

Ejercicio

Analice el siguiente código e indique cómo se comporta cada instrucción respecto a la memoria principal.

¿Qué imprime en pantalla el código?

¿Por qué?

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

const int y = 5;

void addConstant(int x) {
    int r = x + y;
    x = r;
    cout << x << " + " << y << " = " << r << "\n";
}

int main() {
    static int x = -1;
    new int();
    addConstant(x);
    cout << "x = " << x;
}
```

Explicación del código:

1. Cuando se ejecuta el programa, lo primero en ejecutar es el int main.
2. Como podemos observar podemos ver que hay una variable estática entero con una variable "x" que tiene el valor de -1.
3. Podemos ver un new int, que lo único que hace guardar un pedazo en la memoria, pero vacía.
4. Luego ejecuta la función "addconstant(x)", como podemos ver la función requiere un parámetro y el parámetro es el entero estático de -1.
5. Ahora que estamos en la función se dice que el tipo de dato entero "r" es igual a "x" + "y".
6. Sabemos que la variable "y" es una constante, eso quiere decir que no va a cambiar y "y" tiene un valor de 5.
7. Cuando sumamos entonces "x" + "y", es decir, -1 + 5 es igual a 4, entonces r es igual a 4.
8. Ahora en la siguiente línea dice que "x" = "r", como sabemos que la variable "X" valía -1 ahora pasa a valer 4, como "r".
9. En la siguiente línea, imprime en la consola "4 + 5 = 4". Aunque matemáticamente es incorrecto, el programa si cumple con la función de mostrar en pantalla, sabemos que "x" es igual a 4 e igual que "r" y "y" tiene el valor de 5.
10. Ahora la función ya termino de hacer su trabajo y regresamos al main.
11. Luego imprime lo siguiente x = -1, porque "x" no imprimió 4 si en la función se imprimió 4, la razón de esta era que no estábamos cambiando la variable "x fuera de la función", si no la "x" que se encuentra en la función.