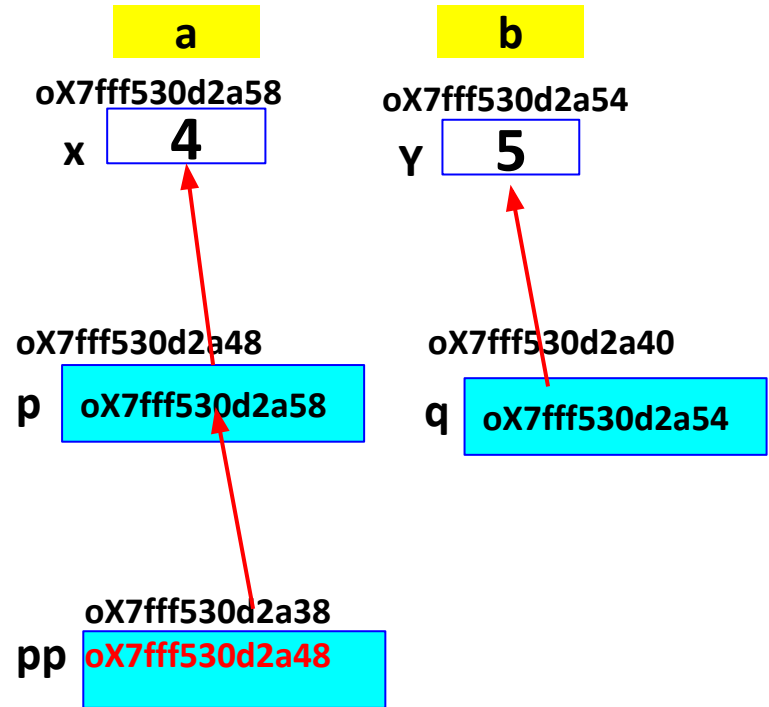
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    return 0;
}
```



temporal
5

```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{ int temporal = a;
  a=b;
  b=temporal;
}

void swap(int *p1, int *p2)
{ int t = *p1;
  *p1 = *p2;
  *p2 = t;
}
```

Pantalla:
Suma = 9
Primer swap
x =4 y = 5


```
int main()
{
  int x=5, y=4, *p, *q, **pp;
  p = &x; q = &y; pp = &p;

  cout <<"Suma = " << fSuma(x,y) << "\n\n";

  cout <<"Primer swap\n";
  swap(x,y);
  cout << "x =" << x << " y =" << y << "\n\n";

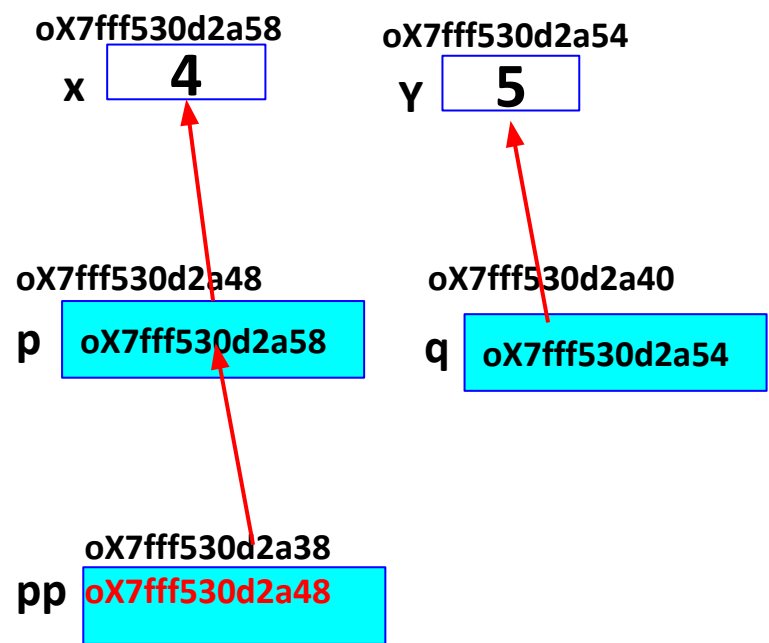
  cout <<"Segundo swap\n";
  swap(&x, &y);
  cout << "x =" << x << " y =" << y << "\n\n";

  cout <<"Tercer swap\n";
  swap(p, q);
  cout << "x =" << x << " y =" << y << "\n\n";

  cout <<"Cuarto swap\n";
  swap(*pp, &y);
  cout << "x =" << x << " y =" << y << "\n\n";

  cout <<"Quinto swap\n";
  swap(*pp, q);
  cout << "x =" << x << " y =" << y << "\n\n";

  return 0;
}
```



```
int fSuma(int a, int b)
{
  return (a+b);
}

void swap(int &a, int &b)
{ int temporal = a;
  a=b;
  b=temporal;
}

void swap(int *p1, int *p2)
{ int t = *p1;
  *p1 = *p2;
  *p2 = t;
}
```

Pantalla:
Suma = 9
Primer swap
x =4 y = 5

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x = " << x << " y = " << y << "\n\n";

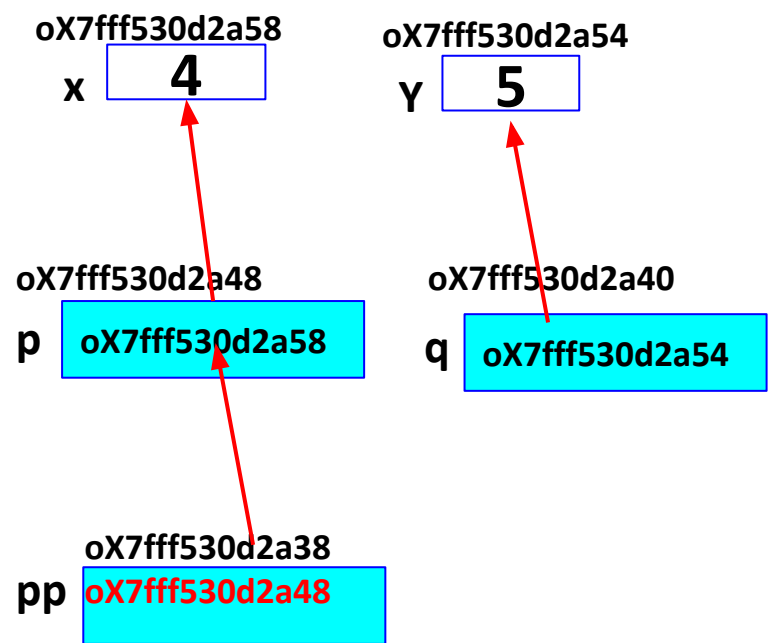
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    return 0;
}
```



```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{ int temporal = a;
  a=b;
  b=temporal;
}

void swap(int *p1, int *p2)
{ int t = *p1;
  *p1 = *p2;
  *p2 = t;
}
```

Pantalla:

Suma = 9

Primer swap

x =4 y = 5

Segundo swap

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x = " << x << " y = " << y << "\n\n";

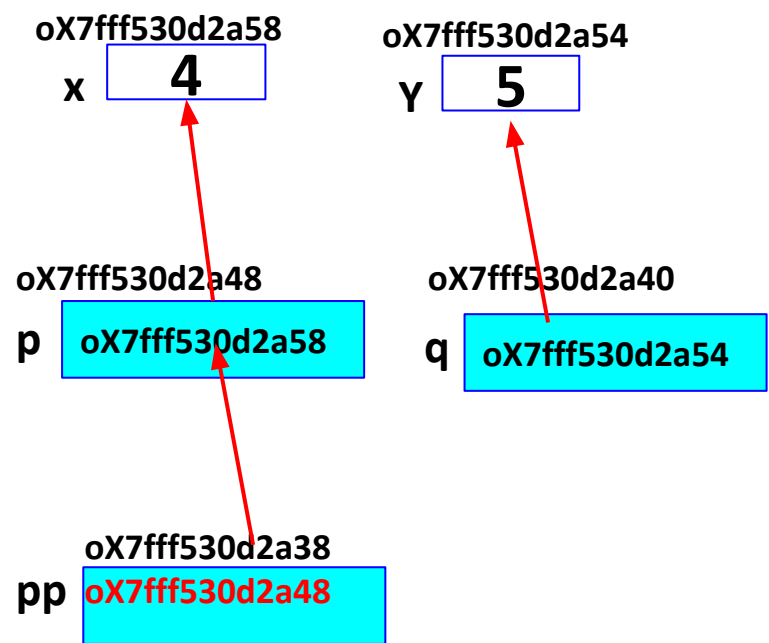
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    return 0;
}
```



```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{ int temporal = a;
  a=b;
  b=temporal;
}

void swap(int *p1, int *p2)
{ int t = *p1;
  *p1 = *p2;
  *p2 = t;
}
```

Pantalla:
Suma = 9
Primer swap
x =4 y = 5

Segundo swap

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x =" << x << " y =" << y << "\n\n";

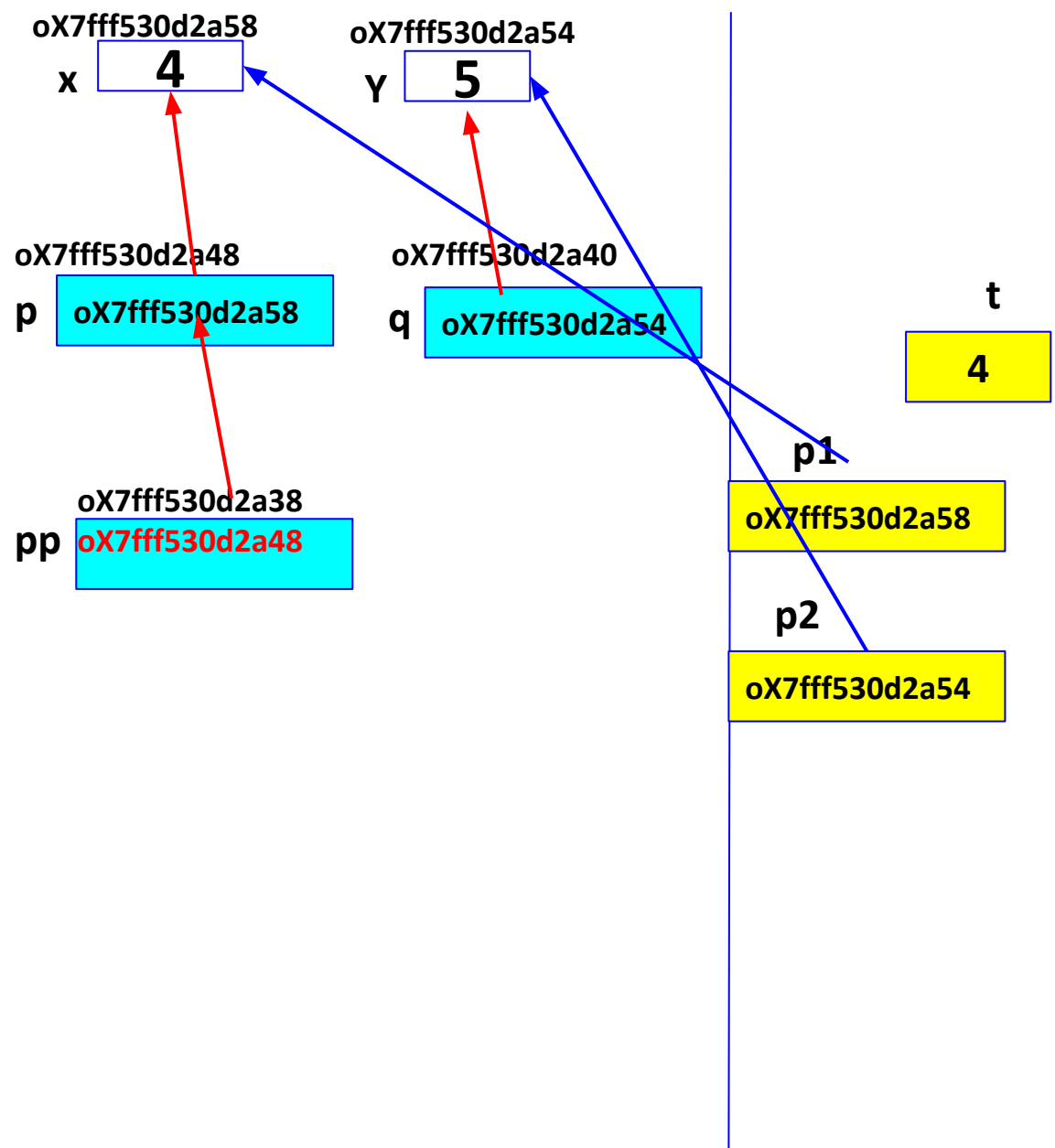
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x =" << x << " y =" << y << "\n\n";

    return 0;
}
```



```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{ int temporal = a;
  a=b;
  b=temporal;
}

void swap(int *p1, int *p2)
{ int t = *p1;
  *p1 = *p2;
  *p2 = t;
}
```

Pantalla:
Suma = 9
Primer swap
x =4 y = 5

Segundo swap

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x = " << x << " y = " << y << "\n\n";

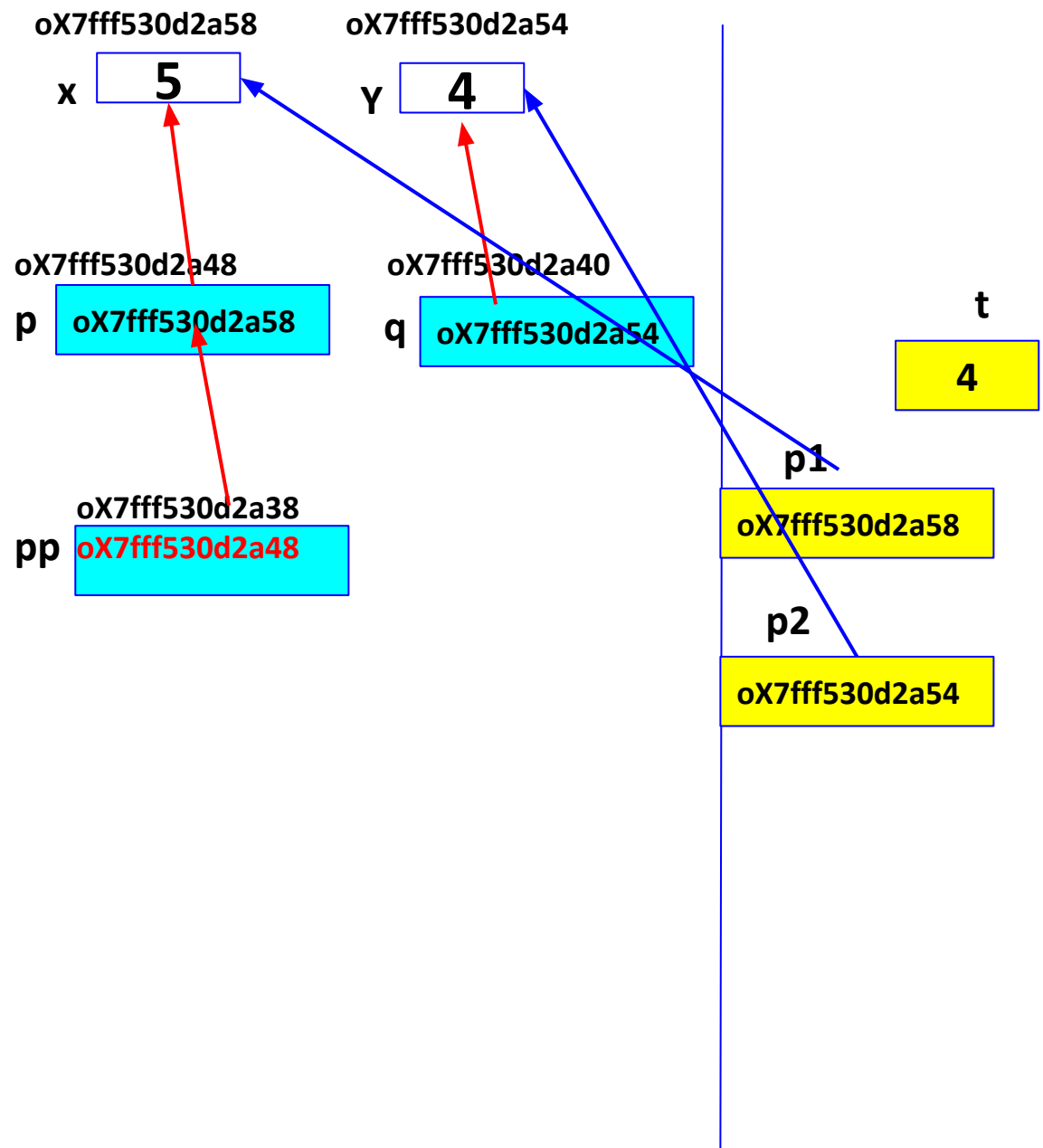
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    return 0;
}
```



```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{ int temporal = a;
  a=b;
  b=temporal;
}

void swap(int *p1, int *p2)
{ int t = *p1;
  *p1 = *p2;
  *p2 = t;
}
```

Pantalla:
Suma = 9
Primer swap
x =4 y = 5

Segundo swap

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x =" << x << " y =" << y << "\n\n";

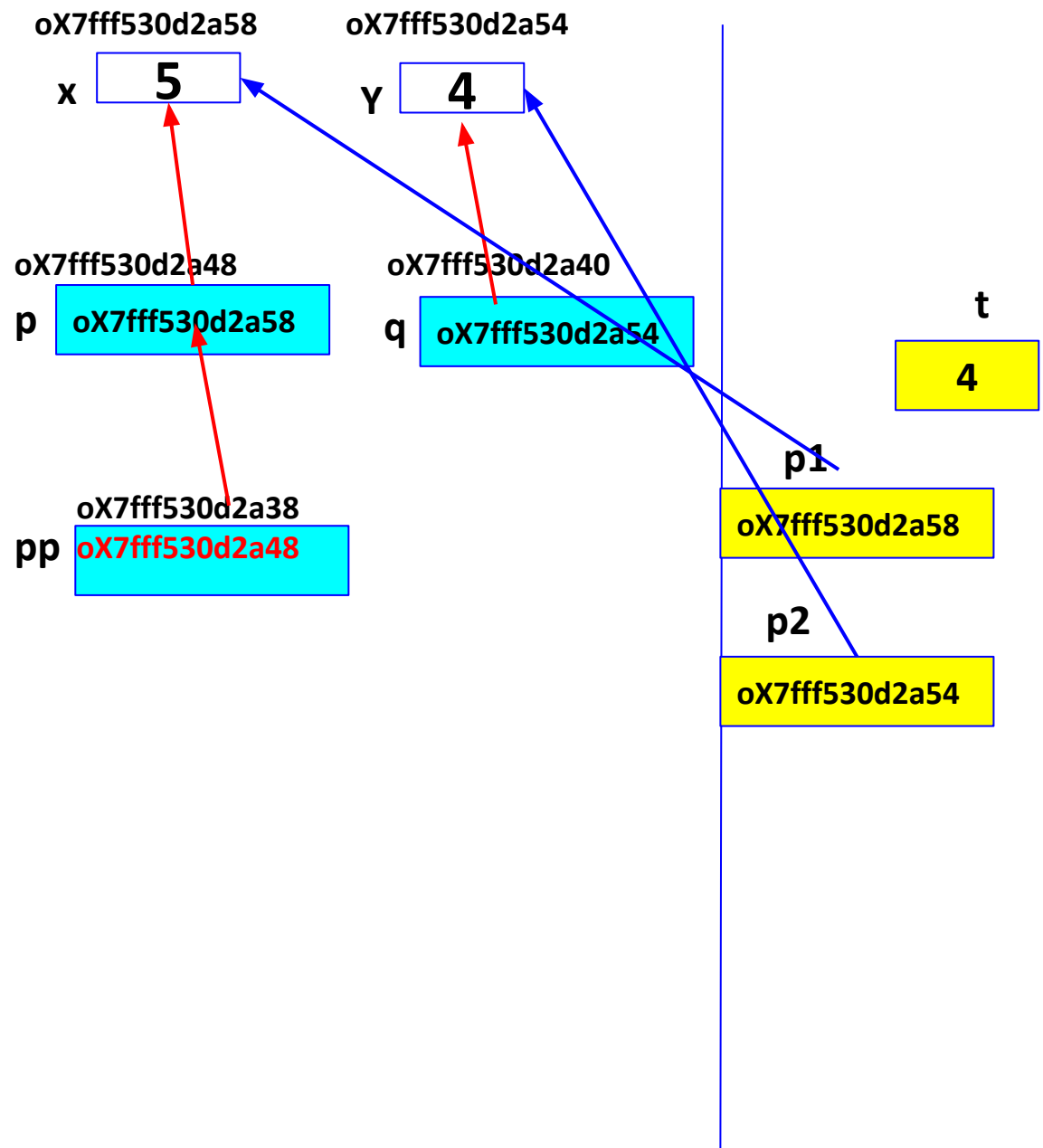
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x =" << x << " y =" << y << "\n\n";

    return 0;
}
```



```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{ int temporal = a;
  a=b;
  b=temporal;
}

void swap(int *p1, int *p2)
{ int t = *p1;
  *p1 = *p2;
  *p2 = t;
}
```

Pantalla:
Suma = 9
Primer swap
x =4 y = 5

Segundo swap
x = 5 y = 4

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x = " << x << " y = " << y << "\n\n";

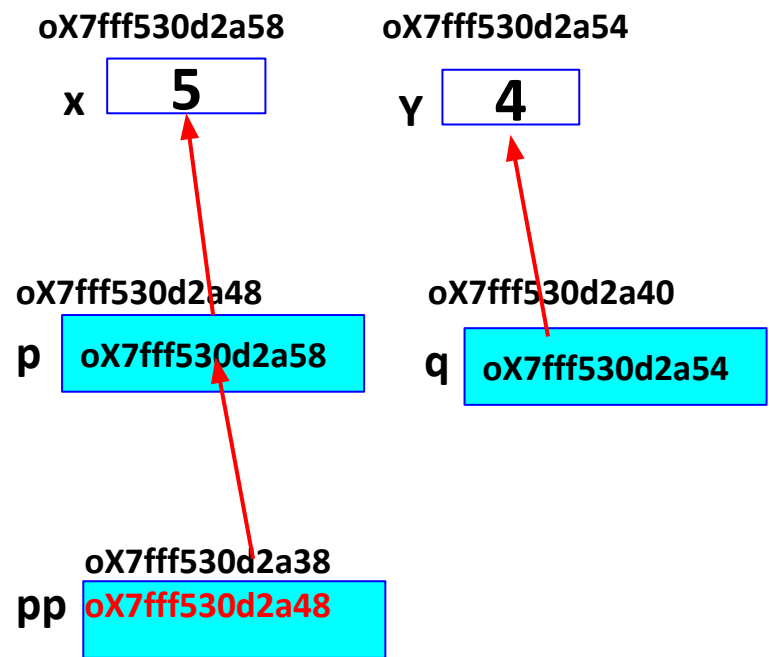
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    return 0;
}
```



```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{ int temporal = a;
  a=b;
  b=temporal;
}

void swap(int *p1, int *p2)
{ int t = *p1;
  *p1 = *p2;
  *p2 = t;
}
```

Pantalla:

Suma = 9

Primer swap

x =4 y = 5

Segundo swap

x = 5 y = 4

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x = " << x << " y = " << y << "\n\n";

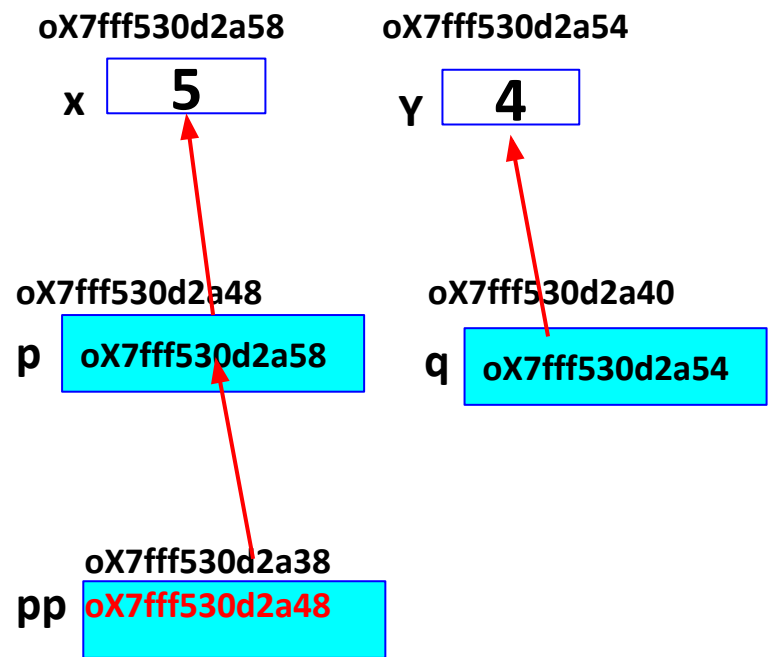
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    return 0;
}
```



```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{ int temporal = a;
  a=b;
  b=temporal;
}

void swap(int *p1, int *p2)
{ int t = *p1;
  *p1 = *p2;
  *p2 = t;
}
```

Pantalla:
Suma = 9
Primer swap
x =4 y = 5

Segundo swap
x = 5 y = 4

Tercer swap


```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x =" << x << " y =" << y << "\n\n";

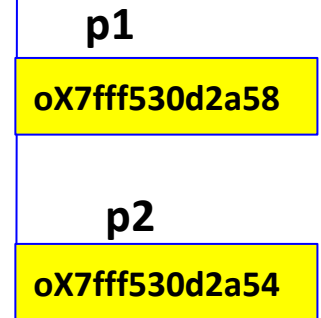
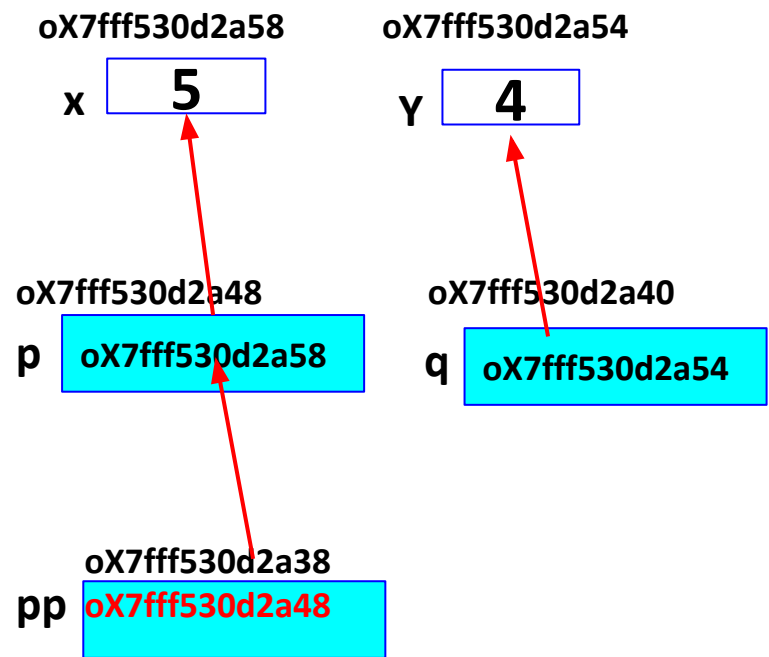
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x =" << x << " y =" << y << "\n\n";

    return 0;
}
```



```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{ int temporal = a;
  a=b;
  b=temporal;
}

void swap(int *p1, int *p2)
{ int t = *p1;
  *p1 = *p2;
  *p2 = t;
}
```

Pantalla:

Suma = 9

Primer swap

x =4 y = 5

Segundo swap

x = 5 y = 4

Tercer swap

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x =" << x << " y =" << y << "\n\n";

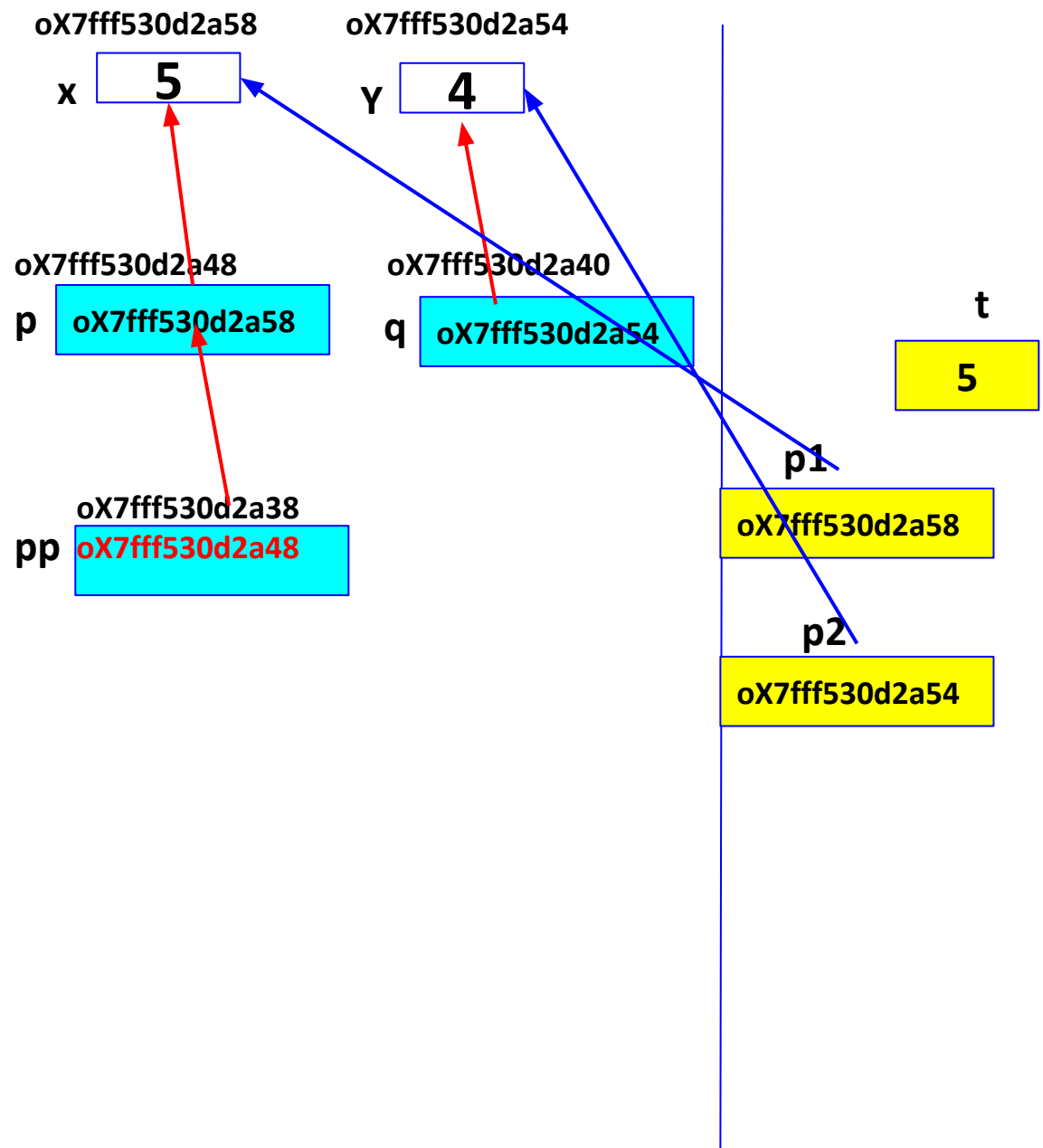
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x =" << x << " y =" << y << "\n\n";

    return 0;
}
```



```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{ int temporal = a;
  a=b;
  b=temporal;
}

void swap(int *p1, int *p2)
{ int t = *p1;
  *p1 = *p2;
  *p2 = t;
}
```

Pantalla:
Suma = 9
Primer swap
x =4 y = 5

Segundo swap
x = 5 y = 4

Tercer swap

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x =" << x << " y =" << y << "\n\n";

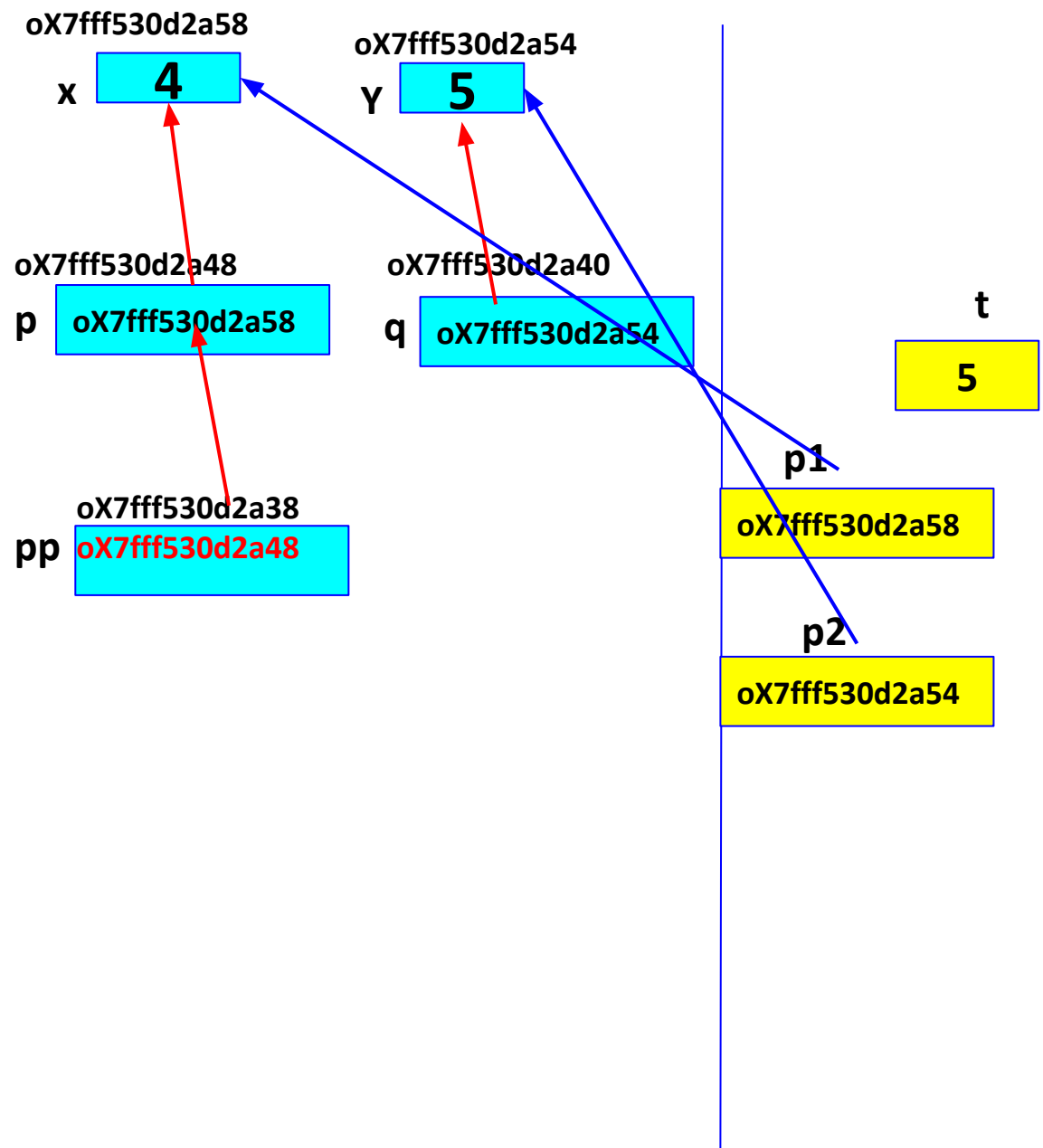
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x =" << x << " y =" << y << "\n\n";

    return 0;
}
```



```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{ int temporal = a;
  a=b;
  b=temporal;
}

void swap(int *p1, int *p2)
{ int t = *p1;
  *p1 = *p2;
  *p2 = t;
}
```

Pantalla:

Suma = 9

Primer swap

x =4 y = 5

Segundo swap

x = 5 y = 4

Tercer swap

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x = " << x << " y = " << y << "\n\n";

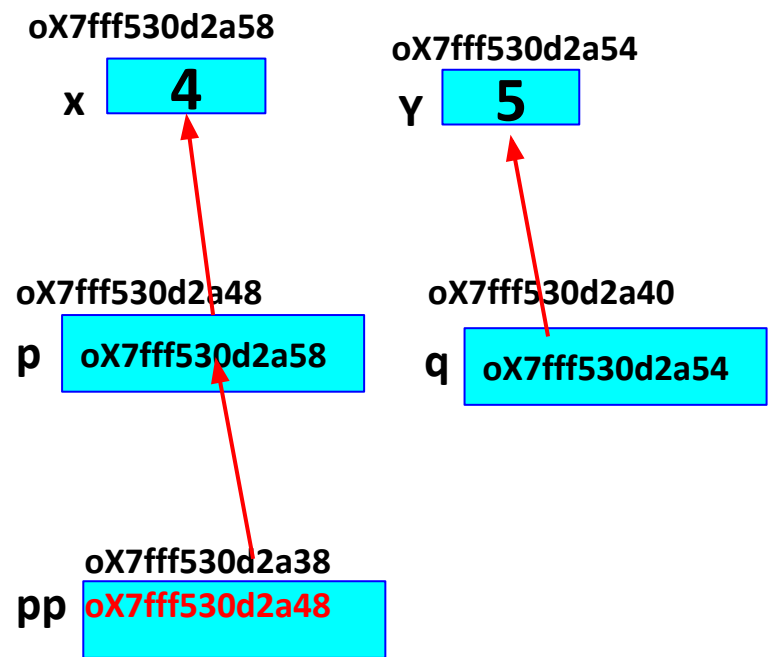
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    return 0;
}
```



```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{ int temporal = a;
  a=b;
  b=temporal;
}

void swap(int *p1, int *p2)
{ int t = *p1;
  *p1 = *p2;
  *p2 = t;
}
```

Pantalla:

Suma = 9

Primer swap
x =4 y = 5

Segundo swap
x = 5 y = 4

Tercer swap
x = 4 y = 5

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x = " << x << " y = " << y << "\n\n";

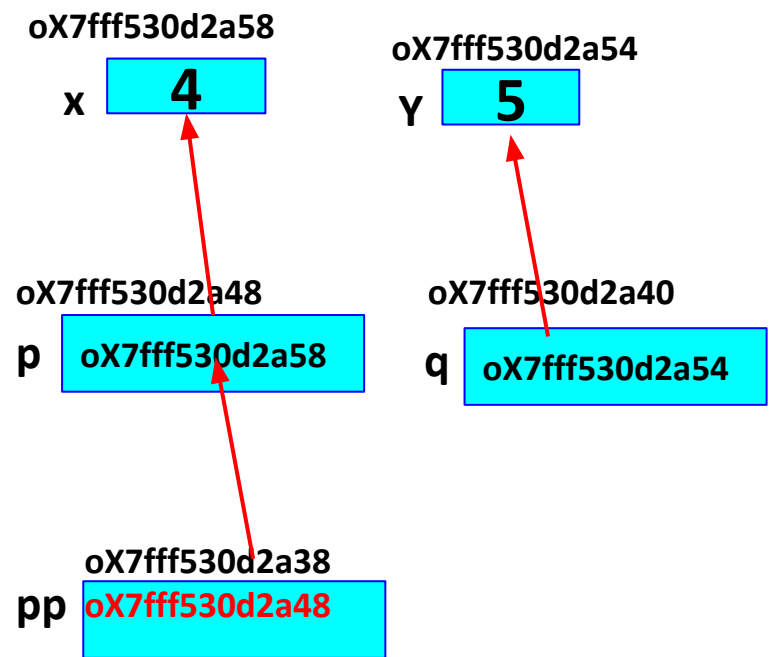
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    return 0;
}
```



```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{ int temporal = a;
  a=b;
  b=temporal;
}

void swap(int *p1, int *p2)
{ int t = *p1;
  *p1 = *p2;
  *p2 = t;
}
```

Pantalla:

Suma = 9

Primer swap
x =4 y = 5

Segundo swap
x = 5 y = 4

Tercer swap
x = 4 y = 5

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x = " << x << " y = " << y << "\n\n";

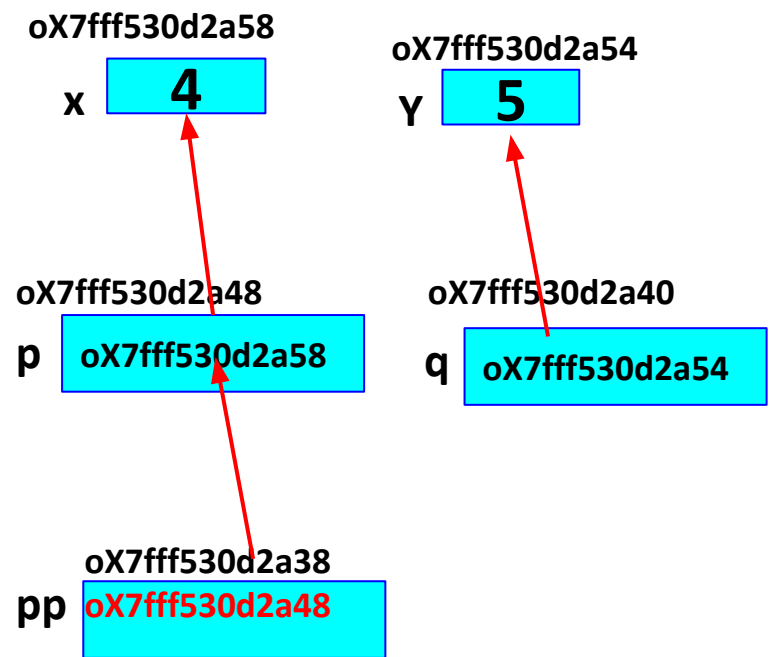
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    return 0;
}
```



```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{ int temporal = a;
  a=b;
  b=temporal;
}

void swap(int *p1, int *p2)
{ int t = *p1;
  *p1 = *p2;
  *p2 = t;
}
```

Pantalla:
Suma = 9
Primer swap
x =4 y = 5

Segundo swap
x = 5 y = 4

Tercer swap
x = 4 y = 5

Cuarto swap

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x =" << x << " y =" << y << "\n\n";

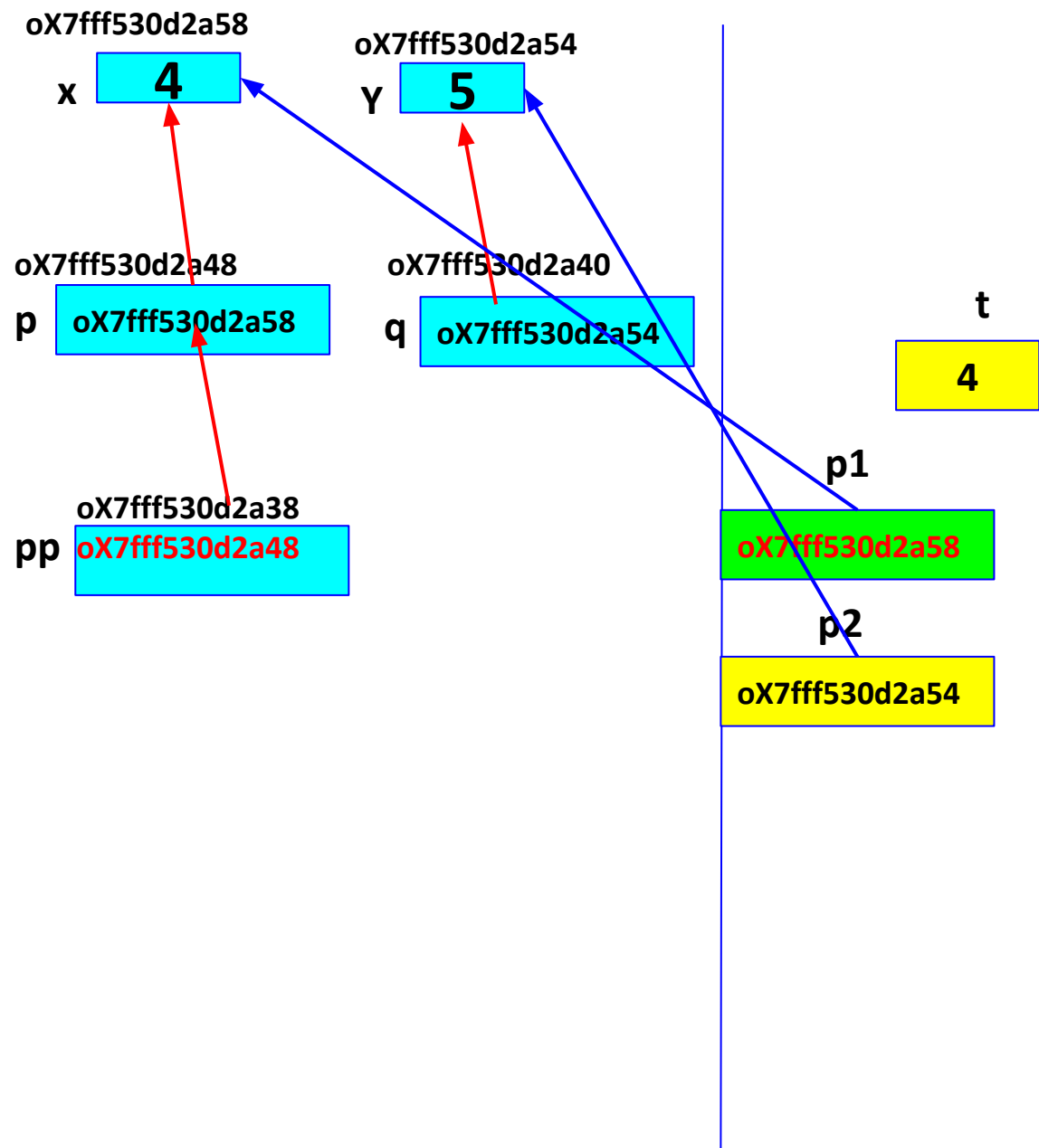
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x =" << x << " y =" << y << "\n\n";

    return 0;
}
```



```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{ int temporal = a;
  a=b;
  b=temporal;
}

void swap(int *p1, int *p2)
{ int t = *p1;
  *p1 = *p2;
  *p2 = t;
}
```

Pantalla:

Suma = 9

Primer swap
x =4 y = 5

Segundo swap
x = 5 y = 4

Tercer swap
x = 4 y = 5

Cuarto swap

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x =" << x << " y =" << y << "\n\n";

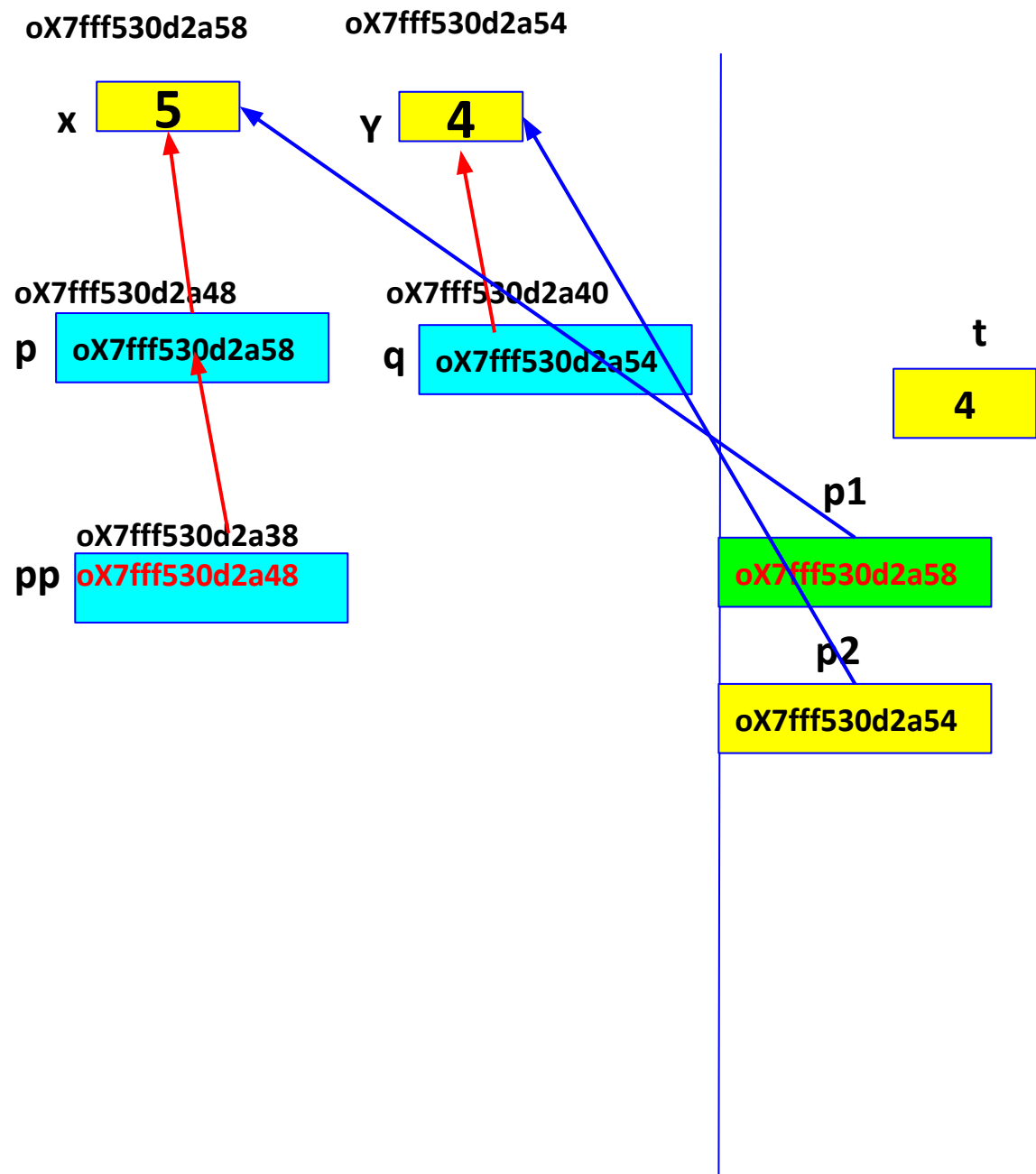
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x =" << x << " y =" << y << "\n\n";

    return 0;
}
```



```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{ int temporal = a;
  a=b;
  b=temporal;
}

void swap(int *p1, int *p2)
{ int t = *p1;
  *p1 = *p2;
  *p2 = t;
}
```

Pantalla:
Suma = 9
Primer swap
x =4 y = 5

Segundo swap
x = 5 y = 4

Tercer swap
x = 4 y = 5

Cuarto swap


```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x = " << x << " y = " << y << "\n\n";

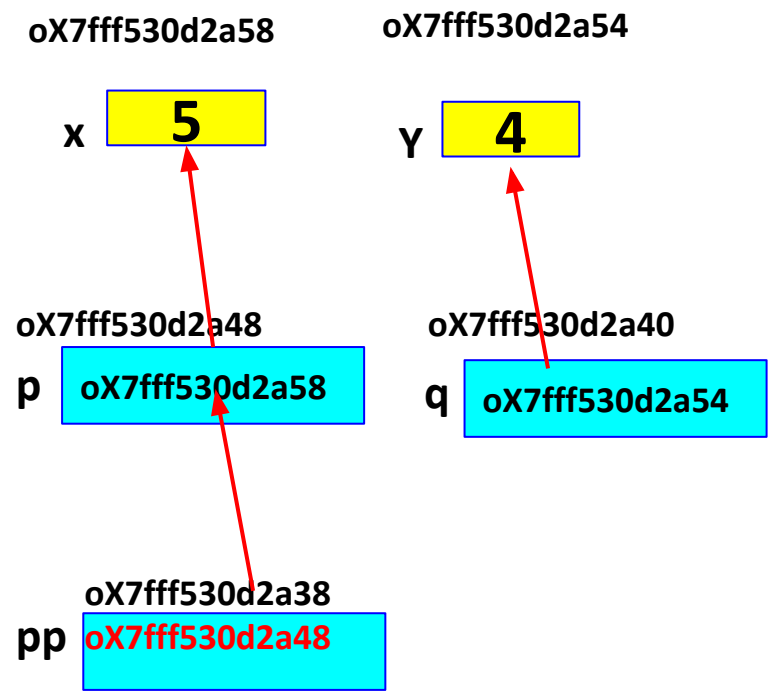
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    return 0;
}
```



```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{ int temporal = a;
  a=b;
  b=temporal;
}

void swap(int *p1, int *p2)
{ int t = *p1;
  *p1 = *p2;
  *p2 = t;
}
```

Pantalla:

Suma = 9

Primer swap
x =4 y = 5

Segundo swap
x = 5 y = 4

Tercer swap
x = 4 y = 5

Cuarto swap
x = 5 y = 4

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x = " << x << " y = " << y << "\n\n";

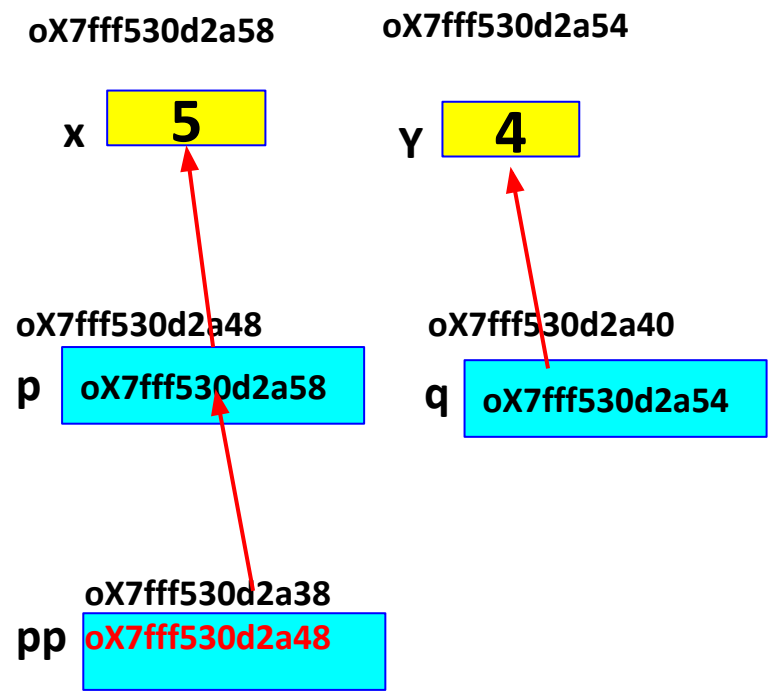
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    return 0;
}
```



```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{
    int temporal = a;
    a=b;
    b=temporal;
}

void swap(int *p1, int *p2)
{
    int t = *p1;
    *p1 = *p2;
    *p2 = t;
}

Pantalla:
Suma = 9
Primer swap
x =4 y = 5
Segundo swap
x = 5 y = 4
Tercer swap
x = 4 y = 5

Cuarto swap
x = 5 y = 4

Quinto swap
```

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x =" << x << " y =" << y << "\n\n";

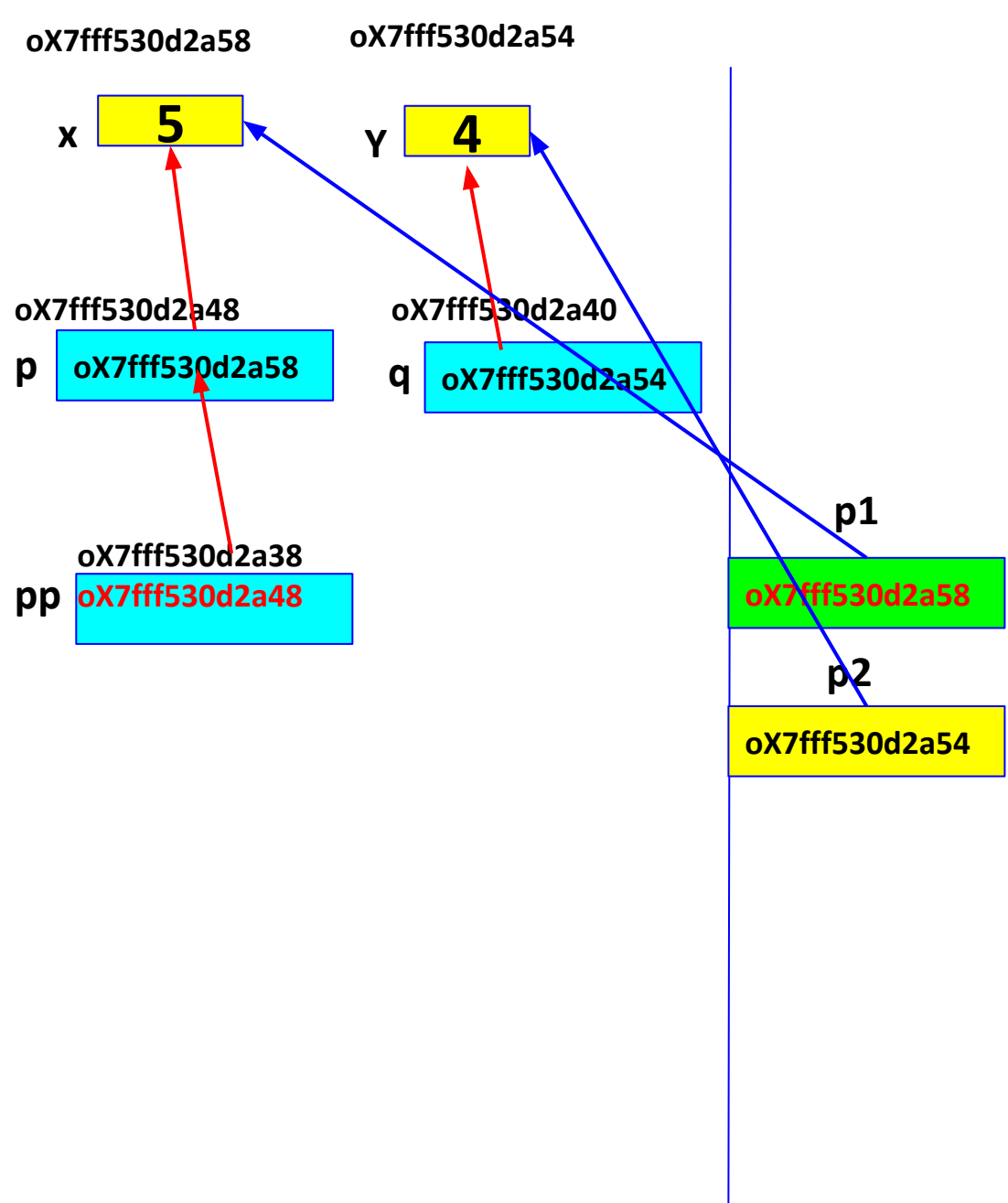
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x =" << x << " y =" << y << "\n\n";

    return 0;
}
```



```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{ int temporal = a;
  a=b;
  b=temporal;
}

void swap(int *p1, int *p2)
{ int t = *p1;
  *p1 = *p2;
  *p2 = t;
}
```

Pantalla:

Suma = 9

Primer swap

x =4 y = 5

Segundo swap

x = 5 y = 4

Tercer swap

x = 4 y = 5

Cuarto swap

x = 5 y = 4

Quinto swap

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x =" << x << " y =" << y << "\n\n";

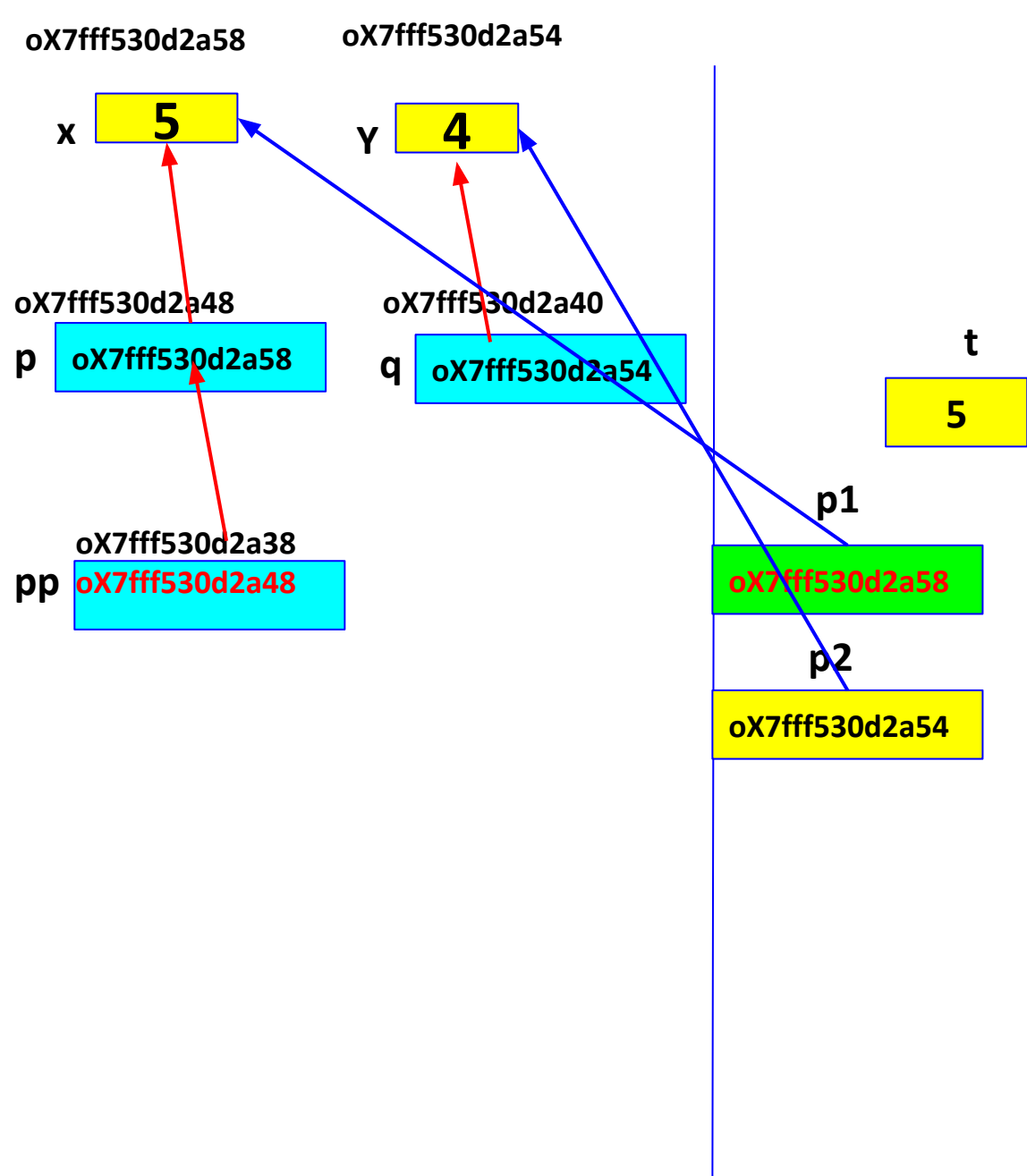
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x =" << x << " y =" << y << "\n\n";

    return 0;
}
```



```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{
    int temporal = a;
    a=b;
    b=temporal;
}

void swap(int *p1, int *p2)
{
    int t = *p1;
    *p1 = *p2;
    *p2 = t;
}

Pantalla:
Suma = 9
Primer swap
x =4 y = 5
Segundo swap
x = 5 y = 4
Tercer swap
x = 4 y = 5

Cuarto swap
x = 5 y = 4

Quinto swap
```

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x = " << x << " y = " << y << "\n\n";

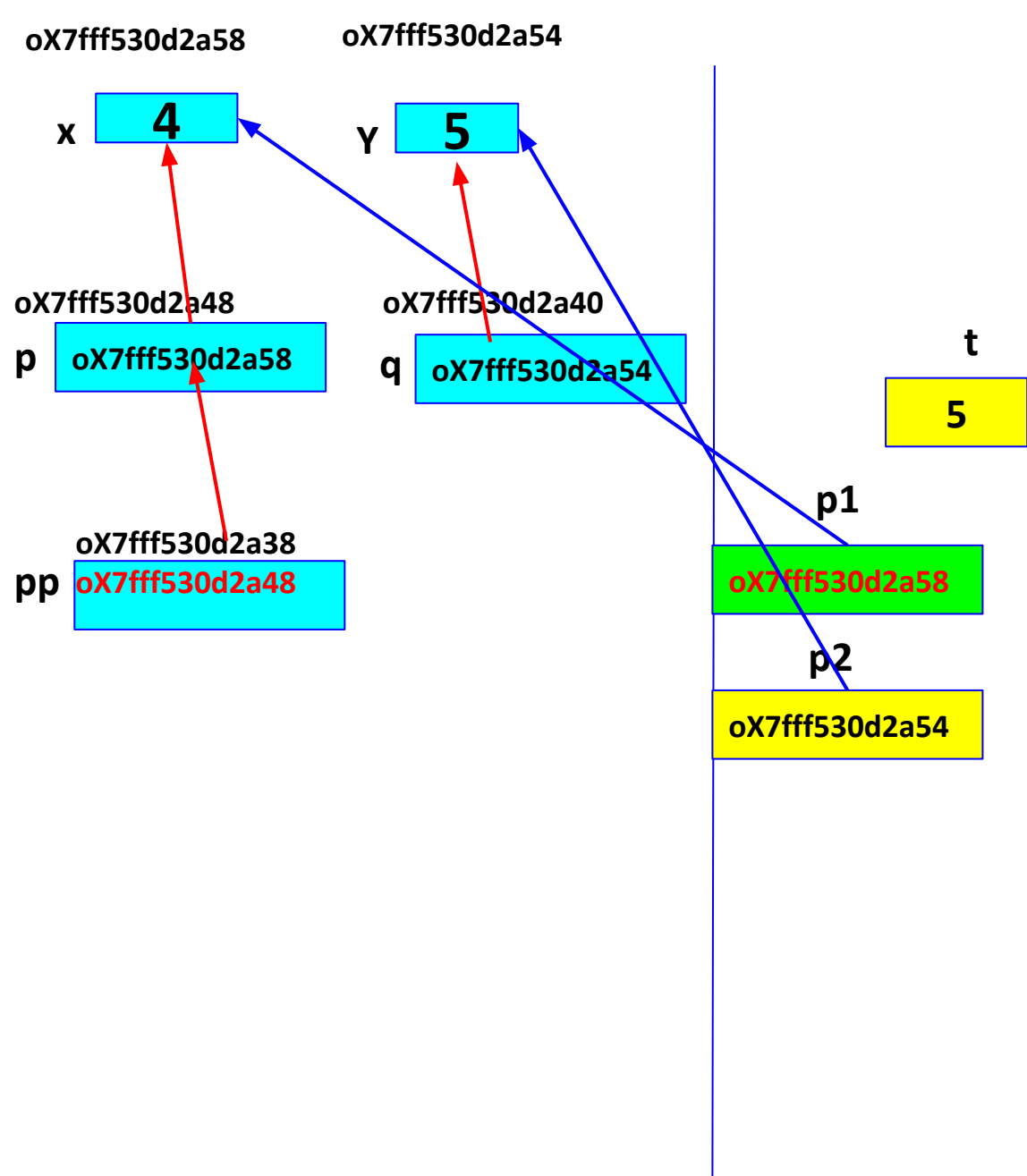
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    return 0;
}
```



```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{
    int temporal = a;
    a=b;
    b=temporal;
}

void swap(int *p1, int *p2)
{
    int t = *p1;
    *p1 = *p2;
    *p2 = t;
}

Pantalla:
Suma = 9
Primer swap
x =4 y = 5
Segundo swap
x = 5 y = 4
Tercer swap
x = 4 y = 5

Cuarto swap
x = 5 y = 4

Quinto swap
```

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x =" << x << " y =" << y << "\n\n";

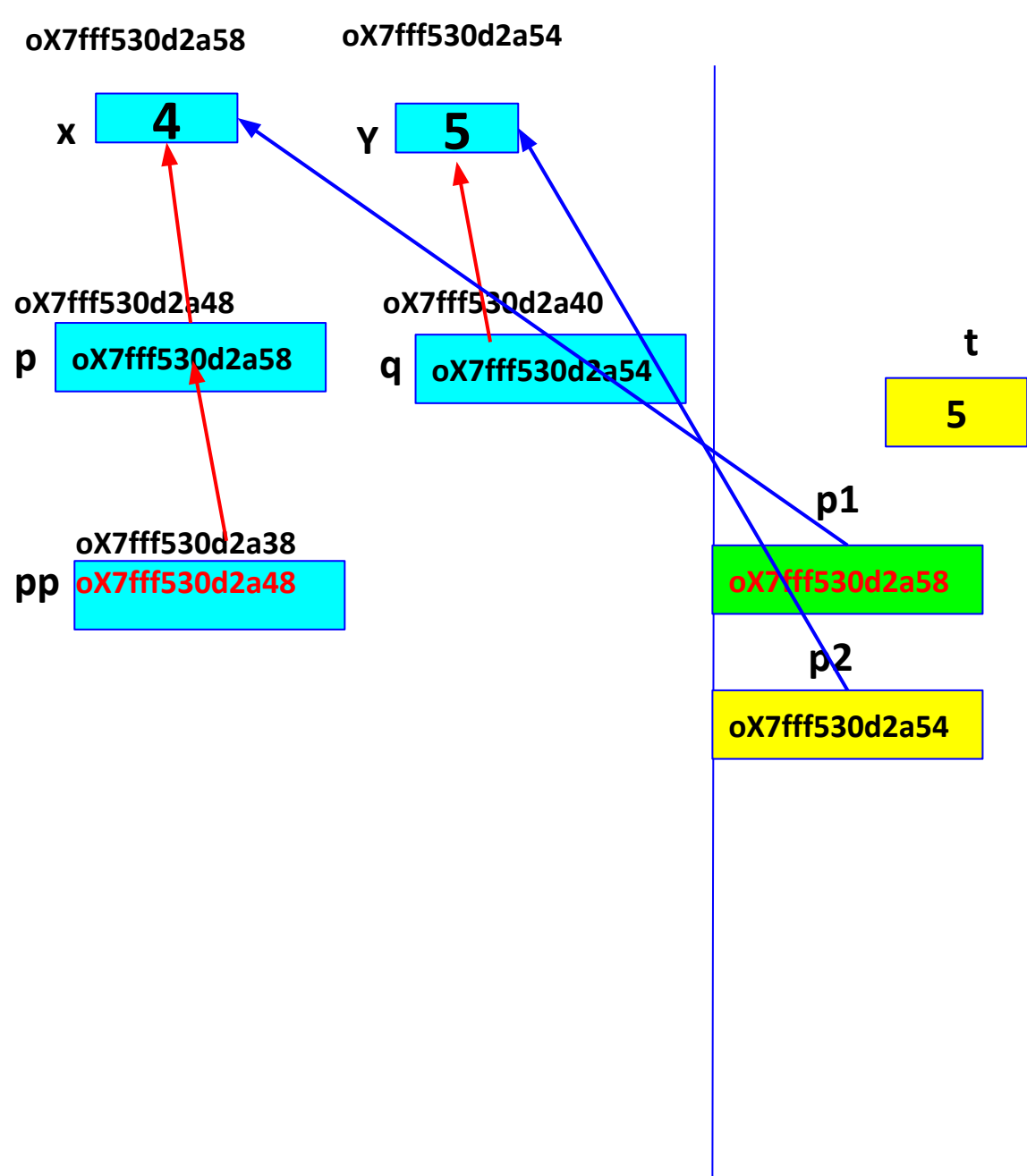
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x =" << x << " y =" << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x =" << x << " y =" << y << "\n\n";

    return 0;
}
```



```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{
    int temporal = a;
    a=b;
    b=temporal;
}

void swap(int *p1, int *p2)
{
    int t = *p1;
    *p1 = *p2;
    *p2 = t;
}

Pantalla:
Suma = 9
Primer swap
x =4 y = 5
Segundo swap
x = 5 y = 4
Tercer swap
x = 4 y = 5

Cuarto swap
x = 5 y = 4

Quinto swap
x = 4 y = 5
```

```
int main()
{
    int x=5, y=4, *p, *q, **pp;
    p = &x; q = &y; pp = &p;

    cout <<"Suma = " << fSuma(x,y) << "\n\n";

    cout <<"Primer swap\n";
    swap(x,y);
    cout << "x = " << x << " y = " << y << "\n\n";

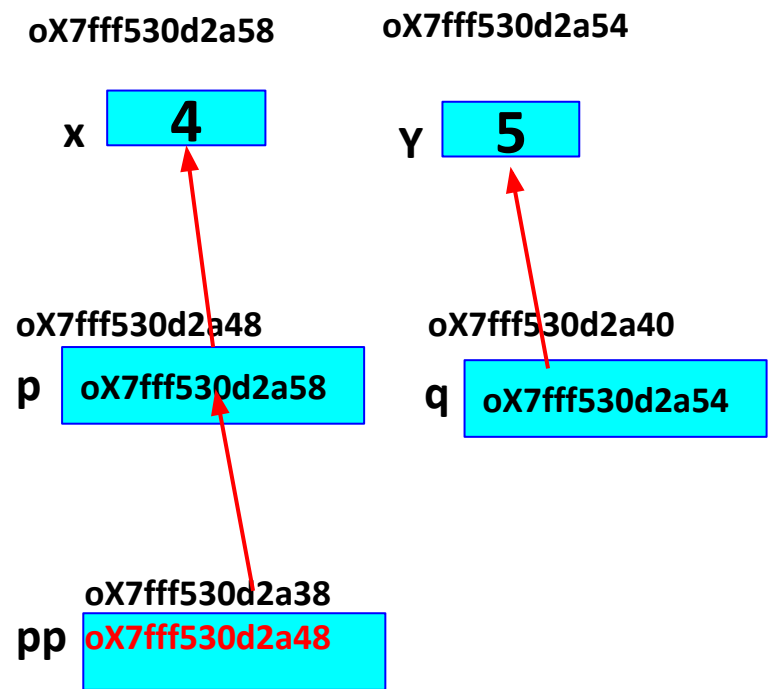
    cout <<"Segundo swap\n";
    swap(&x, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Tercer swap\n";
    swap(p, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Cuarto swap\n";
    swap(*pp, &y);
    cout << "x = " << x << " y = " << y << "\n\n";

    cout <<"Quinto swap\n";
    swap(*pp, q);
    cout << "x = " << x << " y = " << y << "\n\n";

    return 0;
}
```



```
int fSuma(int a, int b)
{
    return (a+b);
}

void swap(int &a, int &b)
{
    int temporal = a;
    a=b;
    b=temporal;
}

void swap(int *p1, int *p2)
{
    int t = *p1;
    *p1 = *p2;
    *p2 = t;
}

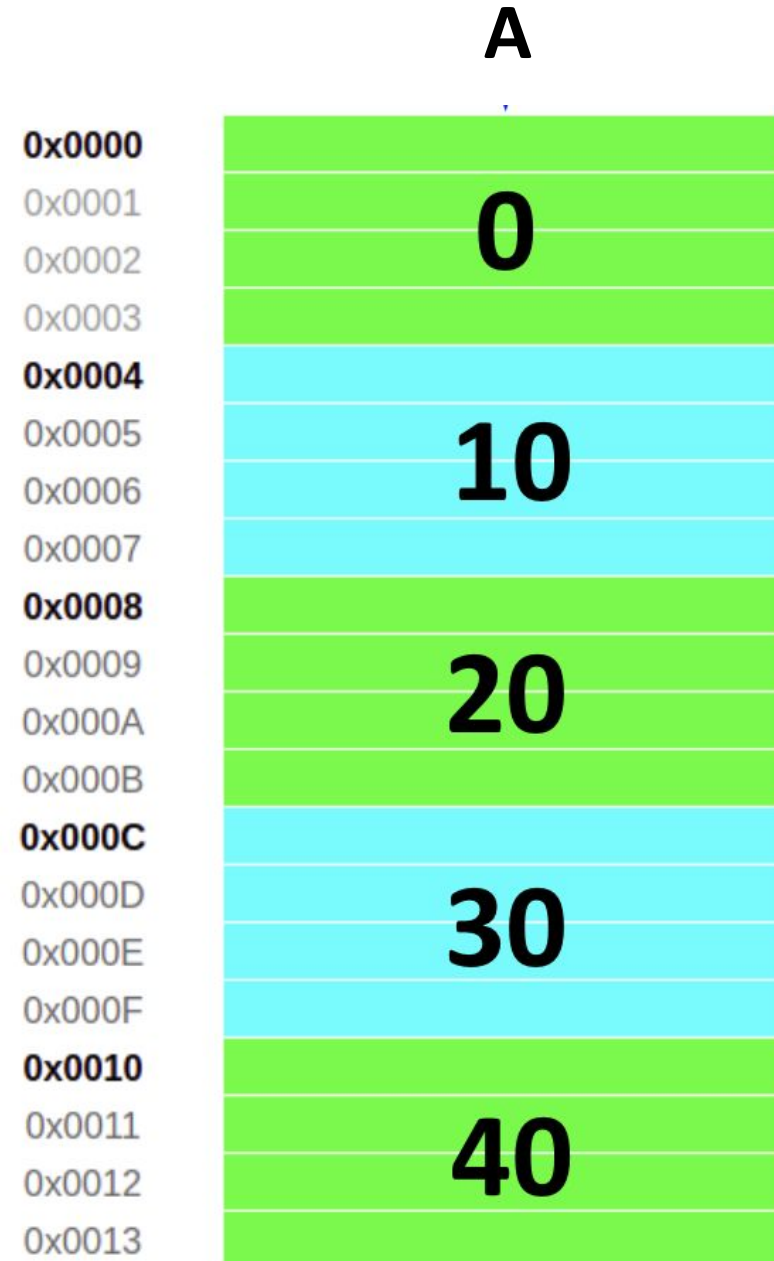
Pantalla:
Suma = 9
Primer swap
x =4 y = 5
Segundo swap
x = 5 y = 4
Tercer swap
x = 4 y = 5

Cuarto swap
x = 5 y = 4

Quinto swap
x = 4 y = 5
```

Array y direcciones


```
int A[5]={0,10,20,30,40};
```



```

#include <iostream>
using namespace std;

int main()
{

double A[7]={ 71,72,73,74,75,76,77};

cout << "\n Inicio del array " << A << "\n";

cout << "\nImprimimos las direcciones de cada casillero del Arreglo A
\n";

for(int i=0; i<7; i++)
    cout << "&A[" << i << "]= " << &A[i] << " Guarda el " << A[i] << "\n";

return 0;
}

```

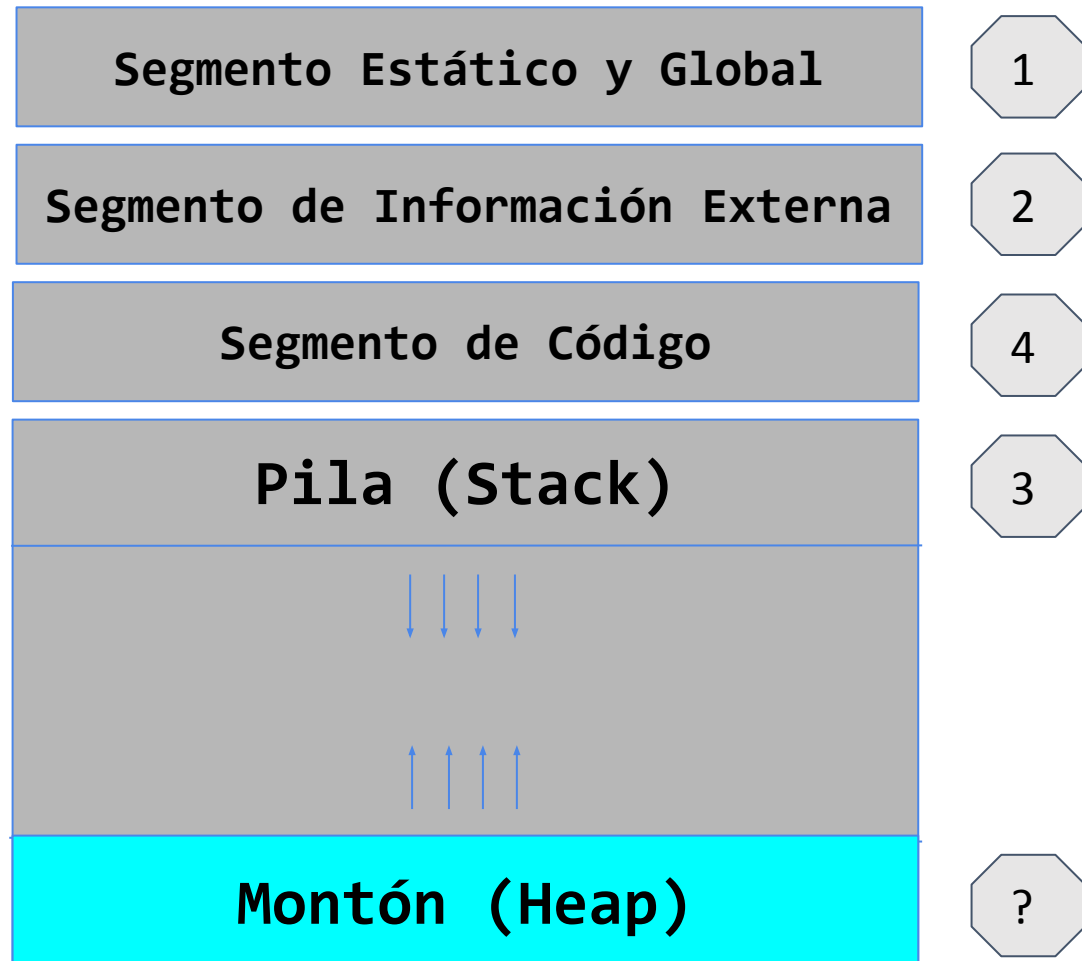
Inicio del array 0x7fff57f45ab0

Imprimimos las direcciones de cada casillero del Arreglo A

&A[0]= 0x7fff57f45ab0	Guarda el 71
&A[1]= 0x7fff57f45ab8	Guarda el 72
&A[2]= 0x7fff57f45ac0	Guarda el 73
&A[3]= 0x7fff57f45ac8	Guarda el 74
&A[4]= 0x7fff57f45ad0	Guarda el 75
&A[5]= 0x7fff57f45ad8	Guarda el 76
&A[6]= 0x7fff57f45ae0	Guarda el 77

Manejando memoria directamente

Programa de C++ en la memoria primaria



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

```
int varGlobal = 20;
```

```
int main(int argc, char * argv[])
{
    int varLocal = 10;
    int* ptrVarLocal = &varLocal;

    cout << varLocal << "\n";
    return 0;
}
```

Al Heap solo se puede acceder a través del uso de punteros.

Operadores para asignar y liberar memoria dinámica

new asigna memoria

delete libera memoria asignada por new

new asigna memoria dinámicamente

int *pi = new int; // **p** apunta a un espacio asignado dinámicamente

new construye un objeto de tipo **int** en un espacio libre de memoria y retorna el puntero a ese objeto. El objeto no se inicializa.

string *ps= new string; // string vacio

int *pi1 = new int; // **pi1** puntero a un **int**, el entero no se ha inicializado

int *pi2 = new int(1024); // **pi2** apunta a un objeto int que tiene el valor 1024

int *pi3 = new int(); // **pi3** apunta a un objeto int que tiene el valor cero

Liberando memoria dinámica:

Para prevenir que la memoria se sature, se debe eliminar el espacio asignado dinámicamente una vez que se haya terminado de utilizar.

```
delete p;    // libera el espacio  
             // p debe ser un puntero a memoria asignada dinámicamente
```

Acceso al Heap

```
int* ptrMonton = nullptr;
```

```
int* ptrVar = nullptr;
```

...

```
int var = 20;
```

```
ptrVar = &var;
```

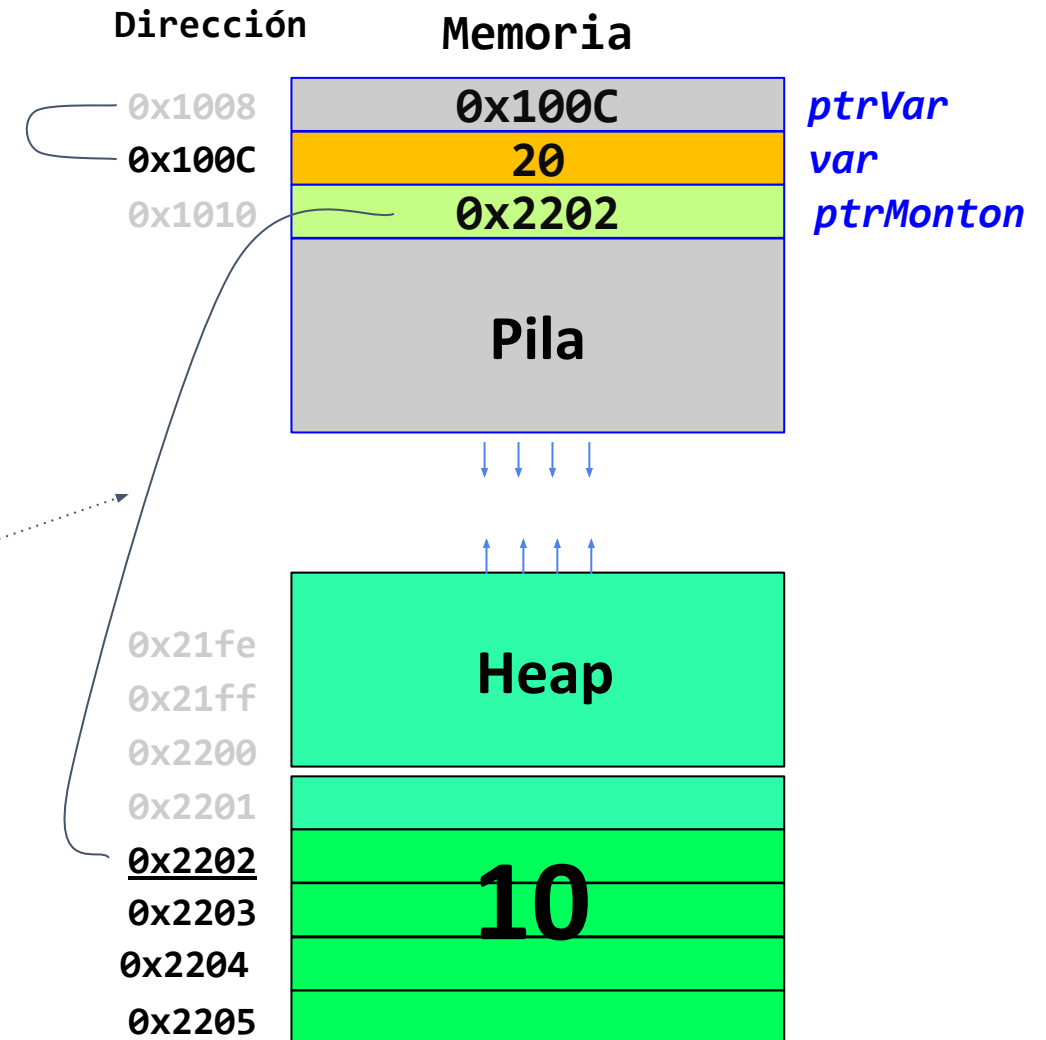
...

```
ptrMonton = new int;
```

```
*ptrMonton = 10;
```

...

```
delete ptrMonton;
```



Ejemplo 1:

Desarrolla un programa que permita leer dos números de tipo double, se almacenen utilizando memoria dinámica y luego halle la suma, la diferencia y el producto de estos números.

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  { double *pnumero1= nullptr, *pnumero2= nullptr;
7      ;
8      pnumero1 = new double;
9      pnumero2 = new double;
10
11     cout << "Numero 1 : ";
12     cin >> *pnumero1;    //-- se lee el numero en el sitio apuntado por el puntero
13     cout << "Numero 2 : ";
14     cin >> *pnumero2;
15     cout << "\n";
16     cout << "La Suma es : " << *pnumero1 + *pnumero2 << "\n";
17     cout << "La Diferencia es : " << *pnumero1 - *pnumero2 << "\n";
18     cout << "El Producto es : " << *pnumero1 * *pnumero2 << "\n";
19     delete pnumero1;
20     delete pnumero2;
21     return 0;
22 }
```

Numero 1 : 5
Numero 2 : 3

La Suma es : 8
La Diferencia es : 2
El Producto es : 15

Importante:

Suponiendo que un dato de tipo double en el ambiente de Clion utiliza 8 bytes para ser almacenado.

¿ Cuántos bytes de memoria se necesita para almacenar todas las variables definidas en la función main?

¿ Se usa espacio de la pila?

¿ Se usa espacio del heap?

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {  double *pnumero1= nullptr, *pnumero2= nullptr;
7
8      pnumero1 = new double;
9      pnumero2 = new double;
10
11     cout << "Numero 1 : ";
12     cin >> *pnumero1;    //-- se lee el numero en el sitio apuntado por el puntero
13     cout << "Numero 2 : ";
14     cin >> *pnumero2;
15     cout << "\n";
16     cout << "La Suma es : " << *pnumero1 + *pnumero2 << "\n";
17     cout << "La Diferencia es : " << *pnumero1 - *pnumero2 << "\n";
18     cout << "El Producto es : " << *pnumero1 * *pnumero2 << "\n";
19     delete pnumero1;
20     delete pnumero2;
21     return 0;
22 }
```

Los objetos creados dinámicamente existen hasta que sean liberados de la memoria:

**Veamos funciones que retornan memoria dinámica.
Es responsabilidad del que programa liberar la memoria.**

```
Foo * factory (T arg)
{
    // process arg as parameter
    return new Foo(arg); //-- caller is responsible for deleting this memory
}
```

```
void use_factory(arg)
{
    Foo *p =factory(arg);
    // use p but do not delete it
    // p goes out of scope, but the memory to which p point is not freed!.
}
```

```
void use_factory(arg)
{
    Foo *p =factory(arg);
    // use p
    delete p;
}
```

Unidad 6: Punteros

<http://bit.ly/2p3fgiD>

Profesores:

Ernesto Cuadros-Vargas, PhD.

[**ecuadros@utec.edu.pe**](mailto:ecuadros@utec.edu.pe)

María Hilda Bermejo, M. Sc.

[**mbermejo@utec.edu.pe**](mailto:mbermejo@utec.edu.pe)